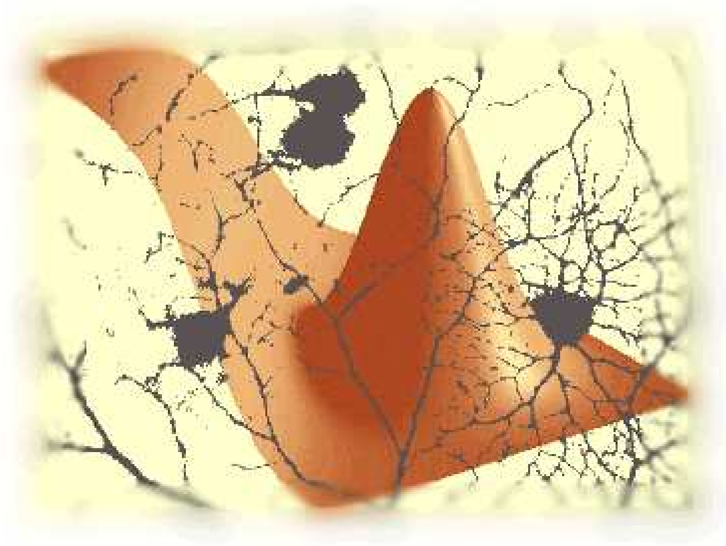

Les Réseaux Neuro-Flous

Yann MORERE

17 mai 2001

Résumé

Les réseaux de neuro-flous ont connu un essor considérable. Un résumé sur tout ce qu'il faut savoir sur les réseau de neuro-flous contenu dans ces pages. Après une introduction à la logique floue, on détaillera les différents modèles d'identification floue et de commandes floue. On terminera par une présentation des réseaux neuro-flous et de leurs principes d'apprentissage ainsi que leurs avantages.

Table des matières

Table des figures

Chapitre 1

Réseaux de Neurones et Systèmes Flous

1.1 Introduction

La théorie des réseaux de neurones et des systèmes flous ont été développée depuis plusieurs année. Mais la plupart du temps dans des domaines séparés et différents. Depuis peu les chercheurs utilisent ces techniques conjointement dans domaines qui ne touchent ni le flou ni les réseaux de neurones. Ils utilisent donc les réseaux de neurones et la théorie floue d'un manière unifiée. Au départ ces techniques sont utilisées pour le calcul élémentaire, algèbre linéaire et la probabilité, elles sont alors utilisées dans des techniques plus complexes comme le traitement du signal, procédés aléatoires ainsi que pour l'identification et la commande de système complexes.

Les systèmes á base de réseaux de neurones et de logique floue font de bons approximateurs de fonctions á partir de données d'échantillons.

Mais ils apportent une chose en plus par rapport á l'approche statistique : ils n'ont pas besoin de connaissance a priori sur le fonctionnement interne du système. Les systèmes flous et neuronaux n'ont pas besoin de modèle mathématique. Ce sont de purs estimateurs.

Les systèmes neuronaux et flous sont des estimateurs numériques et des systèmes dynamiques. La théorie neuronal noie dans la sémantique mathématiques des systèmes dynamiques, la commande adaptative et les statistiques. La théorie floue, elle, se chevauche avec ces sémantiques et en plus avec les probabilités, la logique mathématiques et les théorie de mesure.

En général les systèmes neuronaux et flous sont utilisés pour améliorer les performances des systèmes réels.

1.2 La Théorie Floue

La théorie des ensembles flous est une théorie mathématiques qui permet de modéliser des notions vagues du langage naturel pour pallier l'inadéquation de la théorie des ensembles classiques dans ce domaine. En effet cette dernière définit de façon abrupte une frontière entre deux catégories d'éléments : ceux qui appartiennent à l'ensemble et ceux qui sont à l'extérieur.

Contrairement la logique floue á sa base considère que tout est affaire de degrés. En logique floue rien n'est manichéen. Tout peut être coder en niveau de gris. Elle donne un interprétation déterministe des probabilités et de la stochastique. Mathématiquement en logique floue, les variables floues sont multivaluées et possède une multivalance (on peut pendre comme exemple la vérité, le mensonge et la valeur qui correspond au cas ou on ne sais rien) Le degrés de cette valeur indéterminée correspond á la multivaluation de la variable floue. Par exemple dans une approche floue la paradoxe correspond á la valeur intermédiaire entre les deux valeurs antithétiques. Il s'agit ici d'un paradoxe a 2 dimensions (2 valeurs antithétiques) mais on peut imaginer de créer des points médiants entre plusieurs valeurs antithétiques (> 2).

1.2.1 Théorie Floue au XXème Siècle

Les paradoxes logiques et le principe d'incertitude d'Heisenberg ont permis le développement de la logique floue dans les années 1920 et 1930. A cette époque les théoristes quantiques avaient introduit l'incertitude d'un système bivalent en la valeur médiane de cette bivalence. La deuxième étape fût l'énoncé de l'incertitude comme une suite de différents degrés d'incertitude bornés par les valeur bivalente de la relation.

Le logicien polonais Jan Lukasiewicz a développé un système multivalué composé de 3 valeurs au début des années 30. Ensuite Lukasiewicz a étendue le champ des valeurs de $\{0, 1/2, 1\}$ á tous les rationnels compris en $[0, 1]$

En 1930, le philosophe quantique Max Black a appliqué cette logique continue du point de vue des composants á des ensembles d'éléments. Historiquement il a crée les premières fonctions d'appartenance d'un ensemble flou.

En 1965 Lofti Zadeh a publié l'article "Fuzzy Sets" qui développe la théorie d'ensembles multivalués et qui introduit le terme *flou* dans la littérature scientifique.

Zadeh a donc étendu la fonction indicateur bivalent I_A d'un sous-ensemble non-flou A de X

$$I_A(x) = \begin{cases} 1 \; si \; x \in A \\ 0 \; si \; x \notin A \end{cases} \tag{1.1}$$

en une fonction indicateur multivaluée ou **fonction d'appartenance** $m_A : X \to [0, 1]$. On peut donc combiner ces ensembles flous avec l'opérateur "point" des fonctions indicateurs

$$I_{A \cap B}(x) = \min\left(I_A(x), I_B(x)\right) \tag{1.2}$$

$$I_{A \cup B}(x) = \max\left(I_A(x), I_B(x)\right) \tag{1.3}$$

$$I_{A^c}(x) = 1 - I_A(x) \tag{1.4}$$

$$A \subset B \; ssi \; I_A(x) \leq I_B(x) \; pour \; tout \; x \in X \tag{1.5}$$

La valeur d'appartenance $m_A(x)$ mesure le degrés avec lequel l'élément x appartient á l'ensemble A.

$$m_A(x) = degrés(x \in A) \tag{1.6}$$

1.2.2 Degré d'Appartenance et Probabilité

Ce sont deux notions qu'il ne faut pas confondre étant différente par nature. Ce sont dans les deux cas des nombres réels pris dans l'intervalle fermé $[0,1]$ et mesure tous les deux une incertitude mais avec des objectifs différents.

- Le degrés d'appartenance mesure l'incertitude ou la croyance par rapport à une notion vague. Cette mesure indique suivant l'univers du discours considéré et les conventions adoptées, l'appartenance à des degrés divers des individus aux sous-ensembles considérés.
- La probabilité est associée à la notion d'événement. C'est la mesure associée à l'occurrence d'un événement. Les chances de réalisation de cette événement dépend directement de la valeur attribuée à la probabilité correspondante.

On note la valeur d'appartenance d'un sous-ensemble A á un ensemble B par $S(A,B) = \mathrm{D}egrés(A \subset B)$. La théorie des sous-ensemble offre un environnement d'ensemble théorique unifiée. Par exemple si l'ensemble A n'est composé que de l'ensemble des singletons $\{x_i\}$ alors l'appartenance de $\{x_i\}$ dans B est la valeur d'appartenance $m_B(x_i)$.

Le comportement aléatoire équivaut á l'incertitude que l'on a lorsque l'on a un ensemble non-flou B qui est partiellement contenu dans l'un de ses sous-ensembles non-flous A. On a donc $S(A,B) = 1$ car A est un sous-ensemble de B. Mais on a aussi $S(B,A)$ non nul car B est une partie de

$$A \; et \; 0 < S(B,A) < 1 \tag{1.7}$$

La théorie des ensembles classiques interdit l'inégalité stricte dans 1.7. En effet soit chaque ensemble est une partie de l'autre ou alors ils sont disjoints. La théorie des ensembles flous l'autorise tout à fait.

1.2.3 Définitions de Notions sur les Ensembles Flous

1.2.3.1 Définition d'un Ensemble Flou

Définition 1.2.1: Soit un univers X, on définit un ensemble flou A dans X par l'application de μ_A de X dans l'intervalle $[0,1]$. A tout élément $x \in X$ on associe la valeur $\mu_A(x)$ telle que

$$0 \leq \mu_A \leq 1 \, (\mu_A : X \rightarrow [0,1])$$

L'application μ_A est appelée fonction d'appartenance de l'ensemble flou A.

- $\mu_A(x) = 0, x$ n'appartient pas à A.
- $\mu_A(x) = 1, x$ appartient à A il satisfait pleinement la propriété A.
- $\mu_A \neq 0, 1$ alors $\mu_A(x)$ est une valeur intermédiaire entre 0 et 1. x appartient à l'ensemble flou A. On dit que $x \in A$ avec le degrés $\mu_A(x)$

1.2.3.2 Support d'un Ensemble Flou

Définition 1.2.2: Soit un ensemble flou A dans l'univers X, le support de A noté $supp(A)$ est un sous-ensemble ordinaire de X dont chacun des éléments à une degrés d'appartenance non nul, par rapport a A.

$$supp(A) = \{x \in X; \mu_A(x) \neq 0\}$$

1.2.3.3 Noyau d'un Ensemble Flou

Définition 1.2.3: Le noyau d'un ensemble flou A de X est un sous-ensemble ordinaire de X dont chaque élément a un degrés d'appartenance égal à 1.

$$Ker(A) = \{x \in X; \mu_A(x) = 1\}$$

Lorsque le noyau n'est pas vide $Ker(A) \neq \emptyset$, on dit que l'ensemble flou A est normal ou normalisé.

1.2.3.4 Hauteur d'un Ensemble Flou

Définition 1.2.4: La hauteur d'un ensemble flou A dans X est la valeur maximum de la fonction d'appartenance : C'est la plus grand degrés d'appartenance de A.

$$H(A) = \max(\mu_A(x); x \in X)$$

1.2.3.5 Cardinal d'un Ensemble Flou

Définition 1.2.5: Le cardinal d'un ensemble flou A de support fini est égal à la somme des degrés d'appartenance des éléments de ce support :

$$Card(A) = \sum_{x \in supp(A)} \mu_A(x) \; on \; note \; aussi \; |A|$$

Dans le cas d'un support infini on a :

$$Card(A) = \int_X \mu_A(x) dm(x) \; avec \; \int_X dm(x) = 1$$

Le cardinal d'un ensemble flou A de support fini peut être défini comme un ensemble flou appelé nombre flou. Ce cardinal est noté $FCard(A)$ et vaut :

$$FCard(A) = \sum_{i \in supp(A)} frac \alpha_i i \; aussi \; noté \; \|A\|$$

1.2.4 Opérations sur les Ensembles Flous

Les opérations sur les ensembles flous sont des extensions des opérations connues sur les ensembles classiques.

1.2.4.1 Égalité

Définition 1.2.6: Soit deux ensembles flous A et B dans un univers X. On dit que A et B sont égaux ($A = B$) si leurs fonctions d'appartenance ont la même valeur en tout point x de X :

$$\mu_A(x) = \mu_B(x) \forall x \in X. Si \exists x telque \mu_A(x) \neq \mu_B(x) \Rightarrow A \neq B$$

1.2.4.2 Inclusion

Définition 1.2.7: Soit deux ensembles flous A et B dans un univers X. On dit que A est inclus dans B noté $A \subseteq B$ si leurs fonctions d'appartenance sont telles que :

$$\forall x \in X; \mu_A(x) \leq \mu_B(x)$$

L'inclusion définit une relation d'ordre.

1.2.4.3 Intersection

Définition 1.2.8: L'intersection de deux sous-ensembles flous A et B de X est la sous-ensemble de X qui contient tous les éléments x de X appartenant à la fois à A et B. L'intersection de deux sous-ensemble flous A et B $(A \cap B)$ de X est le sous-ensemble flou C tel que :

$$\forall x \in X; \mu_C(x) = \min\left(\mu_A(x), \mu_B(x)\right)$$

1.2.4.4 Union

Définition 1.2.9: L'union de deux sous-ensemble flous A et B de X est le sous-ensemble flou de X qui contient tous les éléments appartenant ou bien à A ou bien à B. L'union de deux sous-ensemble flous A et B $(A \cup B)$ de X est le sous-ensemble flou D de X tel que :

$$\forall x \in X; \mu_C(x) = \max\left(\mu_A(x), \mu_B(x)\right)$$

1.2.4.5 Propriétés de l'Union et l'Intersection

Propriété 1.2.1: $\forall A et B de X$
- associativité de $\cap$ et $\cup$
- commutativité de $\cap$ et $\cup$
- $A \cap X = A, A \cap \emptyset = \emptyset$
- $A \cup \emptyset = A, A \cup X = X$
- $A \cup B \supseteq A \supseteq A \cap B$
- $A \cap (B' \cup B'') = (A \cap B') \cup (A \cap B'')$
- $A \cup (B' \cap B'') = (A \cup B') \cap (A \cup B'')$
- $|A| + |B| = |A \cap B| + |A \cup B|$

1.2.4.6 Complément d'un Sous-Ensemble Flou

Définition 1.2.10: Soit un sous-ensemble flou A de X, son complément est le sous-ensemble contenant tous le x n'appartenant pas à A. Le complément A^C d'un sous-ensemble flou A de X est définit comme le sous-ensemble flou de X de fonction d'appartenance :

$$\forall x \in X; \mu_{A^C}(x) = 1 - \mu_A(x)$$

On peut énoncés les propriétés suivantes $A^C \cap A \neq \emptyset$ et $A^C \cup A \neq X$

1.2.4.7 Propriétés du Complément d'un Sous-Ensemble Flou

Propriété 1.2.2: $\forall A et B de X$
- Loi de Morgan : $(A \cap B)^C = A^C \cup B^C, (A \cup B)^C = A^C \cap B^C$
- $\left(A^C\right)^C = A$
- $\emptyset^C = X$
- $X^C = \emptyset$
- $|A| + |A^C| = |X|$
- $\left(supp(A^C)\right)^C = Ker(A)$
- $\left(Ker(A^C)\right)^C = supp(A)$

1.2.4.8 Propriétés des Opérateurs sur les Ensembles Flous

Propriété 1.2.3: Les choix des opérateur min pour l'intersection, max pour l'union, $complémentationà1$ pour le complément permet de préserver le structure de la théorie classique des ensembles.

1.2.5 Coupe de Niveau α ou α-coupe

Définition 1.2.11: Une coupe de niveau α ou α-coupe de l'ensemble flou A pour une valeur donnée $\alpha \in [0, 1]$ est le sous-ensemble classique A_α de X définit par

$$A_\alpha = \{x; \mu_A(x) \geq \alpha\}$$

On appelle α-niveau une α-coupe telle que $\mu_A(x) = \alpha$ pour tout $x \in A_\alpha$ Et on note $A^\alpha = \{x; \mu_A(x) = \alpha\}$ On peut en déduire que $A_0 = supp(A)$ et $A_1 = Ker(A)$.

On peut montrer qu'un ensemble flou A peut être représenté par $C(A)$ l'ensemble de toutes ses α-coupes.

Propriétés des α-coupes **Propriété 1.2.4:** Les α-coupes de A sont des parties non-floues de X emboitées par rapport à la valeur du niveau α, si $\alpha' \geq \alpha$ alors $A_\alpha \supseteq A_{\alpha'}$.

