



HAL
open science

Réduction de la complexité des contrôleurs flous : applications à la commande multivariable

Véronique Lacrose

► **To cite this version:**

Véronique Lacrose. Réduction de la complexité des contrôleurs flous : applications à la commande multivariable. Automatique / Robotique. INSA de Toulouse, 1997. Français. NNT : . tel-00010030

HAL Id: tel-00010030

<https://theses.hal.science/tel-00010030>

Submitted on 2 Sep 2005

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre: 438 - Année 1997

Thèse

préparée au

Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS

en vue de l'obtention du

Titre de Docteur de l'Institut National des Sciences Appliquées de Toulouse

Spécialité : Automatique et Informatique Industrielle

par

Véronique LACROSE

Ingénieur de l'Institut National des Sciences Appliquées de Toulouse

RÉDUCTION DE LA COMPLEXITÉ DES CONTRÔLEURS FLOUS : APPLICATION À LA COMMANDE MULTIVARIABLE.

Soutenue le 7 Novembre 1997 devant le jury:

Président et Rapporteur	L.	FOULLOY
Rapporteur	H.B.	VERBRUGGEN
Examineurs	H.	BERSINI
	S.	BOVERIE
	M.	JAMSHIDI
Directeur de thèse	A.	TITLI

Cette thèse a été préparée au LAAS-CNRS
7, Avenue du Colonel Roche, 31077 Toulouse Cedex 4

Rapport LAAS N° 97415

Avant-propos

Les travaux présentés dans ce manuscrit sont le fruit de trois années d'études menées au Laboratoire d'Analyse et d'Architectures des Systèmes du C.N.R.S., à Toulouse, au sein du groupe de recherche *Automatique Symbolique*.

Tout d'abord, je remercie les directeurs successifs du L.A.A.S., *Mr Alain Costes* et *Mr Jean-Claude Laprie*, pour m'avoir permis de travailler dans leur laboratoire et exprime ma profonde reconnaissance à *Mme Louise Travé Massuyès* pour