Les α-coupes des sous-ensembles flous vérifient :
- $(A \cap B)_\alpha = A_\alpha \cap B_\alpha$
- $(A \cup B)_\alpha = A_\alpha \cup B_\alpha$
- si $A \supseteq B$ alors $A_\alpha \supseteq B_\alpha$

1.2.6 Sous-Ensembles Flous Convexes

- **Définition 1.2.12:** Un sous-ensemble flou A de X des nombres réels est convexes, si pour tout couple d'éléments a et b de X, et pour tout nombre λ de $[0, 1]$ la fonction d'appartenance A vérifie :

$$\mu_A(\lambda a + (1 - \lambda)b) \geq \min\left(\mu_A(a), \mu_A(b)\right)$$

- **Définition 1.2.13:**Un sous-ensemble flou A de $\mathcal{R}$ est convexe si toutes ses α-coupes A_α sont convexes, c-à-d si, pour tout élément a et b de A_α et pour tout nombre λ de $[0, 1], x = \lambda a + (1 - \lambda)b$ appartient aussi à A_α
- **Définition 1.2.14:**Si A et B sont deux sous-ensemble flous convexes de $\mathcal{R}$ alors leur intersection est convexe.

1.2.7 Produit Cartésien de Sous-Ensemble Flou

Définition 1.2.15: Soit des sous-ensembles flous $A_1, A_2, \cdots, A_r$ respectivement définis sur $X_1, X_2, \cdots, X_r$, on définit leur produit cartésien $A = A_1 \times A_2 \times \cdots \times A_r$, comme un sous-ensemble flou de X de fonction d'appartenance :

$$\forall\ x = (x_1, x_2, \cdots, x_r)\ \in\ X, \mu_A(x) = \min\left(\mu_{A_1}(x_1), \cdots, \mu_{A_r}(x_r)\right)$$

1.2.8 Projection d'un Sous-Ensemble Flou

Définition 1.2.16: Soit un sous-ensemble flou A défini sur un univers $X_1 \times X_2$, produit cartésien des deux sous-ensembles de références X_1 et X_2.
La projection sur $X1$ du sous-ensemble flou A de $X_1 \times X_2$ est le sous-ensemble flou $Proj_{X_1}(A)$ de X_1, dont la fonction d'appartenance est définie par :

$$\forall\ x_1 \in X_1, \mu_{Prog_{X_1}(A)}(x_1) = \max_{x_2 \in X_2} \mu_A\left((x_1, x_2)\right).$$

1.2.9 Normes et Conormes Triangulaires

On peut remplacer les opérations usuelles sur les sous-ensembles flous (intersection, union ...) par d'autres opérateurs que le $\min, \max$ et complément à 1.

1.2.9.1 Définition des Normes et Conormes Triangulaires

Définition 1.2.17: Une norme triangulaire (t-norme) est une fonction T:

$$[0,1] \times [0,1] \to [0,1]$$

qui vérifie pour tout $u, v\ et\ w\ de\ [0,1]$:
- $T(u,v) = T(v,u)$ (Commutativité)
- $T\left(u, T(v,w)\right) = T\left(T(u,v), w\right)$ (Associativité)
- $T(u,v) \leq T(w,t)$ si $u \leq w$ et $v \leq t$ (Isotonie)
- $T(u,1) = u$ (Élément neutre 1)

Cas particulier : L'opérateur $T = \min$ est une norme triangulaire.

Définition 1.2.18: Une conorme triangulaire (t-conorme) est une fonction $\perp$:

$$[0,1] \times [0,1] \to [0,1]$$

qui vérifie pour tout $u, v\ et\ w\ de\ [0,1]$:
- $\perp(u,v) = \perp(v,u)$ (Commutativité)
- $\perp\left(u, \perp(v,w)\right) = \perp\left(\perp(u,v), w\right)$ (Associativité)
- $\perp(u,v) \leq \perp(w,t)$ si $u \leq w$ et $v \leq t$ (Isotonie)
- $\perp(u,0) = u$ (Élément neutre 0)

Cas particulier : L'opérateur $\perp = \max$ est une conorme triangulaire.

Propriété 1.2.5: Toute t-normes T et toutes t-conormes $\perp$ vérifient:

$$T(0,0) = 0,\ T(1,1) = 1,\ \perp(0,0) = 0,\ \perp(1,1) = 1$$

1.3 Les Variables Linguistiques et Propositions Floue

1.3.1 Les Variables Linguistiques

Une variable linguistiques sert à modéliser les connaissances imprécises ou vagues sur une variable dont la valeur précise peut être inconnue.

Définition 1.3.1: Une variable linguistique est un triplé (V, X, T_V) dans lequel V est une variable définie sur un ensemble X. L'ensemble $T_V = \{A_1, A_2, \dots\}$, fini ou infini, contient des sous-ensembles flous normalisé de X, utilisables pour caractériser V (Cf. figure 1.1).

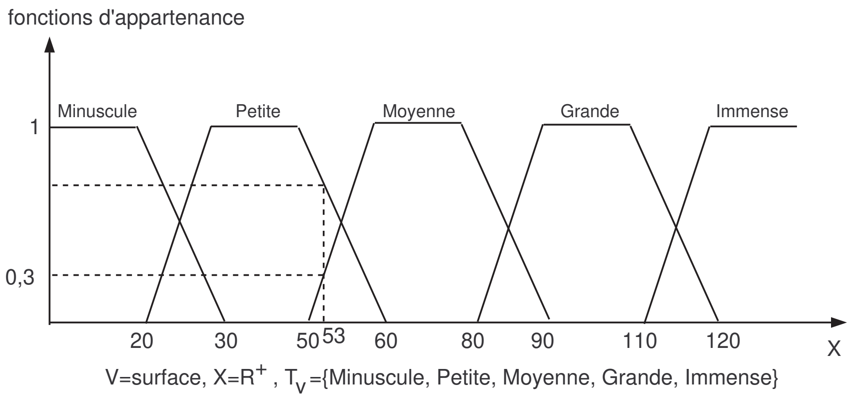

FIG. 1.1 – Exemple de variable linguistique (V, X, T_V) utilisé pour décrire la surface d'un appartement

Toutes les descriptions imprécises d'une même variable V ne peuvent être décrites par la liste T_V. Il faut donc à partir de ces données de bases, construire des caractérisations intermédiaires à l'aide de modificateurs qui atténuent ou renforcent une description donnée.

Définition 1.3.2: Un modificateur linguistique est un opérateur mod qui permet à partir de toute caractérisation floue de A de V, de produire une nouvelle caractérisation floue mod(A). Si la fonction d'appartenance de A est f_A, celle de mod(A) est $f_{mod(A)} = t_{mod}(f_A)$, obtenue par l'intermédiaire d'une transformation mathématique t_{mod} attachée à mod.

La définition d'un modificateur linguistique est générale et ne correspond pas nécessairement à une faible variation autour de la qualification initiale A. Mais l'intérêt d'un modificateur est de pouvoir engendrer des caractérisations voisines les unes de autres. Ces modificateur peuvent affaiblir ou renforcer une caractérisation.

Définition 1.3.3: Un modificateur linguistique mod est dit restrictif si la transformation associée vérifie :

$$\forall u \in [0, 1] \; t_{mod}(u) \leq u.$$

Un modificateur simple mis en évidence par L.A. ZADEH est défini par $t_{mod}(u) = u^2$. Ce modificateur préserve le support et le noyau du sous-ensemble flou et ne modifie la fonction d'appartenance de A sur les partie de X où elle est strictement comprise entre 0 et 1.
Un modificateur restrictif renforce la caractérisation à laquelle il est appliqué c-à-d $Supp\,(mod(A)) \subseteq Supp(A)$.

Définition 1.3.4: Un modificateur linguistique mod est dit expansif si la transformation associée vérifie :

$$\forall u \in [0,1]\ t_{mod}(u) \geq u.$$

Un modificateur simple mis en évidence par L.A. ZADEH est défini par $t_{mod}(u) = u^{1/2}$. Ce modificateur préserve le support et le noyau du sous-ensemble flou et ne modifie la fonction d'appartenance de A sur les partie de X où elle est strictement comprise entre 0 et 1.
Un modificateur restrictif affaiblit la caractérisation à laquelle il est appliqué c-à-d $Supp\,(mod(A)) \supseteq Supp(A)$.

1.3.2 Propositions Floues

1.3.2.1 Propositions Floues Élémentaires

Définition 1.3.5: Étant donné une ensemble L de variables linguistiques et un ensemble M de modificateurs, une proposition floue élémentaire est définie à partir d'une variable linguistique (V, X, T_V) de L par la qualification "V est A", où A est une caractérisation floue appartenant à T_V ou à $M(T_V)$.

1.3.2.2 Propositions Floues Générales

Définition 1.3.6: Une proposition floue générale est obtenue par l'utilisation conjointe de propositions floues élémentaires.

On peut construire des propositions floues par conjonction, disjonction ou implication portant sur des propositions floues quelconques.

Définition 1.3.7: Une règle floue est une proposition floue de la forme "si p alors q" utilisant une implication entre deux propositions floues quelconques p et q

Les règles floues les plus utilisées sont de la forme "Si V est A et U est B et $\ldots$ alors W est C", où "V est A","U est B", $\ldots$ sont les prémisses de la règle et "W est C" sa conclusion.

1.3.2.3 Distribution de Possibilité Associée à une Proposition Floue

Définition 1.3.8: Étant donnée une proposition floue élémentaire "V est A", la distribution de possibilité $\pi_{V,A}$ qui lui est associée est définie sur X à partir de la fonction d'appartenance f_A de A par :

$$\forall x \in X\ \pi_{V,A}(x) = f_A(x)$$

$\pi_{V,A}$ ainsi définie est une distribution de possibilité.

Définition 1.3.9: Un quantificateur flou est un sous-ensemble flou Q de l'ensemble $\mathbb{R}$ des nombres réels qui décrit un nombre de cas approximatif, ou un sous-ensemble flou de $[0,1]$ qui décrit une proportion approximative.

Définition 1.3.10: Soit un quantificateur flou Q, de fonction d'appartenance f_Q. Une proposition quantifiée est de la forme "QW est B", pour une proposition floue "Q est B" définie à partir d'une variable linguistique (W, X, T_W). La valeur de vérité de la proposition floue quantifiée "QW est B" est égale à la valeur de $f_Q(p)$.

1.3.3 Raisonnement en Logique Floue

La logique floue permet le traitement souple de connaissances imprécises ou incertaines, ce qui serait impossible avec le logique classique.

La première tentative pour assouplir la logique classique vient de Lukasiewicz. C'est une logique à trois valeurs de vérité :

- 0 pour le faux
- 1 pour le vrai
- 1/2 pour le doute

D'autres logiques multivalentes ont ensuite été développées.

On peut considérer que la logique floue est une extension de la logique classique. Les propositions sont des propositions floues définies à partir d'un ensemble L de variables linguistiques (V, X, T_V) et d'un ensemble M de modificateurs. Leurs valeurs de vérité appartient à tout l'intervalle $[0,1]$ et elle est fournie par la fonction d'appartenance de la caractérisation floue utilisée dans la proposition floue.

1.3.3.1 Implication Floue

Définition 1.3.11: L'implication floue entre deux propositions floues élémentaires "V est A" et "W est B" est une proposition floue concernant le couple de variables (V, W), dont la valeur de vérité est donnée par la fonction d'appartenance f_R d'une relation floue R entre X et Y définie, pour tout (x, y) de $X \times Y$ par :

$$f_R(x,y) = \Phi\left(f_a(x), f_b(y)\right) \ \textit{pour une fonction}\ \Phi\ \textit{de}\ [0,1] \times [0,1] \rightarrow [0,1]$$

1.3.3.2 Modus Ponens Généralisé

Le modus ponens de la logique classique ne permet d'obtenir une conclusion à la seule condition que la proposition soit exactement vérifiée. Il convient donc d'adapter cette forme de raisonnement à nos ensembles flous. Il convient donc d'utiliser le modus ponens généralisé énoncé sous la forme :

Règles floue	$(R1)$	si V est	A	alors W est	B
Fonction d'appartenance			f_A		f_B
Fait observé		V est	A'		
Fonction d'appartenance			$f_{A'}$		
Conclusion				W est	B'
Fonction d'appartenance					$f_{B'}$

TAB. 1.1 – Condition d'utilisation du modus ponens généralisé

Définition 1.3.12: En présence d'une règle de la forme "si V est A alors W est B", et d'une observation de la forme "V est A'", la conclusion définie par le modus ponens généralisé est la proposition floue "W est B'", B' ayant la fonction d'appartenance suivante :

$$\forall y \in Y \ f_{B'} = sup_{x \in X} T\left(f_{A'}(x), f_R(x,y)\right)$$

pour une t-norme T qui est un opérateur de modus ponens généralisé.

1.4 La Commande Floue

1.4.1 Introduction

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

1.4.2 Caractéristiques de la Commande Floue

Son but est de traiter les problèmes de commande de processus avec une approche différente de l'automatique classique. Le plus souvent elle se sert de la connaissance des experts du processus. La spécificité de la commande floue réside dans les points suivants :

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

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

Intérêt de la commande floue :

- La commande est simple à réaliser, facilement adaptable aux conditions de fonctionnement. La plupart du temps un petit nombre de règle suffit à décrire le système.
- La coordination de plusieurs objectifs est possible (système multivariable).
- Cette commande est reconnue comme robuste.

1.4.3 Le Contrôleur Flou

Le contrôleur flou peut être considéré comme un système expert simple Cf. figure 1.2: La représentation des connaissances est basée sur les ensembles flous dont les règles fournissent directement les conclusions. L'interface avec le flou assure la conversion des variables linguistiques du système en variables floues compréhensibles par le régulateur (Fuzzyfication). Le contrôleur flou, grâce à sa base de règle génère alors la conclusion qui sera rendu compréhensible pour le système par l'interface non floue (Défuzzyfication).

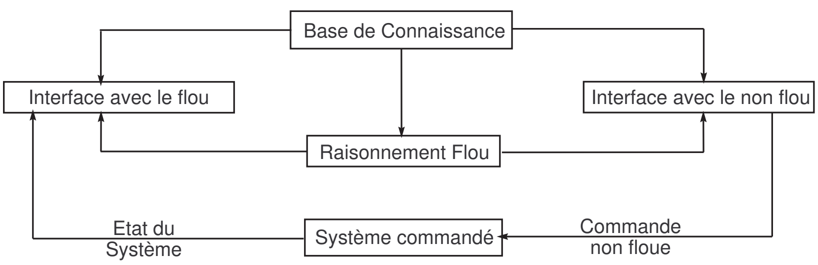

FIG. 1.2 – Configuration générale d'un contrôleur flou

Le contrôleur (ou régulateur)flou peut être représentée de la façon suivante (Cf. figure 1.3):

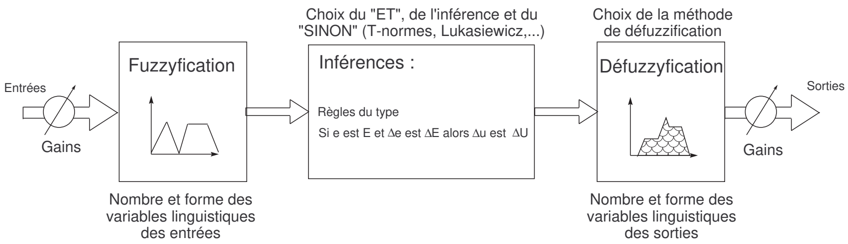

FIG. 1.3 – Schéma d'un régulateur flou basé sur le raisonnement flou par inférences

1.4.4 Principe de la Commande Floue

Le principe consiste à caractériser le mieux possibles les variables qui agissent sur le processus en fonction de tous les états du processus à commander et des avis et recommandations des experts.

1.4.4.1 Formalisation

Les connaissances du processus sont décrites à partir des variables linguistiques $(V_1, X_1, T_1), (V_2, X_2, T_2)$ définies sur un univers X_1, X_2 correspondant aux variables d'entrées du processus.
Les connaissances sont ensuite sous formes de règles de la forme :
- (R1) si V_1 est A_{11} et V_2 est A_{12} alors W est B_1
- (R2) si V_1 est A_{21} et V_2 est A_{22} alors W est B_2

ou les A_{ij} sont des sous-ensembles flous des univers X_i appartenant aux caractérisations floues T_i et représentant une partition floue des valeurs possibles des variables floues V_i et ou les sous-ensembles flous B_i des univers Y_i appartenant aux caractérisations T_w représentent les valeurs possibles prise par la variables floue de sortie W.

On peut remarquer que la partition de l'univers est plus ou moins fin suivant le nombre de sous-ensembles flous.

Avec la donnée de l'état du système (le plus souvent fournie par des capteurs) on détermine une caractérisation non floue de la variable de commande en sortie du régulateur.

Cette réalisation se fait en trois étapes :

1. Pour chaque règle on utilise la caractérisation de l'état du système pour obtenir un résultat intermédiaire.
2. Agrégation des résultats intermédiaires.
3. Déduction de l'action non floue du résultat agrégé.

1.4.4.2 La Commande Floue, un Approximateur Universel

Plusieurs chercheurs ont montrés que la commande floue peut être considérée comme un approximateur universel [KOSKO, 1992a]. Alors pour une fonction $W = f(V_1, V_2, \ldots, V_n)$ et un niveau d'approximation ϵ il est possible de construire un ensemble fini de règles $R_1, R_2, \ldots, R_n$ du type

- (R1) si V_1 est A_{11} et V_2 est A_{12} alors W est B_1
- (R2) si V_1 est A_{21} et V_2 est A_{22} alors W est B_2
- etc.

qui permet d'approcher f à ϵ près, quand f est une fonction continue définie sur un compact de dimension finie. Le théorème suivant [WANG, 1995] montre que les systèmes flous sont capable d'approximer n'importe quelle fonction non-linéaire sur U avec n'importe quel précision si U est compacte.

Théorème 1.4.1. *Pour toute fonction continue réelle g sur un ensemble compact $U \subset \mathbb{R}^n$ et ϵ arbitraire (assez petit), il existe un système flou f de la forme :*

$$f(x) = \frac{\sum_{l=1}^{M} \bar{y}_l \left[\prod_{i=1}^{n} a_{l,i} \exp\left(-\left(\frac{x_i-\bar{x}_{l,i}}{\sigma_{l,i}}\right)^2\right)\right]}{\sum_{l=1}^{M} \left[\prod_{i=1}^{n} a_{l,i} \exp\left(-\left(\frac{x_i-\bar{x}_{l,i}}{\sigma_{l,i}}\right)^2\right)\right]} \tag{1.8}$$

tel que :

$$\sup_{x \in U} |f(x) - g(x)| < \epsilon$$

où $\bar{y}_l, a_{l,i}$ et $\sigma_{l,i}$ sont les paramètres ajustables du système flou sous les contraintes suivantes : $\bar{y}_l \in V, a_{l,i} \in [0,1], \bar{x}_{l,i} \in U_i,$ *et* $\sigma_{l,i} > 0$.

On peut étendre ce théorème au cas discret.

1.4.5 Méthodes de Calcul de la Commande

Il existe plusieurs méthodes de calculs de la commande floue de processus. La première d'entre elle vient directement des méthodes d'inférence des systèmes experts.

1.4.5.1 Méthode Logique

Obtention du Résultat Intermédiaire La première étape consiste à effectuer un modus ponens généralisé, donc de choisir l'une des implications floues R. La partie condition des règles étant complexe, (plusieurs variables floues $V_1, V_2, \dots$) on doit choisir un opérateur de conjonction pour définir le sous-ensemble flou conjoint associé à la partie de condition de la règle R_i. Cette étape fournit une caractérisation floue B'_i de la variable de commande V_i pour chaque Règle R_i.

Agrégation des Résultats Intermédiaires Chacune des règles R_i donnent une caractérisation de la variable de commande W en présence d'une donnée d'entrée précise. Il faut faire la synthèse de ces caractérisations pour parvenir à une décision. Pour cela on choisit un opérateur d'agrégation pour obtenir une caractérisation floue générale à partir des résultats intermédiaires.Ces opérateurs sont le plus souvent la t-norme (respectivement la t-conorme) définies par ZADEH, donnant un résultat agrégé de fonction d'appartenance :

- $\forall\ y \in Y\ f_{A(B'_1,B'_2,\dots)}(y) = \max_{i=1,2,\dots} f_{B'_i}(y)$

respectivement

- $\forall\ y \in Y\ f_{A(B'_1,B'_2,\dots)}(y) = \min_{i=1,2,\dots} f_{B'_i}(y)$

Calcul du Résultat Non-Flou ou Défuzzification Il faut maintenant déduire une action de commande du résultat agrégé flou obtenu par l'opération précédente. Pour déterminer cette valeur y_0 plusieurs méthodes sont disponibles mais la plus couramment utilisée est la méthode du centre de gravité donnée par :

$$y_0 = \frac{\int_Y f_{A\left(B'_1, B'_2, \ldots\right)}(y)\ y\, dy}{f_{A\left(B'_1, B'_2, \ldots\right)}(y)\ \ dy}$$

en continu et

$$y_0 = \frac{\sum_{y_j \in Y} f_{A\left(B'_1, B'_2, \ldots\right)}(y_j)\ y_j}{\sum_{y_j \in Y} f_{A\left(B'_1, B'_2, \ldots\right)}(y_j)}$$

en échantillonné.

D'autres méthodes sont utilisables :

- ∘ Le maximum, dans le cas où $\max(Y)$ est réduit à un élément avec $\max(Y) = \{y_0\}$
- ∘ Le maximum moyen $y_0 = \frac{\left[\sum_{y_j \in \max(Y)} y_j\right]}{|\max(Y)|}$
- ∘ Le milieu des maxima $y_0 = (y' + y'')/2$ avec y' et y'' respectivement le plus petit et le plus grand des éléments de $\max(Y)$
- ∘ Le premier maximum y'

Certaines méthodes permettent de réaliser en même temps les étapes 2 et 3 en choisissant pour représentant de l'ensemble des résultats intermédiaires la moyenne pondérée des représentants des résultats intermédiaires B'_i obtenus par chacune des règles R_i.

- ∘ La méthode des hauteurs $y_0 = [\sum_i y_i h(B'_i)] / [\sum_i h(B'_i)]$
- ∘ La méthode des aires $y_0 = [\sum_i y_i S(B'_i)] / [\sum_i S(B'_i)]$

1.4.5.2 Méthode de Mandani et de Larsen

Cette méthode est historiquement la première à avoir été proposée et est assez simple à mettre en œuvre. Elles peuvent être regardée comme des méthodes logiques particulières reposant sur le choix de la fonction d'agrégation

- ∘ la t-norme de ZADEH pour la méthode de Mandani
- ∘ la t-norme probabiliste pour la méthode de Larsen

Avec la méthode de Mandani le résultat intermédiaire est obtenu par

$$\forall\ y\ \in\ Y, f_{B'_1}(y) = \min\left[\min\left(f_{11}(x_1), f_{12}(x_2)\right), f_1(y)\right]$$

et par la méthode de Larsen

$$\forall\ y\ \in\ Y, f_{B'_1}(y) = \min\left[\min\left(f_{11}(x_1).f_{12}(x_2)\right).f_1(y)\right]$$

1.4.5.3 Méthode par Interpolation

Cette méthode est complètement différente et est seulement utilisable dans le cas où les fonctions d'appartenance caractérisant les variables sont des fonctions monotones. Cette monotonie permet de procéder par interpolation entre les valeurs de Y obtenues grâce aux diverses règles R_i. Ainsi pour une observation telle que $V_1 = x'_1, V_2 = x'_2$ on utilise les valeurs suivantes :

$$\alpha_1 = \min\left(f_{11}(x_1), f_{12}(x_2)\right),\ \alpha_2 = \min\left(f_{21}(x_1), f_{22}(x_2)\right)\ \ldots$$

et les points $y_1, y_2, \ldots$ tels que :

$$\alpha_1 = f_1(y_1), \alpha_2 = f_2(y_2), \ldots$$

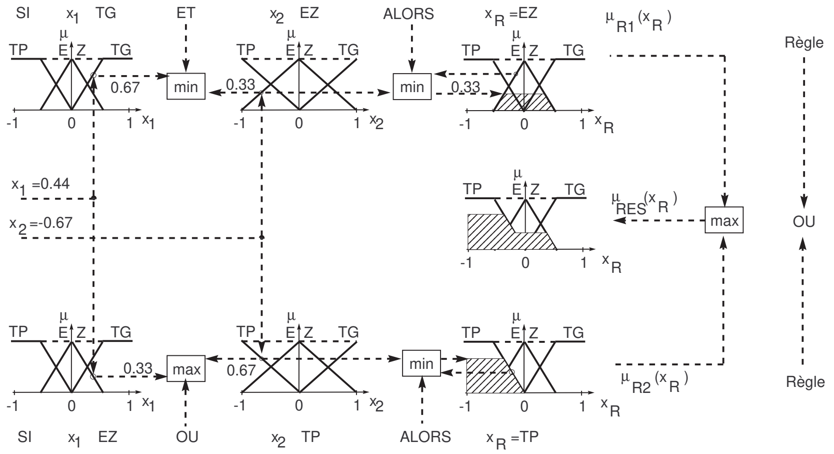

FIG. 1.4 – Représentation graphique de la méthode de Mandani

1.4.5.4 Méthode de Takagi et Sugeno

La plupart des implications floues utilisent des variables floues avec fonctions d'appartenances monomodales. L'utilisation de l'agrégation par inférence des règles ne permet pas d'avoir de bon résultat dans la commande de système multivariable. Il faut alors augmenter considérablement le nombre de variables floues et donc le nombre de règles.

L'implication floue présentée par Takagi et Sugeno est basée sur la partition floue de l'ensemble des entrées. Dans chaque sous-ensemble flou une équation linéaire lie les entrées et les sorties. La sortie non floue de ce raisonnement flou est donnée par l'agrégation des valeurs floues résultantes des règles utilisées par les entrées.

Forme des Règles Floues Utilisées par Takagi & Sugeno Soit $\mu_A(x)$ la fonction d'appartenance à un sous-ensemble flou A avec $x \in X$ univers du discours. On définit la valeur réelle de "x est A et y est B" par l'expression suivante :

$$|x\ est\ A\ et\ y\ est\ B| = A(x) \wedge B(y)$$

La règle floue R est de la forme :

$$R : si\ f(x_1\ est\ A_1, \ldots,\ x_k\ est\ A_k)\ alors\ y = g(x_1, \ldots, x_k)$$

où

- y est la variable de conclusion
- $x_1, \ldots, x_k$ Variables de prémices utilisées aussi dans la conclusion
- $A_1, \ldots, A_k$ Ensembles flous avec fonctions d'appartenances linéaires représentant un sous-espace flou dans lequel la règle R peut être vue comme le raisonnement
- f fonction logique qui lie les propositions dans les prémices
- g fonction qui calcule la valeur de y lorsque $x_1, \ldots, x_k$ vérifient les prémices

Dans le cas de Takagi & Sugeno la règle R s'écrit :

$$Si\ x_1\ est\ A_1\ et\ \ldots\ x_k\ est\ A_k\ alors\ y = p_0 + p_1x_1 + \ldots + p_kx_k$$

Ensuite la valeur réelle finale de la sortie y inférée depuis n règles est calculée comme la moyenne de tous les sorties de chaque règle y^i avec les poids $\omega^i = \prod \mu_i(x_i^0)$ avec $\mu(x)$

fonction d'appartenance du sous-ensemble flou A_i :

$$y = \frac{\sum \omega^i y^i}{\sum \omega^i}$$

L'approximation linéaire faite par les règles R^i prises séparément, avec le raisonnement flou, ces équations linéaires liant entrées et sorties permettent de passer de façon continue de l'une des équations à l'autre. Ceci est aussi un moyen de réduire le nombre de règles.

Exemple d'Utilisation de Règles Floue T & S Soit les trois règles floues suivantes :

$R1$	Si x_1 est $petit_1$ et x_2 est $petit_2$	alors	$y = x_1 + x_2$
$R2$	Si x_1 est $grand_1$	alors	$y = 2 \times x_1$
$R2$	Si x_2 est $grand_2$	alors	$y = 3 \times x_2$

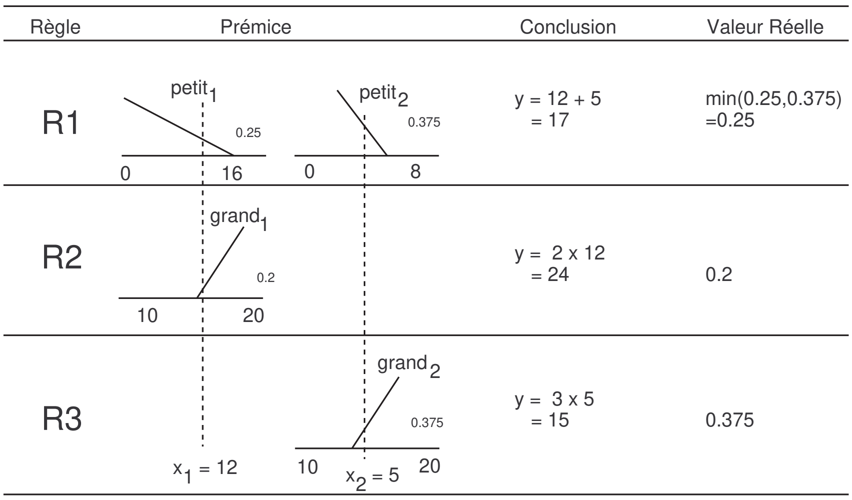

FIG. 1.5 – Représentation de la méthode de Takagi & Sugeno

et les valeurs d'entrées $x_1 = 12, x_2 = 5$ alors la valeur résultante des inférences des règles floues est Cf. 1.5:

$$y = \frac{0.25 \times 17 + 0.2 \times 24 + 0.375 \times 15}{0.25 + 0.2 + 0.375} = 17.8$$

1.5 Apprentissage des Systèmes Flous

1.5.1 introduction

Pour tous les systèmes complexes qui utilisent des données numériques, la connaissance *a priori* du système par concertation avec les experts donne une bonne base pour la définition du contrôleur flou, mais il peut être très intéressant d'ajuster les paramètres de ce dernier afin d'obtenir de meilleurs résultats. On parle alors d'apprentissage ou entraînement de systèmes flous. La suite de cette section donne un série d'algorithme d'adaptation pour les systèmes flous.

1.5.2 Apprentissage de Systèmes Flou par Rétropropagation

1.5.2.1 Introduction

Au départ l'algorithme de rétropropagation était appliqué à l'adaptation des poids des réseaux de neurones artificiels multicouches. L'idée est donc de représenter le système flou comme un réseau de neurones et de lui appliquer l'algorithme [WANG, 1995].

Avec la forme floue de l'équation 1.8, on peut représenter le système flou par un réseau de neurones à trois couches comme le montre la figure 1.6.

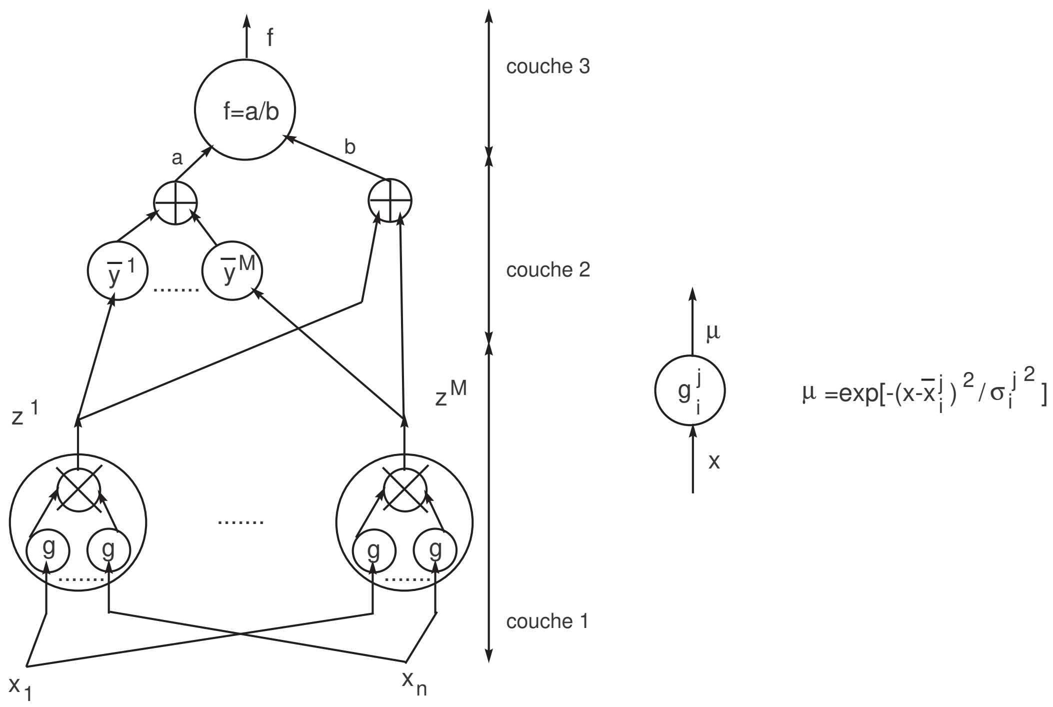

FIG. 1.6 – Modélisation neuronale d'un système flou

1.5.2.2 Rétropropagation pour les Systèmes Flous

Soit une paire d'entrées/sorties $(x_p, d_p), x_p \in U \subset \mathbb{R}^n, d_p \in V \subset \mathbb{R}$. On doit alors définirr un système flou $f(x)$ du type 1.8 tel que

$$e_p = \frac{1}{2}[f(x_p) - d_p]^2 \tag{1.9}$$

soit minimisé.

On suppose que M est connu et $a_{l,i} = 1$, le problème devient alors l'apprentissage des paramètres $\bar{y}_l, \bar{x}_l, \sigma_{l,i}$ qui minimise e_p de 1.9. Afin d'ajuster les paramètres $\bar{y}_l, i$ on utilise la relation suivante :

$$\bar{y}_l(k+1) = \bar{y}_l(k) - \alpha \frac{\partial e}{\partial \bar{y}_l(k)}|_k \tag{1.10}$$

où $l = 1, 2, \ldots, M$ et $k = 1, 2, \ldots$ et α est le pas d'apprentissage.

D'après la figure 1.6 on a $f = a/b$ et $a = \sum_{l=1}^{M}(\bar{y}_l z_l), b = \sum_{l=1}^{M} z_l$ et $z_l = \prod_{i=1}^{n} \exp\left(-\left(\frac{x_i - \bar{x}_i^j}{\sigma_i^j}\right)^2\right)$

La loi d'adaptation devient alors :

$$\bar{y}_l(k+1) = \bar{y}_l(k) - \alpha \frac{f-d}{b} z_l \tag{1.11}$$

De la même façon on utilise les équations suivantes pour ajuster les paramètres de $\bar{x}_{l,i}$ et $\sigma_{l,i}$

$$\bar{x}_{l,i}(k+1) = \bar{x}_{l,i}(k) - \alpha \frac{f-d}{b}(\bar{y}_l - f) z_l \frac{2(x_{i,p} - \bar{x}_{l,i}(k))^2}{\sigma_{l,i}^2(k)} \tag{1.12}$$

$$\bar{\sigma}_{l,i}(k+1) = \bar{\sigma}_{l,i}(k) - \alpha \frac{f-d}{b}(\bar{y}_l - f) z_l \frac{2(x_{i,p} - \bar{x}_{l,i}(k))^2}{\sigma_{l,i}^3(k)} \tag{1.13}$$

La mise à jour des paramètres $\bar{x}_{l,i}$ et $\bar{\sigma}_{l,i}$ se font suivant les équations 1.12 et 1.13.

1.5.2.3 Conclusion

Cet algorithme peut être appliqué pour l'identification de système non linéaire dynamique. En effet comme les systèmes flous sont des approximateurs universelles et que l'on possède maintenant l'algorithme de rétropropagation, par analogie avec les réseaux de neurones artificiels il est possible comme ces derniers d'approximer n'importe quelle système. L'intérêt par rapport au réseaux de neurones est que l'identification floue des paramètres à une signification physique. Les $\bar{y}_{l,i}$ sont les centres des conclusions floues. Les $\bar{x}_{l,i}$ et $\sigma_{l,i}$ sont les centres et rayons des prémices floues. De plus la construction d'un modèle floue initial est basé sur les connaissances des experts, ce qui permet d'intégrer des données propres au système et donc lors de la phase d'apprentissage d'accélérer la convergence de l'algorithme.

1.5.3 Moindres Carrés Orthogonaux pour les Systèmes Flous

1.5.3.1 Introduction

Le principal inconvénient de la rétropropagation du gradient de l'erreur est que l'algorithme peut se bloquer dans des minima locaux lors de la phase d'apprentissage et sa convergence est assez lente (si le pas de calcul est très petit). Ici On adapte le système flou en le rendant équivalent à une série de fonctions à base floue(FBF). Ce développement en série de FBFs permet d'utiliser l'algorithme des moindres carrés orthogonaux (OLS : Orthogonal Least Square)car le système est alors linéaire par rapport à ses paramètres ajustables. Cet algorithme est plus rapide que la rétropropagation et ne nécessite qu'une seule itération.

1.5.3.2 Du Système Flou à la Série de Fonctions à Base Floue

Soit le système flou définit par l'équation 1.8 on définit une fonction à base floue par la relation :

$$p_i(x) = \frac{\prod_{i=1}^{n} \mu_{F_i^j}(x_i)}{\sum_{j=1}^{M} \prod_{i=1}^{n} \mu_{F_i^j}(x_i)}, \; j = 1, 2, \ldots, M \tag{1.14}$$

où $\mu_{F_i^j}(x_i) = a_{i,j} \exp\left(-\frac{1}{2}\left(\frac{x_i - \bar{x}_{i,j}}{\sigma_{i,j}}\right)^2\right)$ sont des fonctions d'appartenances gaussiennes. Le système flou s'écrit alors :

$$f(x) = \sum_{j=1}^{M} p_j(x)\Theta_j \tag{1.15}$$

où les $\Theta_j = \bar{y}_j \in \mathbb{R}$ sont des constantes.

D'après 1.14 le système flou peut être vu comme non linéaire par rapport à ses paramètres $\sigma_{i,j}, a_{i,j}, \bar{x}_{i,j}$ et alors on doit utiliser des algorithmes d'optimisation non linéaire comme la rétropropagation. Mais d'un autre côté si l'on fixe certains paramètres comme les centres et les rayons des fonctions FBFs on obtient un système linéaire par rapport à ses paramètres suivant l'équation 1.15. L'avantage de ce point de vue est qu'il permet d'utiliser l'algorithme des moindres carrés orthogonaux très puissant pour l'estimation des paramètres.

1.5.3.3 Apprentissage par Moindres Carrés Orthogonaux

Le développement en fonction à bases floues peut être vu comme un cas particulier d'un modèle de régresseur linéaire du type :

$$d(t) = \sum_{j=1}^{M} p_j(t)\Theta_j + e(t) \tag{1.16}$$

où $d(t)$ est la sortie du système, Θ_j les paramètres réels du système flou et $p_j(t)$ les régresseurs qui sont des fonctions fixées par rapport aux entrées $x(t)$ et $e(t)$ est l'erreur entre le système réel et le système identifié.

Si l'on applique N paires d'entrées/sorties aux systèmes décrit dans l'équation 1.16 on obtient la forme matricielle suivante

$$D = P\Theta + E \tag{1.17}$$

où $D = [d(1)\ \dots\ d(N)]^T, P = [p_1\ \dots\ p_n]$ *avec* $p_i = [p_i(1), \dots, p_i(N)]^T, \Theta = [\Theta_1\ \dots\ \Theta_N]^T, E = [e(1)\ \dots\ e(N)]^T$.

Afin d'exécuter l'algorithme OLS sur notre équation 1.17, il faut commencer par fixer les paramètres $\sigma_{i,j}, a_{i,j}$ et $\bar{x}_{i,j}$ dans les FBFs qui caractérisent $p_j(x)$.

On choisit pour cela très simplement tous les $a_{i,j} = 1$ car les fonctions d'appartenances du système flou sont supposées avoir la valeur maximale 1 en leurs centres.

Les $\bar{x}_{i,j}$ centres des FBFs sont initialisées aux premières valeurs des entrées c-à-d $\bar{x}_{i,j} = x_{i,0}$ et les rayons des FBFs sont initialisés aux valeurs $\sigma_{i,j} = [\max(x_{i,0}(j),\ j = 1, 2, \dots, N) - \min(x_{i,0}(j),\ j = 1, 2, \dots, N)]/M_S$ où $i = 1, 2, \dots, n,\ j = 1, 2, \dots, N$ et M_S est le nombre de FBFs qui est le plus souvent défini par les contraintes ($M << N$).

Ensuite après cette phase d'initialisation il suffit d'appliquer l'algorithme OLS pour sélectionner les fonction FBFs les plus représentatives de N fonction FBFs déterminées par la phase d'initialisation.

- A la première itération, pour $1 \leq i \leq N$ on calcul

$$\omega_1^{(i)} = p_i,\ g_1^{(i)} = \left(\omega_1^{(i)}\right)^T d_0 \ / \left(\left(\omega_1^{(i)}\right)^T \omega_1^{(i)}\right) \tag{1.18}$$

$$[err]_1^i = \left(g_1^{(i)}\right)^2 \left(\omega_1^{(i)}\right)^T \omega_1^{(i)} / \left(d_0^T d_0\right) \tag{1.19}$$

où $p_i = [p_i(x_0(1)), \ldots, p_i(x_0(N))]^T$ et les $p_i(x_0(t))$ sont donné par la méthode de détermination de FBF initiales. On cherche ensuite

$$[err]_1^{i_1} = \max\left([err]_1^i, 1 \leq i \leq N\right) \tag{1.20}$$

et on sélectionne

$$\omega_1 = \omega_1^{i_1} = p_{i_1},\ g_1 = g_1^{i_1} \tag{1.21}$$

- A l'étape k avec $2 \leq k \leq M_s$, pour $1 \leq i \leq N, i \neq i_1, \ldots, i \neq i_{k-1}$ on calcul

$$\alpha_{jk}^{(i)} = \omega_j^T p_i / \left(\omega_j^T \omega_j\right),\ 1 \leq j \leq k \tag{1.22}$$

$$\omega_k^{(i)} = p_i - \sum_{j=1}^{k-1} \alpha_{jk}^{(i)} \omega_j,\ g_k^{(i)} = \left(\omega_k^{(i)}\right)^T\ d_0 / \left(\left(\omega_k^{(i)}\right)^T \omega_k^{(i)}\right) \tag{1.23}$$

$$[err]_k^{(i)} = \left(g_k^{(i)}\right)^2 \left(\omega_k^{(i)}\right)^T\ \left(\omega_k^{(i)}\right) / \left(d_0^T d_0\right) \tag{1.24}$$

On cherche alors

$$[err]_k^{i_k} = \max\left([err]_k^{(i)}, 1 \leq i \leq N,\ i \neq i_1, \ldots, i \neq i_{k-1}\right) \tag{1.25}$$

et on sélectionne

$$\omega_k = \omega_k^{(i_k)},\ g_k = g_k^{(i_k)} \tag{1.26}$$

On résout le système triangulaire suivant

$$A^{(M_S)} \Theta^{(M_S)} = g^{(M_S)} \tag{1.27}$$

où

$$A^{(M_S)} = \begin{bmatrix} 1 & \alpha_{12}^{(i_2)} & \alpha_{13}^{(i_3)} & \ldots & \alpha_{1M_s}^{(i_{M_S})} \\ 0 & 1 & \alpha_{23}^{(i_3)} & \ldots & \alpha_{2M_s}^{(i_{M_S})} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 & \alpha_{M_S-1\ M_S}^{(i_{M_S})} \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix} \tag{1.28}$$

avec

$$g^{(M_S)} = [g_1, \ldots, g_{M_S}]^T,\ \Theta^{(M_S)} = [\Theta_1^{(M_S)}, \ldots, \Theta_{M_S}^{(M_S)}]^T \tag{1.29}$$

Le développement en FBF final est donné par

$$f(x) = \sum_{j=1}^{M_S} p_{i_j}(x) \Theta^{(M_S)} \tag{1.30}$$

où $p_{i_j}(x)$ est le sous ensemble final des FBFs séléctionnées par l'algorithme OLS.

1.5.3.4 Conclusion

On peut donc représenter un système flou comme un combinaison linéaire de fonctions à bases floues et avec l'algorithme des moindres carrés orthogonaux déterminer un contrôleur efficace.

1.5.4 Apprentissage par Base de Règles Floues

1.5.4.1 Introduction

Ici l'algorithme développé consiste à définir un système floue adaptatif basé sur des paires de données d'entrées/sorties et des règles floues linguistiques de type "Si-Alors". L'idée principale consiste à générer des règles floues à partir des paires d'entrées/sorties et de les combiner avec les règles linguistiques dans un base de règle.

1.5.4.2 Création des Règles Floues avec les Données Numériques

La tâche consiste donc à transformer des données numériques en données floues sous la formes de règles. L'approche de transformation des données se décompose en cinq étapes.

Étape 1: Division des espaces d'entrées et de sorties en régions floues

On suppose que l'on peut bornés les domaines d'appartenance des variables d'entrées et sorties sur des intervalles du types $[x_i^-, x_i^+]$ pour les entrées et $[y_i^-, y_i^+]$ pour les sorties. On divise alors chacun des ces intervalles en $2N+1$ régions (N peut être différent pour chaque variable) Cf. figure 1.7.

Le choix de la fonction d'appartenance autres que des triangles est tout à fait possible.

Étape 2: Création des règles floues à partir des données Tout d'abord il faut déterminer les valeurs d'appartenance de chaque variable d'entrées $x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}$ et de sorties $y_1^{(i)}, y_2^{(i)}, \ldots, y_n^{(i)}$ dans chaque région.

Ensuite on fixe pour variable x_i, y_j ont la région qui lui correspond le mieux en terme de valeur d'appartenance.

Finalement on obtient un règle pour chaque paire d'entrée/sortie du type :

Règle 1 : Si x_1 est A_1 et x_2 est B_1 Alors y est U_1.

Étape 3: Détermination de la valeur d'appartenance de chaque règle Comme chaque paire de données entraîne la création de d'une règle il est fort probable que des règles contradictoires apparaissent, il faut donc créer une sorte de compétition entre les règles en leur affectant une valeur d'appartenance. Cette valeur d'appartenance est définie de la manière suivante : Pour une règle R1 du type "Si x_1 est A et x_2 est B Alors y est C" on à la valeur suivante :

$$D(R1) = \mu_A(x_1)\mu_B(x_2)\mu_C(y)$$

Lorsque deux règles rentrent en concurrence on choisit celle qui possède la plus grande valeur d'appartenance.

Étape 4: Création de la base de règle combinée Il faut donc maintenant remplir le tableau de la bas de règle : pour cela il faut maintenant prendre en compte les règles linguistiques décrites par les experts (ce type de règle possède aussi une valeur d'appartenance qui ici serait plus une valeur de croyance aux dires des experts). Si pour une même conclusion deux règles entre en concurrence (celle crée par les donnée numériques et celle extraite des dires des experts) on choisit celle qui possède la plus grande valeurs d'appartenance.

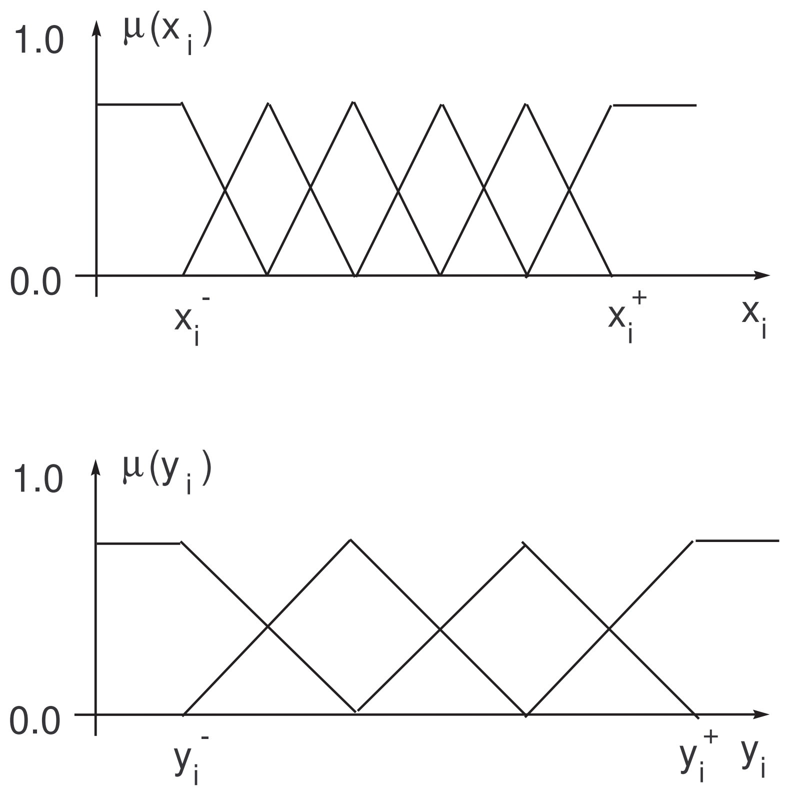

FIG. 1.7 – Partitions des variables d'entrée et de sorties en régions floues

On voit bien ici que les deux types d'information sont codées dans la même base de règle.

Étape 5: Création d'une cartographie basée sur la base de règle Afin de déterminer la valeur réelle de la sortie du système floue on utilise la défuzzification suivante :

- Premièrement pour une paires de données d'entrées/sorties, on combine les prémices de la règle i en utilisant le produit pour obtenir le degrés d'appartenance de la sortie $\mu^i_{O^i} = \mu_{I^i_1}(x_1)\mu_{I^i_2}(x_2)$où O^i représente la région de la sortie de la règle i et I^i_j représente les régions des des entrées de la règles i pour le j^{e} composant.
- La formule de défuzzification est un moyenne centrée :

$$y = \frac{\sum_{i=1}^{M} \mu^i_{O_i}\bar{y}_i}{\sum_{i=1}^{M} \mu^i_{O_i}}$$

  où $\bar{y}_i$ représente le centre de la région O_i.

1.5.4.3 Conclusion

Cette procédure en cinq étapes peut être facilement étendue au cas multivariable. Avec cette méthode il est possible de codé ensemble sous la forme d'un système flou unique des données différentes. L'avantage principale de cette méthode est la simplicité de mise en œuvre

1.5.5 Apprentissage de Système Flou par Agrégation avec le Plus Proche Voisin

1.5.5.1 Introduction

Cette méthode permet la configuration d'un système flou de façon optimal par l'apprentissage de ces paires d'entrées/sorties à n'importe quel degrés d'acuité. L'idée est très simple, au départ on considère le système flou composé d'autant de règle floue qu'il y a de paires d'entrées-sorties.

Ensuite il faut agrégé les données échantillonnées entre elles par la méthodes des plus proches voisins. Il suffit ensuite de choisir le meilleur des systèmes flous pour ce groupe de données. On appelle cela aussi système flou adaptatif.

1.5.5.2 Système Flou Optimal

On suppose que pour N paires d'entrées/sorties (avec N assez petit), on peut construire un système floue qui copie les paires d'entrées/sorties à la précision ϵ. Ce système flou est donné par l'équation 1.31 :

$$f(x) = \frac{\sum_{l=1}^{N} y_l \exp\left(-\frac{|x-\bar{x}_l|^2}{\sigma^2}\right)}{\sum_{l=1}^{N} \exp\left(-\frac{|x-\bar{x}_l|^2}{\sigma^2}\right)} \tag{1.31}$$

Il est à noter que σ est un paramètre que l'on peut faire varier. Si σ est très petit alors l'erreur $|f(x)^l - y^l|$ devient très petite, mais alors la fonction $f(x)$ devient très chaotique. Il faut donc choisir σ avec en vue le problème de minimisation de l'erreur et de généralisation.

On peut remarquer que $f(x)$ est une fonction de régression non-linéaire qui permet une interpolation douce entre les données d'entrées/sorties.

1.5.5.3 Système Flou Optimal Adaptatif

On a pu voir que le système flou optimal utilise une règle pour chaque paire d'entrée/sortie. Ceci devient impraticable lorsque les données deviennent nombreuses. Il convient alors d'utiliser les techniques d'agrégation de données et chaque groupe de données sera représenté par une règle bien choisie.

Méthode d'Agrégation

- On commence avec la première paire d'entrée/sortie (x_1, y_1), on fixe le centre du groupe $x_{1,0}$ en x_1 et on fixe $A_1(1) = y_1$ et $B_1(1) = 1$. On choisit un rayon r.
- On suppose que lorsque l'on choisit la k^e paire d'entrée/sortie il y a déjà M groupe de données de créés avec M centres $x_{1,0}, x_{2,0}, x_{3,0}, \ldots, x_{M,0}$. Il faut alors calculer la distance de x_k à tous les centres des groupes par $|x_k - x_{l,0}|$, $l = 1, 2, \ldots, M$ et on considère la plus petite distance $|x_k - x_{l_k,0}|$ qui corresponds au groupe le plus proche de x_k et celui dont le centre est $x_{l_k,0}$.

  1. Si $|x_k - x_{l_k,0}| > r$,on fixe x_k comme un nouveau centre de groupe $x_{M+1,0}$, et on fixe $A_{M+1}(k) = y_k,\ B_{M+1}(k) = 1$ et on conserve $A_l(k) = A_l(k-1),\ B_l(k) = B_l(k-1)$ pour $l = 1, 2, \ldots, M$

2. Si $|x_k - x_{l_k,0}| \leq r$ on calcul de la manière suivante :

$$A_{l_k}(k) = A_{l_k}(k-1) + y_k \tag{1.32}$$

$$B_{l_k}(k) = B_{l_k}(k-1) + 1 \tag{1.33}$$

et on fixe

$$A_l(k) = A_l(k-1) \tag{1.34}$$

$$B_l(k) = B_l(k-1) \tag{1.35}$$

pour $l = 1, 2, \ldots, M$ avec $l \neq l_k$.

- Le système flou à l'étape k est calculer de la manière suivante :

$$f_k(x) = \frac{\sum_{l=1}^{M} A_l(k) \exp\left(-\frac{|x - x_{l,0}|^2}{\sigma^2}\right)}{\sum_{l=1}^{M} B_l(k) \exp\left(-\frac{|x - x_{l,0}|^2}{\sigma^2}\right)} \tag{1.36}$$

si x_k ne crée pas de nouveau groupe. Si x_k crée un nouveau groupe alors M de vient $M+1$ dans 1.36.

On peut remarquer que le rayon r est la variable qui permet de régler la complexité du système flou. Plus r est petit plus il y a de groupes et donc de règles floues.

On peut aussi voir que l'apprentissage de fait autant au niveau des paramètres que sur la structure : en effet la principale caractéristique est la possibilité d'ajouter des règles puisque l'on ajoute des groupes.

Il est possible d'ajouter un facteur d'oubli au équations des coefficients A et B 1.32,1.33,1.34,1.35. Ceci peut être très pratique dans le cas d'un système flou est utilisé pour modéliser un système variant dans le temps.

les équations 1.32,1.33,1.34,1.35 deviennent alors :

$$A_{l_k}(k) = \frac{\tau - 1}{\tau} A_{l_k}(k-1) + \frac{1}{\tau} y_k \tag{1.37}$$

$$B_{l_k}(k) = \frac{\tau - 1}{\tau} B_{l_k}(k-1) + \frac{1}{\tau} \tag{1.38}$$

et on fixe alors

$$A_l(k) = \frac{\tau - 1}{\tau} A_l(k-1) \tag{1.39}$$

$$B_l(k) = \frac{\tau - 1}{\tau} B_l(k-1) \tag{1.40}$$

où τ peut être considérer comme une constante temporelle d'une fonction de retard exponentiel.

1.5.5.4 Conclusion

Il est donc possible de construire un système flou performant grâce à cette algorithme. Ce type de système flou adaptatif est utilisé pour construire des régulateurs de systèmes non-linéaires dynamiques.

1.6 Logique Floue et Réseaux de Neurones

L'utilisation conjointe des réseaux de neurones et de la logique floue permet de tirer avantage des deux méthodes. Les capacités d'apprentissage de la première et la lisibilité et la souplesse de la seconde.

Diverses associations de ces deux méthodes ont été développées de puis 1988 et sont le plus souvent orientées vers la commandes de systèmes complexes et les problèmes de classification.

1.6.1 Méthodes Neuro-Floues

1.6.1.1 Première Méthode

L'une des premières méthodes est le codage du système d'inférences floues sous la forme d'un réseau de neurone multicouche dans lequel les poids correspondent aux paramètres du système. L'architecture du réseau dépend du type de règle et des méthodes d'inférence, d'agrégation et de défuzzification choisies (Cf. figure 1.8).

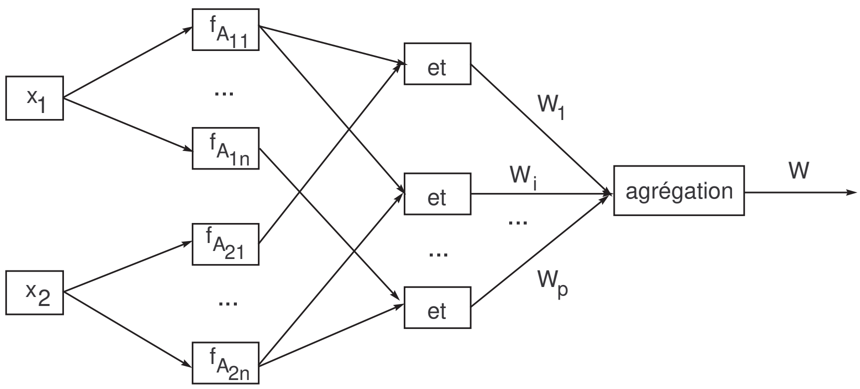

FIG. 1.8 – Exemple de réseau de neurones pour la commande floue

D'après la figure 1.8 pour des règles de la formes "si V_1 est A_{1i} et V_2 est A_{2i} alors $W = w_i$", on a un réseau de neurones qui admet pour entrée les valeurs x_1 et x_2 prises par les variables V_1 et V_2 et dont les deux couches cachées correspondent respectivement au calcul de la valeur des fonctions d'appartenance A_{1i} pour l'entrée x_1 et A_{2i} pour l'entrée x_2 et à celui de la valeur prise par la conjonction des conditions de chaque règle utilisant un opérateur adéquat. Les fonctions d'appartenance sont considérées comme des paramètres, ajustées par les poids entrant dans la première couche cachée. Les conclusions W_i des règles sont également des paramètres ajustables par l'intermédiaire des poids associés à la dernière couche.

1.6.1.2 Seconde Méthode

Cette seconde approche consiste à utiliser les réseaux de neurones pour remplacer chacune des composantes d'un système de commande floue. Ces réseaux servent d'apprentissage des fonctions d'appartenance, au calcul de l'inférence à la réalisation de la phase d'agrégation et de défuzzification. Ils peuvent réaliser l'extraction des règles floues en analysant la corrélation qui existe entre les entrées et les sorties du réseau de neurones. Ces approches permettent de résoudre deux problèmes important de la logique floue :
- Détermination des fonction d'appartenance
- Adaptation a l'environnement du système

1.6.1.3 Troisième Méthode

Elle utilise des réseaux de neurones et des systèmes flous associés en série ou en parallèle. Plusieurs variantes d'utilisation sont possibles :
- Le réseau de neurones fonctionne en amont du système flou. Les variables d'entrées du système flou sont déterminées à partir des sorties du réseau de neurones (dans le cas où elles ne sont pas mesurables directement) Cf. figure 1.9 ou encore un réseau de neurones effectue une tâche de classification ou de reconnaissances de formes, qui est suivit d'un système flou d'aide à la décision.
- Un réseau de neurones qui fonctionne en aval du système flou, dans le but d'ajuster les sortie d'un système de commande floue à de nouvelles connaissances obtenues, les variables de sortie étant les erreurs sur les variable de sortie du système flou Cf. figure 1.10

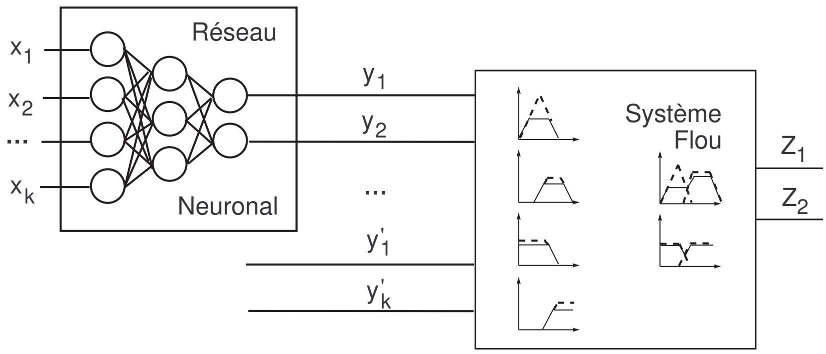

FIG. 1.9 – Exemple d'association en série d'un réseau de neurone et d'un système flou

1.7 Système Flou Adaptatif et Réseau de Neurones

1.7.1 Introduction

Les systèmes flous et les réseaux de neurones ont le même objectif, tous les deux tentent de modéliser le fonctionnement du cerveau humain. Mais il y une nuance entre les deux. Les réseaux de neurones tentent de modéliser l'architecture du cerveau en créant un modélisation de l'entité du cerveau : le neurone. Les systèmes flous eux modélisent le cerveau par son mode de fonctionnement (apprentissage et déduction).

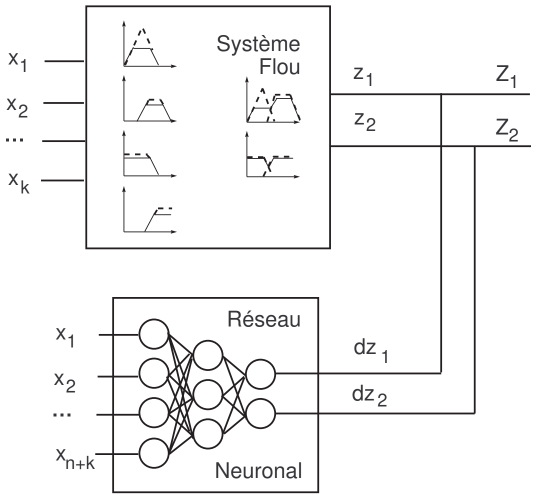

FIG. 1.10 – Exemple d'association en parallèle d'un réseau de neurone et d'un système flou

1.7.2 Perceptron et Système Flou Adaptatif

Un perceptron est un réseau de neurones multicouche à propagation de signal avant Cf. 1.11 Le signal se propage donc de la couche d'entrées vers la couche de sortie. Les

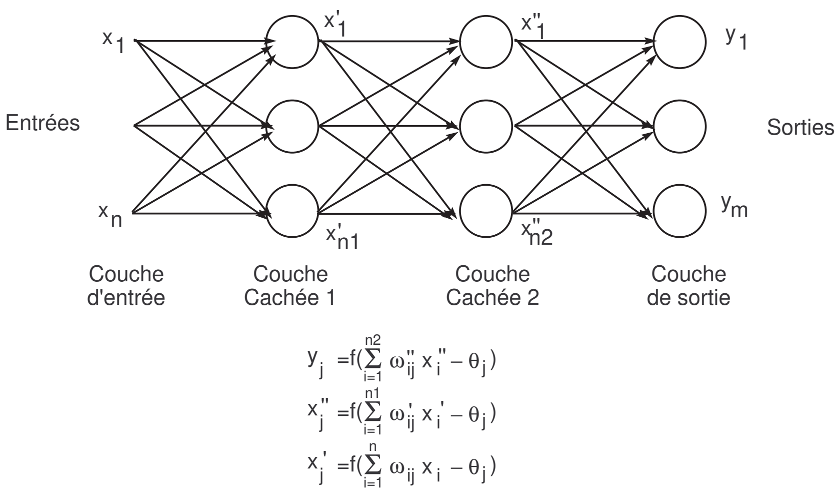

FIG. 1.11 – Perceptron à deux couches cachées

ω_{ij} représentent les poids des neurones, θ_j un seuillage et enfin la fonction $f(*)$ qui est le plus souvent une sigmoïde.

$$f(x) = \frac{1}{1 + \exp(-x)} \tag{1.41}$$

Ainsi pour un vecteur d'entrée $x = (x_1, \dots, x_n)^T$ on calcule successivement les sorties des deux couches cachées $x' = (x'_1, \dots, x'_{n1})^T$ et $x'' = (x''_1, \dots, x''_{n2})^T$ et enfin le sortie du réseau $y = (y_1, \dots, y_m)^T$

Si on utilise des données d'entrées/sorties pour l'apprentissage du réseau de neurones, il est alors possible d'utiliser l'algorithme de rétropropagation du gradient de l'erreur afin de minimiser une erreur quadratique.

A chaque couple entrée/sortie, une erreur est calculée, le gradient, ou pente, de l'erreur est déterminé. Ensuite les poids et les biais sont modifiés en ligne sur le réseau. On réitère ces calculs jusqu'à l'obtention du critère d'arrêt.

1.7.2.1 Algorithme

1. Initialisation des poids ω_{ij}^{q} à des petites valeurs aléatoires
2. Présentation d'une entrée x_k et de la sortie désirée d_k.
3. Calcul de la sortie actuelle par propagation à travers les couches par $y_j^{[q]} = f\left(\sum_i \omega_{ji}^{[q]} \cdot y_i^{[q-1]}\right)$ où f est la fonction de transfert du neurone et $[q]$ la q$^{\text{e}}$couche du réseau.
4. Accumulation des erreurs en sortie $\epsilon = \sum_k \left(d_k - y_k^{[s]}\right)$ Où d_k est la sortie désirée associée au vecteur d'entrée x_k. Où $y_k^{[s]}$ est la sortie obtenue sur la dernière couche au temps t. ϵ est l'erreur cumulée pour k présentations de couples (x_k, d_k).
5. Rétropropagation du gradient de l'erreur (δ) depuis la dernière couche vers la première couche
   - F Pour chaque cellule de sortie $\delta_i^{[s]} = -\left(d_i - y_i^{[s]}\right) \cdot f'\left(p_i^{[s]}\right)$
   - F Pour chaque cellule cachée $\delta_i^{[q]} = -\sum_k \delta_k^{[q+1]} \cdot \omega_{ki} \cdot f'\left(\delta_i^{[p]}\right)$
6. Mise à jour des poids selon la règle : $\Delta\omega_{ij}^{[q]} = \alpha \cdot \left(\delta_k^{[q]} \cdot x_j^{[q]}\right)$ où α est le coefficient d'apprentissage compris dans l'intervalle $[0, 1]$.
7. Retour à 2 tant qu'il y a des couples à présenter.

Choix du critère à minimiser

Dans le cas de la rétropropagation de l'erreur, le critère à minimiser est une erreur quadratique. L'application de l'algorithme du gradient nécessité la dérivabilité de la fonction de transfert. Le critère de minimisation d'erreur est le suivant $\epsilon = \sum_k \left(d_k - y_k^{[s]}\right)^2$

1.7.2.2 Avantages et Inconvénients

Avantages Ce fût un des premiers algorithmes développés pour l'apprentissage des réseaux de neurones multicouches de type feedforward. Il permet de pallier une carence de l'algorithme du perceptron qui est incapable de modifier les poids des couches cachées [DAVALO et NAÏM, 1993]. L'implémentation informatique ne présente pas de difficultés. Il a surtout permis de sortir le connexionnisme de l'impasse que représente l'apprentissage des réseaux de types feedforward multicouches en permettant la modification des poids des couches cachées.

Inconvénients En ce qui concerne l'algorithme

- L'algorithme de rétro-propagation du gradient de l'erreur suit la descente du gradient de l'erreur : un minimum local peut rapidement bloquer la recherche des optima globaux.

- L'algorithme de rétropropagation est gourmand en temps de calcul.
- Importance du choix du coefficient d'apprentissage, si le coefficient est trop grand la dynamique du réseau va osciller autour de l'optimum, si il est trop petit, la convergence est lente et elle va se faire piéger dans un minimum local.
- Les fonctions de transfert doivent être dérivables.

1.7.3 Réseau à Fonctions à Base Radiale et Système Flou Adaptatif

Basé sur une réalité biologique, Moody et Darken [MOODY et DARKEN, 1988] ont proposé un réseau de neurone nommé "Radial Basis Function Network (RBF Network) ".

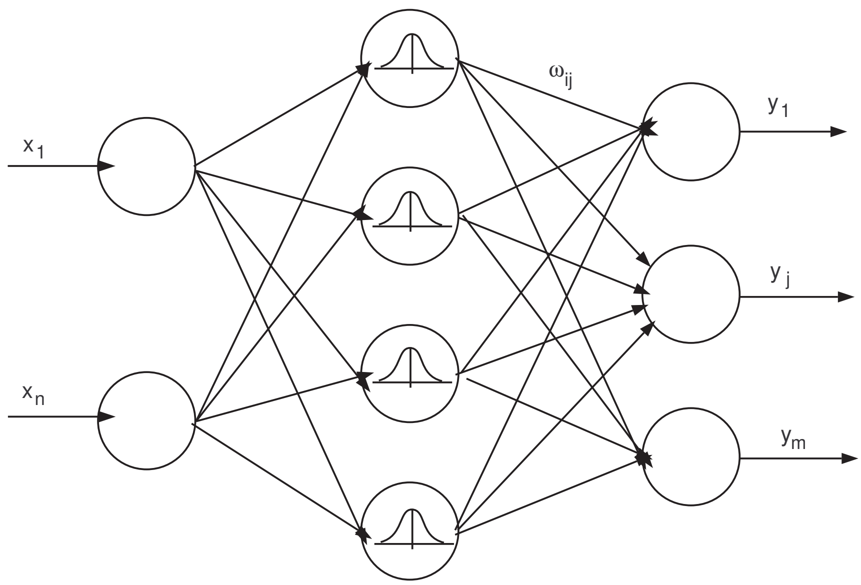

FIG. 1.12 – Réseau à Fonctions à Base Radiale

Chaque sortie y_i peut être calculée de deux manières :

$$y_i = \sum_{j=1}^{n} \omega - ij R_j(x) \tag{1.42}$$

la plus simple ou

$$y_i = \frac{\sum_{j=1}^{n} \omega - ij R_j(x)}{\sum_{j=1}^{n} R_j(x)} \tag{1.43}$$

qui donne une réponse normée avec $R_j(x)$ tel que

$$R_j = \exp\left(\sum_{i=1}^{n} -\frac{(x_i - c_{i,j})^2}{\sigma_{i,j}^2}\right) \tag{1.44}$$

ou

$$R_j = \frac{1}{1 + \exp\left(\sum_{i=1}^{n} -\frac{(x_i - c_{i,j})^2}{\sigma_{i,j}^2}\right)} \tag{1.45}$$

Après avoir ecrit ces équations (1.10 et 1.43, il est très facile de se rendre compte de la similitude entre les réseaux RBF et les systèmes flous définit par Wang 1.8.

Mais ces deux approches sont différentes en ce sens que les réseaux RBF sont un cas particulier des systèmes flous adaptatifs du fait des restrictions choisie par [WANG, 1995].

De plus les fonctions d'appartenance peuvent prendre différentes formes (gaussienne, triangulaire, trapézoïdale, ...) alors que dans un réseau RBF les fonctions de transferts sont toujours gaussiennes.

1.7.3.1 Conclusion

Bien que les deux approches soit différentes, l'une basée sur l'architecture du cerveau humain et l'autre sur son mode de fonctionnement, les équations résultantes sont très semblables.

Mais on peut remarquer que les systèmes flous comparés aux réseaux RBF sont plus généraux.

De plus les réseaux de neurones ne peuvent prendre intégrer facilement de la connaissances linguistiques, ils sont basés sur un apprentissage des entrées/sorties.

1.8 Commande Floue Adaptative à Stabilité Indirecte de Systèmes Non-Linéaires

1.8.1 Introduction

Lorsque les systèmes flous développés précédemment sont utilisés comme régulateurs on parle alors de régulateurs flous. Ces régulateurs sont largement utilisés dans l'industrie [TANAKA *et al.*, 1995]

La commande floue possède des avantages indéniables. Dans le cas d'un modèle mathématique du système difficile à obtenir, il est très pratique d'utiliser les informations que l'on peut obtenir facilement (mesures numériques des variables utiles, description linguistiques du système par les experts humains). Par définition le système flou peut prendre en compte les deux types de données.

Cette approche ne nécessite pas de modèle mathématique du système.

La commande floue génère des régulateurs non-linéaires qui est en concordance avec le fait qu'il sont des approximateurs universels. Il devient alors possible de construire un régulateur flou pour n'importe quel système non-linéaire.

La commande floue est aussi simple à comprendre car elle reprend le langage humain dans ses définitions de règles.

Ce type de commande reste simple à mettre en œuvre avec les ordinateur et calculateur dont nous disposons de nos jours ce qui contribue à en faire approche très bon marché pour les entreprise.

1.8.2 Les différents Types de Commandes Floues

1.8.2.1 Commande Floue Adaptative

Le plus souvent les régulateurs flous sont utilisés dans des systèmes qui possèdes des variations inconnues intrinsèques. L'objectif est donc de conserver de bonne performance du système complet en adaptant le régulateur en fonction des variations du système.

La commande floue adaptative se compose donc d'un régulateur flou adaptatif (soit unique soit choisi parmi un groupe de régulateur en fonction de sa performance). La chose la plus importante qui différencie un régulateur flou adaptatif et un régulateur adaptatif conventionnel est que le premier peut prendre en compte des données linguistiques. Ce qui est très important lorsque le système possède des incertitudes que l'opérateur humain à appris à anticiper.

1.8.2.2 Commande Adaptative Floue Directe et Indirecte

D'après NARENDRA et PARTHASARATHY [NARENDRA et PARTHASARATHY, 1990], les régulateurs flous sont séparés en deux classes :

- Les régulateurs adaptatifs directs : ici les paramètres du régulateurs sont modifiés en ligne afin de minimiser l'erreur entre le modèle de référence et le système réel. Donc si le régulateur possède un système à base de logique floue on parle alors de régulateur flou adaptatif direct.
- Les régulateurs adaptatifs indirects : les paramètres du système réel sont estimés et le régulateur est choisi en supposant que les valeurs estimées du sont les vrais valeurs du système réel. Si le modèle du procédé réel utilise un système à base de logique floue on parle alors de régulateur flou adaptatif indirect.

1.8.2.3 Types de Commande floue Adaptative

Trouver un bon régulateur flou dépend de la facilité de trouver les bons paramètres de celui-ci par les différentes méthodes d'apprentissage décrites précédemment.

Il est aisé de comprendre qu'il estplus simple de trouvé un régulateur flou optimal (qui répond le mieux au critère de performance choisi) s'il est linéaire par rapport à ces paramètres ajustables. D'un autre coté un régulateurs flou non-linéaire par rapport à ces paramètres sera beaucoup plus difficile à trouver mais ses performances seront sans doute meilleures.

On classe donc ces régulateurs flous en deux types :

- Les régulateurs flous comprenant des systèmes flous linéaires par rapport à leurs paramètres ajustables sont dit «Régulateur Flou Adaptatif du Premier Type».
- Les régulateurs flous comprenant des systèmes flous non-linéaires par rapport à leurs paramètres ajustables sont dit «Régulateur Flou Adaptatif du Second Type».

Dans le cas du premier type de régulateur on utilise le système flou décrit par l'équation suivante :

$$f(\underline{x}) = \sum_{l=1}^{M} \theta_l \xi_l(\underline{x}) = \underline{\theta}^T \underline{\xi}(\underline{x}) \tag{1.46}$$

où $\underline{\theta} = (\theta_1, \ldots, \theta_M)^T, \underline{\xi} = (\xi_1(\underline{x}), \ldots, \xi_M(\underline{x}))^T, \xi_l(\underline{x})$ est n fonction à base floue définie par

$$\xi(\underline{x}) = \frac{\prod_{i=1}^{n} \mu_{F_i^l}(x_i)}{\sum_{l=1}^{M} \prod_{i=1}^{n} \mu_{F_i^l}(x_i)} \tag{1.47}$$

θ_l sont les paramètres ajustables et $\mu_{F_i^l}$ sont des fonctions d'appartenance données.

Dans le cas du second type de régulateur on utilise le système flou décrit par l'équation suivante

$$f(\underline{x}) = \frac{\sum_{l=1}^{M} \bar{y}_l \left[\prod_{i=1}^{n} a_{l,i} \exp\left(-\left(\frac{x_i-\bar{x}_{l,i}}{\sigma_{l,i}}\right)^2\right)\right]}{\sum_{l=1}^{M} \left[\prod_{i=1}^{n} a_{l,i} \exp\left(-\left(\frac{x_i-\bar{x}_{l,i}}{\sigma_{l,i}}\right)^2\right)\right]} \tag{1.48}$$

où $\bar{y}_l, \bar{x}_{l,i}$ et $\sigma_{l,i}$ sont les paramètres ajustables.

On peut remarquer d'après les équations que les deux types de régulateurs flous adaptatifs sont non-linéaires.

1.8.3 Conception de Régulateur Flou Adaptatif Indirect basé sur l'approche de Lyapunov

1.8.3.1 Méthode de Lyapunov

La méthode de Lyapunov permet d'étudier la stabilité de systèmes complexes (régit par des systèmes différentielles).

L'étude la stabilité des systèmes différentielle est énoncé de la manière suivante.

Théorème 1.8.1. *Soit $V(x)$ une fonction candidate de Lyapunov.*

a) $V(x) > 0, \quad$ *et* $\forall x \neq 0, V(0) = 0$

b) $\exists \alpha(.), \beta(.)$ *fonctions scalaires* $\mathbb{R}^{(n)} \to \mathbb{R}$ *continues telles que* $\alpha(\|x\|) \leq V(x) \leq \beta(\|x\|)$

c) *quand* $\|x\| \to \infty$ *alors* $V(x) \to \infty$ *fonction indéfiniment croissante.*

Si $\dot{V}(x) < 0, \ \forall x \in \mathcal{D}_n$ *domaine de stabilité, la solution de su système différentiel est asymptotiquement stable.*

1.8.3.2 Application à l'Automatique dans le Cas d'un Système Libre

En Continu Soit le système décrit par la représentation d'état suivante

$$\dot{X} = AX(t)$$

Il s'agit d'un système libre (pas de commande sur ce système la matrice B est nulle). Soit Q une matrice positive.

Théorème 1.8.2. *Le système est stable si et seulement si* $\exists P > 0, \ A^T P + PA + Q = 0$ *(Equation de Lyapunov).*

De plus $V(X) = X^T PX$ *est une fonction candidate de Lyapunov décroissante et* $\int_0^\infty X^T QX \, dA = X_{(0)}^T PX_{(0)}$ *avec* $\lambda_{\min}\|X\|^2 \leq X^T PX \leq \lambda_{\max}\|X\|^2$

En Discret Soit le système décrit par la représentation d'état suivante

$$X(k+1) = AX(k)$$

Il s'agit d'un système libre (pas de commande sur ce système la matrice B est nulle). Soit Q une matrice positive.

Théorème 1.8.3. *Le système est stable si et seulement si* $\exists P > 0,\ A^T PA - P + Q = 0$ *(Equation de Lyapunov en discret).*

De plus $V(X) = X^T PX$ *est une fonction candidate de Lyapunov décroissante et* $\sum_{i=0}^{\infty} X^T(i)QX(i) = X_{(0)}^T PX_{(0)}$ *et* ΔV *devient*

$$\begin{aligned} \Delta V\left(X(k)\right) &= V\left(X(k+1)\right) - V\left(X(k+1)\right) \\ &= X(k+1)^T PX(k+1) - X(t)^T PX(k) \\ &= X^T(A^T PA - P)X \end{aligned}$$

1.8.3.3 Exemples

1. Soit l'équation différentielle suivante :

$$\dot{x} = -x + \frac{1}{2}\sin x,\ \text{avec } x \in \mathbb{R}$$

Soit $V(x) = \frac{1}{2}\|x\|^2 = \frac{1}{2}\underline{x}^T x$. On a alors

$$\dot{V}(x) = \dot{x}x = \underbrace{-x^2 + \underbrace{\frac{1}{2}x\sin x}_{<x^2}}_{<0}$$

La dérivée est donc strictement décroissante. La solution de l'équation est donc asymptotiquement stable.

2. Soit le système différentiel suivant :

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & -1 + x_1^2(t) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Soit $V(x) = \frac{1}{2}(x_1^2 + x_2^2)$. On a alors

$$\begin{aligned} \dot{V}(x) &= x_1\dot{x}_1 + x_1\dot{x}_1 \\ &= x_1x_2 - x_1x_2 + \underbrace{x_2^2(x_1^2 - 1)}_{<0?} \end{aligned}$$

Le domaine de stabilité $\mathcal{D}_s$ est tel que $|x_1| < 1$
Le système est donc asymptotiquement stable si $x(0) \in \mathcal{D}_s$ Cf. figure 1.13.

1.8.3.4 Application à l'Automatique dans le Cas Général

Soit le système décrit par la représentation d'état suivante

$$\dot{X} = AX(t) + Bu(t)$$

. Soit $Q \geq 0$ et $R > 0$.

Soit l'équation de Riccati : $A^T P + PA - PBR^{-1}B^T P + Q = 0$

Théorème 1.8.4. *Si les matrices* A *et* B *sont sous la forme commandable alors* $\exists! P > 0$ / *la loi* $u(t) = -R^{-1}B^T PX(t)$ *stabilise le système en boucle fermée et minimise le critère* $J = \int_0^{\infty} X^T QX + U^T RU\, dt$.

De plus $V = X^T PX$ *est une fonction candidate de Lyapunov décroissante et* $\int_0^{\infty} X^T QX\, dA = X_{(0)}^T PX_{(0)}$ *avec* $\lambda_{\min}\|X\|^2 \leq X^T PX \leq \lambda_{\max}\|X\|^2$

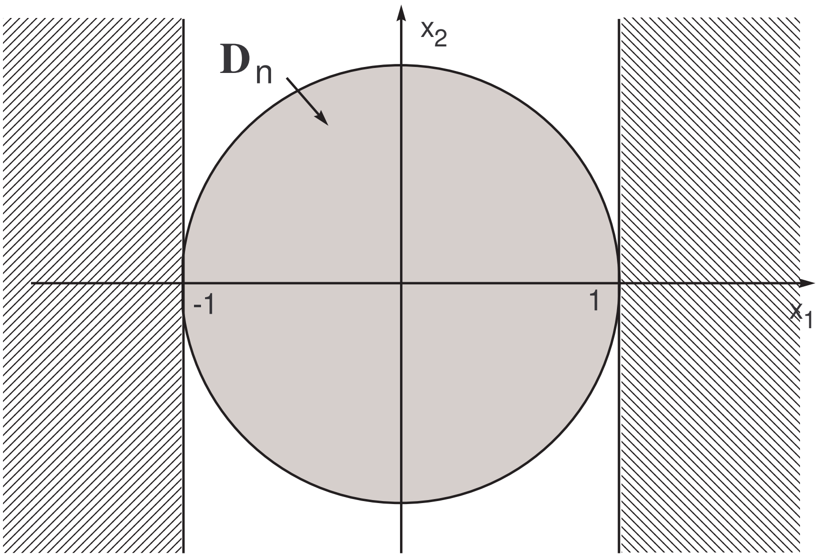

FIG. 1.13 – Domaine de stabilité du système

1.8.3.5 Application à la Conception de Régulateurs Flous

Soit le système flou décrit par les règles suivantes $R_i \ : \ i \in \{1, \dots, r\}$ de la forme :

$$\text{si } z_1(t) \text{ est } F_1^i \text{ et } \dots \ z_p(t) \text{ est } F_p^i$$

Sa solution est décrite par le système suivant :

$$\begin{cases} \dot{X}(t) & = & A_i X(t) + B_i u(t) \\ Y(t) & = & C_i(t) \end{cases}$$

Et d'après la définition des règles floues on a $\omega_i\left(z(t)\right) = \prod_{j=1}^{p} \mu_{F_j^i}(z_j(t))$ d'où

$$\begin{cases} \dot{X}(t) & = & \frac{\sum_{i=1}^{r} \omega_i(z(t))(A_i X(t) + B_i u(t))}{\sum_{i=1}^{r} \omega_i(z(t))} \\ \\ Y(t) & = & \frac{\sum_{i=1}^{r} \omega_i(z(t))(C_i X(t))}{\sum_{i=1}^{r} \omega_i(z(t))} \end{cases} \tag{1.49}$$

Posons $h(t)$ tel que $0 \leq h_i\left(z(t)\right) = \frac{\omega_i(z(t))}{\sum_{i=1}^{r} \omega_i(z(t))} \leq 1$ et $\sum_{i=1}^{r} h_i\left(z(t)\right) = 1$
le système devient alors

$$\begin{cases} \dot{X}(t) & = & \sum_{i=1}^{r} h_i\left(z(t)\right)\left(A_i X(t) + B_i u(t)\right) \\ Y(t) & = & \sum_{i=1}^{r} h_i\left(z(t)\right)\left(C_i X(t)\right) \end{cases} \tag{1.50}$$

Cas du système libre Soit le système libre décrit par $\sum_{i=1}^{r} h_i\left(z(t)\right)\left(A_i X(t)\right)$ D'après le théorème de Lyapunov soit $V = X^T P X$ on a alors

$$\dot{V} = \sum_{i=1}^{r} \underbrace{h_i\left(z(t)\right)}_{\geq 0} \underbrace{\left[X^T \left(A_i^T P + P A_i\right) X\right]}_{\forall i, <0}$$

Pour assurer la stabilité il faut donc trouver P telle que $\forall i, \exists \ P; A_i^T P + P A_i < 0$

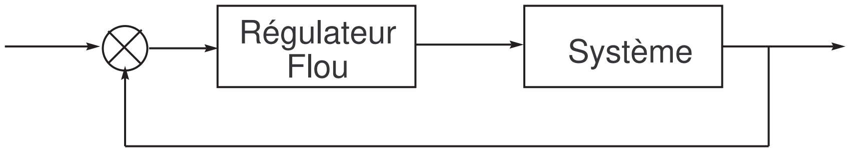

FIG. 1.14 – Schéma d'un système complexe contrôlé par régulateur flou

Cas du système général Soit le système de la figure 1.14 décrit par

$$\left\{ \begin{array}{rcl} \dot{X}(t) & = & \sum_{i=1}^{r} h_i\left(z(t)\right)\left(A_i X(t) + B_i u(t)\right) \\ U(t) & = & -\sum_{i=1}^{r} h_i\left(z(t)\right)\left(F_i X(t)\right) \end{array} \right. \tag{1.51}$$

On a $u(t) = \sum_{i=1}^{r} h_i\left(z(t)\right)\left(F_i X(t)\right)$

Donc $\dot{X}$ devient :

$$\begin{array}{rcl} \dot{X} & = & \sum_{i=1}^{r} h_i\left(z(t)\right)\left(A_i X(t) - B_i \sum_{i=1}^{r} h_i\left(z(t)\right) F_i\right) X(t) \\ & = & \sum_{i=1}^{r} h_i A_i X(t) - \sum_{i=1}^{r} h_i B_i \sum_{i=1}^{r} h_i F_i X \\ & = & \sum_{i=1}^{r} \sum_{j=1}^{r} h_i h_j \left(A_i - B_i F_i\right) X(t) \end{array}$$

de plus on $\sum h_i = 1$ D'après Lyapunov il faut trouver P telle que :

$$\forall i, \forall j \ \in \mathbb{R};\ (A_i - B_i F_j)^T P + P(A_i - B_i F_j) < 0$$

Mais $\dot{X}$ peut s'écrire :

$$\dot{X} \ = \ \sum_{i=1}^{r} h_i^2\left(A_i - B_i F_i\right) X(t) + \sum_{j>i}^{r} h_i h_j\left(A_i - B_i F_j + A_j - B_j F_i\right) X(t)$$

1.8.3.6 Synthèse de Régulateur Flou d'après Wang

Soit le système d'ordre n sous sa forme commandable :

$$\begin{array}{rcl} \dot{x}_1 & = & x_2 \\ \dot{x}_2 & = & x_3 \\ & \cdots & \\ \dot{x}_{n-2} & = & x_{n-1} \end{array}$$

$$\dot{x}_n = f(x_1, \ldots, x_n) + g(x_1, \ldots, x_n)u,\ y = x_1 \tag{1.52}$$

ou la forme équivalente

$$x^{(n)} = f(x, \dot{x}, \ldots, x^{(n-1)}) + g(x, \dot{x}, \ldots, x^{(n-1)})u,\ y = x \tag{1.53}$$

où f et g sont des fonctions inconnues, $u \in \mathbb{R}$ et $y \in \mathbb{R}$ sont les entrées et les sorties du système. $\underline{x} = (x_1, \ldots, x_n)^T = (x_1, \ldots, x_n)^T \in \mathbb{R}$ est le vecteur d'état du système supposé mesurable.

Stabilité du régulateur On suppose aussi pour la commandabilité du système que $\forall \underline{x} \in U_c \subset \mathbb{R}^n, g(\underline{x}) \neq 0$ et par commodité on prendra $g(\underline{x}) > 0, \forall \underline{x} \in U_c$ région certaine de controllabilité.

On doit maintenant déterminer une commande $u = u(\underline{u} \mid \underline{\theta})$ floue et une loi d'adaptation des paramètres du vecteur $\underline{\theta}$ telles que :

1. Le système en boucle fermée doit être asymptotiquement stable c-à-d que toutes les variables $\underline{x}(t), \underline{\theta}(t)$ et $u(\underline{x} \mid \underline{\theta}$doivent être bornées.
2. L'erreur entre la sortie réelle et la sortie simulée, $e = y_m - y$ doit être minimisée.

Soit $\underline{e} = (e, \dot{e}, \ldots, e^{n-1})^T$ vecteur des dérivées n-ième et $\underline{k} = (k_n, \ldots, k_1)^T \in \mathbb{R}$ tels que toutes les racines du polynômes de stabilité $h(s) = s^n + k_1 s^{n-1} + \ldots + k_n$ soient dans le demi-plan gauche.

Si les fonction f et g sont connues alors la loi de commande est :

$$u = \frac{1}{g(\underline{x})}\left[-f(\underline{x}) + y_m^{(n)} + \underline{k}^T \underline{e}\right] \tag{1.54}$$

en remplaçant u de 1.54 dans 1.52 on obtient :

$$y^{(n)} = -\underline{k}^T e - y_m^{(n)} e^{(n)} + \underbrace{k_1 e^{(n-1)} + \ldots + k_n e}_{\underline{e}.\underline{k}^T} = 0 \tag{1.55}$$

donc $\lim_{t\to\infty} e(t) = 0$ qui est l'objectif de la commande.

Régulateur équivalent On remplace f et g inconnue par par les systèmes flous $\hat{f}(\underline{x} \mid \underline{\theta})$ et $\hat{g}(\underline{x} \mid \underline{\theta})$ du types de 1.46 et 1.48.

$\hat{f}(\underline{x} \mid \underline{\theta})$ et $\hat{g}(\underline{x} \mid \underline{\theta})$ sont donc le la forme :

$$\hat{f}(\underline{x} \mid \underline{\theta}_f) = \xi_f(\underline{x}).\underline{\theta}_f$$

$$\hat{g}(\underline{x} \mid \underline{\theta}_g) = \xi_g(\underline{x}).\underline{\theta}_g$$

et la loi de commande devient :

$$u_c = \frac{1}{\hat{g}(\underline{x} \mid \underline{\theta})}\left[-\hat{f}(\underline{x} \mid \underline{\theta}_f) + y_m^{(n)} + \underline{k}^T \underline{e}\right] \tag{1.56}$$

En remplaçant u_c de 1.56 dans 1.52 on obtient :

$$\begin{aligned} e^{(n)} =& y_m^{(n)} - f(\underline{x}) - \frac{\hat{g}(\underline{x} \mid \underline{\theta}_g) - g(\underline{x})}{\hat{g}(\underline{x} \mid \underline{\theta}_g)}.\left(-\hat{f}(\underline{x} \mid \underline{\theta}_f) + \underline{k}^T \underline{e} + y_m^{(n)}\right) \\ &+ \hat{f}(\underline{x} \mid \underline{\theta}_f) - \underline{k}^T \underline{e} - y_m^{(n)} \\ =& -\underline{k}^T \underline{e} + \hat{f}(\underline{x} \mid \underline{\theta}_f) - f(\underline{x}) + \left(\hat{g}(\underline{x} \mid \underline{\theta}_g) - g(\underline{x})\right) u_c \end{aligned} \tag{1.57}$$

ou de façon équivalente on peut écrire

$$\dot{\underline{e}} = \Lambda_c \underline{e} + \underline{b}_c \left[\left(\hat{f}(\underline{x} \mid \underline{\theta}_f) - f(\underline{x})\right) + \left(\hat{g}(\underline{x} \mid \underline{\theta}_g) - g(\underline{x})\right) u_c\right] \tag{1.58}$$

avec

$$\Lambda_c = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & l\dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \\ -k_n & -k_{n-1} & \dots & \dots & \dots & \dots & -k_1 \end{bmatrix}, \underline{b}_c = \begin{bmatrix} 0 \\ \dots \\ 0 \\ 1 \end{bmatrix} \tag{1.59}$$

On sait alors que si Λ_c est une matrice stable, alors $\exists P$ une matrice définie positive unique qui satisfait l'équation de Lyapunov :

$$\Lambda_c^T P + P\Lambda_c = -Q \tag{1.60}$$

ou Q est une matrice définie positive.

Prenons $V = \underline{e}^T P \underline{e}$ la dérivée $\dot{V}$ s'écrit d'après 1.58 donc :

$$\begin{aligned} \dot{V} &= \underline{e}^T \left(\Lambda_c^T P + P\Lambda_c\right) \underline{e} + 2\underline{e}^T P \underline{b}_c \left(\hat{f}(\underline{x} \mid \underline{\theta}_f) - f(\underline{x})\right) + \left(\hat{g}(\underline{x} \mid \underline{\theta}_g) - g(\underline{x})\right) u_c \\ &= - \underbrace{\underline{e}^T Q \underline{e}}_{<0 \ pour \ la \ stabilit\acute{e}} \underbrace{+ 2\underline{e}^T P \underline{b}_c \left(\hat{f}(\underline{x} \mid \underline{\theta}_f) - f(\underline{x})\right) + \left(\hat{g}(\underline{x} \mid \underline{\theta}_g) - g(\underline{x})\right) u_c}_{<0 \ ce \ qui \ est \ difficile \ \grave{a} \ assurer} \end{aligned}$$

Commande de supervision Pour simplifier le problème on remplace u par $u = u_c + u_s$. Le terme additionnel u_s est une commande forte dont le but est de forcer $\dot{V} \leq 0$ et en reprenant le calcul on obtient :

$$\underline{\dot{e}} = \Lambda_c \underline{e} + \underline{b}_c \left[\left(\hat{f}(\underline{x} \mid \underline{\theta}_f) - f(\underline{x})\right) + \left(\hat{g}(\underline{x} \mid \underline{\theta}_g) - g(\underline{x})\right) u_c - g(x) u_s\right] \tag{1.61}$$

de même $\dot{V}$ devient :

$$\begin{aligned} \dot{V} &= \underline{e}^T \left(\Lambda_c^T P + P\Lambda_c\right) \underline{e} + 2\underline{e}^T P \underline{b}_c \left(\left(\hat{f}(\underline{x} \mid \underline{\theta}_f) - f(\underline{x})\right)\right. \\ &\qquad \left. + \left(\hat{g}(\underline{x} \mid \underline{\theta}_g) - g(\underline{x})\right) u_c - g(\underline{x}) u_s\right) \\ &\leq -\underline{e}^T Q \underline{e} + 2|\underline{e}^T P \underline{b}_c| \left(|\hat{f}(\underline{x} \mid \underline{\theta}_f)| + |f(\underline{x})|\right. \\ &\qquad \left. + |\hat{g}(\underline{x} \mid \underline{\theta}_g) u_c| + |g(\underline{x}) u_c|\right) - 2\underline{e}^T P \underline{b}_c g(\underline{x}) u_s \end{aligned} \tag{1.62}$$

Hypothèse On suppose qu'il existe $f^U(\underline{x})$, $g^U(\underline{x})$ et $g_L(\underline{x})$ telles que $\forall \underline{x} \in \mathbb{R}, |f(x)| \leq f^U(\underline{x}), 0 < g_L(\underline{x}) \leq g(\underline{x}) \leq g^U(\underline{x}) < \infty$

On choisit alors u_s de la manière suivante :

$$u_s = I_1^* sign\left(\underline{e}^T P \underline{b}_c\right) \frac{1}{g_L(\underline{x})} \left[\left(|\hat{f}(\underline{x} \mid \underline{\theta}_f)| + f^U(\underline{x}) + |\hat{g}(\underline{x} \mid \underline{\theta}_g) u_c| + |g^U(\underline{x}) u_c|\right)\right]$$

avec $u_s \neq 0$ si $V_e > \bar{V}$ c-à-d $I_1^* = 1$ si $V_e > \bar{V}$ et $I_1^* = 0$ sinon.

et d'après l'équation 1.62 on obtient :

$$\begin{aligned} \dot{V} &\leq -\underline{e}^T Q \underline{e} + 2|\underline{e}^T P \underline{b}_c| \left[\left(|\hat{f}| + |f| + |\hat{g} u_c| + |g u_c|\right)\right. \\ &\qquad \left. - \frac{g}{g_L}\left(|\hat{f}| + f^U + |\hat{g} u_c| + |g^U u_c|\right)\right] \\ &\leq -\underline{e}^T Q \underline{e} < 0 \end{aligned} \tag{1.63}$$

Loi d'adaptation des paramètres du régulateur flou On définit les relations suivantes :

$$\underline{\theta}_f^* = argmin_{\underline{\theta}_f \in \Omega_f} \left[\sup_{\underline{x} \in U_c} |\hat{f}(\underline{x} \mid \underline{\theta}_f) - f(\underline{x})| \right] \tag{1.64}$$

$$\underline{\theta}_g^* = argmin_{\underline{\theta}_g \in \Omega_g} \left[\sup_{\underline{x} \in U_c} |\hat{g}(\underline{x} \mid \underline{\theta}_g) - g(\underline{x})| \right] \tag{1.65}$$

où Ω_f et Ω_g sont les espaces de contraintes de $\underline{\theta}_f$et $\underline{\theta}_g$ définit tels que

$$\Omega_f = \{\underline{\theta}_f \ : \ |\underline{\theta}_f| \leq M_f\} \tag{1.66}$$

$$\Omega_g = \{\underline{\theta}_g \ : \ |\underline{\theta}_g| \leq M_g\} \tag{1.67}$$

On définit alors l'erreur d'approximation minimum par

$$\omega = \left(\hat{f}(\underline{x} \mid \underline{\theta}_f^*)\right) - f(\underline{x}) + \left(\hat{f}(\underline{x} \mid \underline{\theta}_g^*) g(\underline{x}) - \right) u_c \tag{1.68}$$

On peut alors réécrire l'équation 1.61 :

$$\begin{aligned} \underline{\dot{e}} =& \Lambda_c \underline{e} + \underline{b}_c g(\underline{x}) u_s + \underline{b}_c \left[\left(\hat{f}(\underline{x} \mid \underline{\theta}_f) - \hat{f}(\underline{x} \mid \underline{\theta}_f^*) \right) \right. \\ & \left. + \left(\hat{g}(\underline{x} \mid \underline{\theta}_g) - \hat{g}(\underline{x} \mid \underline{\theta}_g^*) \right) u_c - \omega \right] \end{aligned} \tag{1.69}$$

Si les systèmes flous $\hat{f}$ et $\hat{g}$ sont choisi de la forme de 1.46 on peut écrire :

$$\underline{\dot{e}} = \Lambda_c \underline{e} + \underline{b}_c g(\underline{x}) u_s + \underline{b}_c \omega + \underline{b}_c \left[\underline{\phi}_f^T \underline{\xi}(\underline{x}) + \underline{\phi}_g^T \underline{\xi}(\underline{x}) u_c \right] \tag{1.70}$$

où $\underline{\phi}_f = \underline{\theta}_f - \underline{\theta}_f^*$ et $\underline{\phi}_g = \underline{\theta}_g - \underline{\theta}_g^*$ et $\underline{\xi}(\underline{x})$ est le vecteurs des fonctions à bases floues.
On choisi alors la fonction candidate de Lyapunov V :

$$V = \underline{e}^T P \underline{e} + \frac{1}{\gamma_1} \underline{\phi}_f^T \underline{\phi}_f + \frac{1}{\gamma_2} \underline{\phi}_g^T \underline{\phi}_g$$

avec $\gamma_1 > 0$ et $\gamma_2 > 0$.

$$\begin{aligned} \dot{V} &= -\underline{e}^T Q \, \underline{e} \ + \ \underline{e}^T P \, \underline{b}_c \ \left(\underline{\phi}_f^T \underline{\xi}(\underline{x}) + \underline{\phi}_g^T \underline{\xi}(\underline{x}) u_c + \omega \right) \\ & \quad + \ \frac{2}{\gamma_1} \underline{\phi}_f^T \, \dot{\theta}_f \ + \ \frac{2}{\gamma_2} \, \underline{\phi}_g^T \, \dot{\theta}_g \ - \ 2\underline{e}^T P \, \underline{b}_c \, g(\underline{x}) \, u_s \\ \dot{V} &= -\underline{e}^T Q \, \underline{e} \ \underbrace{-2\underline{e}^T P \, \underline{b}_c \, g(\underline{x}) \, u_s}_{<0} + 2\underline{e}^T P \, \underline{b}_c \, \omega \\ & \quad + \ 2\frac{\underline{\phi}_f^T}{\gamma_1} \ \left(\underline{\dot{\theta}}_f \ + \ \gamma_1 \, \underline{e}^T P \, \underline{b}_c \, \underline{\xi}(\underline{x}) \right) \\ & \quad + \ 2\frac{\underline{\phi}_g^T}{\gamma_1} \ \left(\underline{\dot{\theta}}_g \ + \ \gamma_1 \, \underline{e}^T P \, \underline{b}_c \, \underline{\xi}(\underline{x}) \, u_c \right) \end{aligned} \tag{1.71}$$

On choisi alors la loi d'adaptation des paramètres suivantes :

$$\begin{cases} \dot{\underline{\theta}}_f &= -\gamma_1 \underline{e} P \underline{b}_c \underline{\xi}(\underline{x}) \\ \dot{\underline{\theta}}_g &= -\gamma_2 \underline{e} P \underline{b}_c \underline{\xi}(\underline{x}) u_c \end{cases} \tag{1.72}$$

Ceci a pour effet d'annuler une partie de l'équation 1.71 qui devient :

$$\dot{V} \leq -\underline{e}^T Q \underline{e} + \underline{e}^T P \underline{b}_c \omega \tag{1.73}$$

Le terme $\underline{e}^T P \underline{b}_c \omega$ est de l'ordre de l'erreur d'approximation minimum.

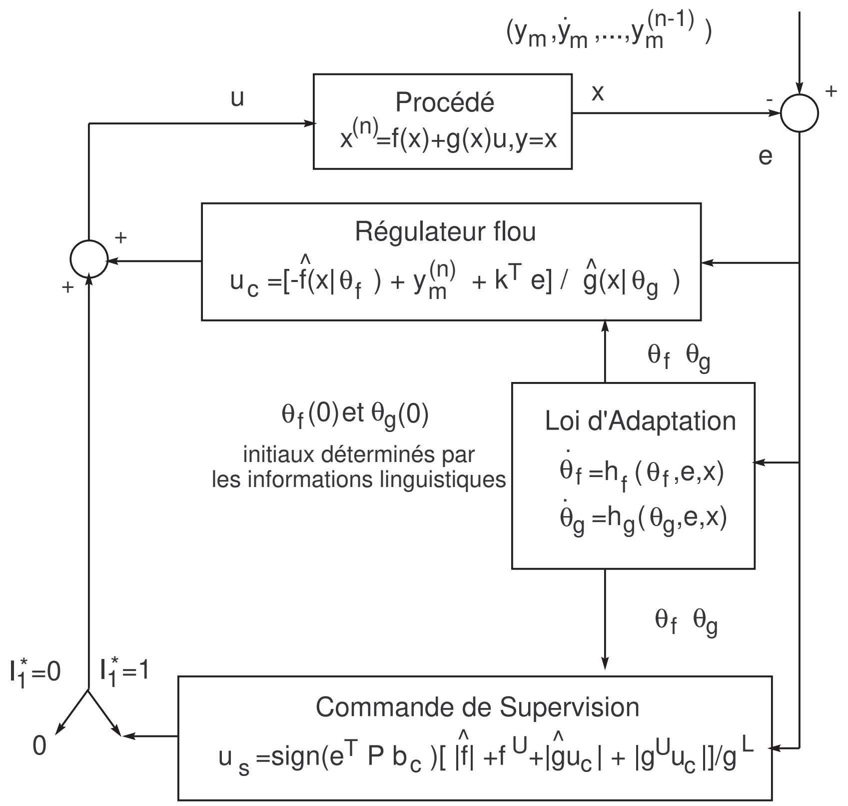

Fig. 1.15 – Schéma complet de la commande adaptative floue indirecte

1.8.3.7 conclusion

Afin de mener à bien cette approche le vecteur d'état $\underline{x}$ doit être accessible. Dans le même odre d'idée la commande u.

Si la connaissance a priori du système est très faible la convergence de l'algorithme sera beaucoup plus longue.

Cette méthode permet de déterminer un régulateur flou adaptatif qui assure la stabilité en boucle fermée.

$$
\left|\begin{array}{rcl}
\dot{V} & = & \underline{e}^T \left(\Lambda_c^T P \;+\; P\Lambda_c\right) \underline{e} \;+\; 2\,\underline{e}^T P \underline{b}_c \left(\left(\hat{f}(\underline{x} \mid \underline{\theta}_f) - f(\underline{x})\right)\right. \\
& & \left.\qquad +\; \left(\hat{g}(\underline{x} \mid \underline{\theta}_g) - g(\underline{x})\right) u_c - g(\underline{x}) u_s\right) \\
& \leq & -\underline{e}^T Q\, \underline{e} \;+\; 2\, \left|\underline{e}^T P \underline{b}_c\right| \left(\left|\hat{f}(\underline{x} \mid \underline{\theta}_f)\right| \;+\; |f(\underline{x})|\right. \\
& & \left.\qquad +\; \left|\hat{g}(\underline{x} \mid \underline{\theta}_g) u_c\right| \;+\; |g(\underline{x})\, u_c|\right) \;-\; 2\,\underline{e}^T P \underline{b}_c\, g(\underline{x})\, u_s
\end{array}\right. \tag{1.74}
$$

Bibliographie

[BERTOLD, 1994] M. R. BERTOLD. A time delay radial basis function network for phoneme recognition. In *Proceedings of the international conference on Neural Networks*, Orlando, 1994. Intel Corporation, Santa Clara.

[BOUCHON-MEUNIER, 1995] B. BOUCHON-MEUNIER. *La Logique Floue et ses applications*. Vie Artificielle. Addison-Wesley, 1995.

[DAVALO et NAÏM, 1993] E. DAVALO et P. NAÏM. *Des Réseaux de Neurones*. EYROLLES, 1993.

[HARP et SAMAD, 1992] S. HARP et T. SAMAD. *Genetic synthesis of neural network architecture*. L. DAVIS. Van Nostrand Reinhold, New-York, 1992.

[KOSKO, 1992a] B. KOSKO. Fuzzy systems are universal approximators. In *1st International IEEE Conference on Fuzzy Systems*, pages 1153–1162, 1992. San Diego.

[KOSKO, 1992b] B. KOSKO. *Neural Networks and Fuzzy Systems : A Dynamical Approach to Machnie Intelligence*. Prentice Hall, 1992.

[MOODY et DARKEN, 1988] J. MOODY et C. DARKEN. Learning with localized receptive fields. In *1988 Connectionnist Models Summer School*, Carnegie Mellon University, 1988.

[NARENDRA et PARTHASARATHY, 1990] K. S NARENDRA et K. PARTHASARATHY. Identification and control using neural networks. In *IEEE Transactions On Neural Networks*, volume 1, pages 4–27, 1990.

[ORR,] M. J. L. ORR. Introduction to radial basis function networks.

[TANAKA *et al.*, 1995] K. TANAKA, M. SANO, et H. WATANABE. Modeling and control of carbon monoxide concentration using neuro-fuzzy technique. In *IEEE Transactions On Fuzzy Systems*, volume 3, pages 271–279, 1995.

[TONG-TONG, 1992] J. R. TONG-TONG. *La Logique Floue*. Edition Hermès, 1992.

[UNAR et MURRAY-SMITH,] M. ALI UNAR et D. J. MURRAY-SMITH. Radial basis function networks for ship steering control.

[WANG, 1995] L. X. WANG. *Adaptative Fuzzy System And Control : Design And Stability Analysis*. PRENTICE HALL, 1995.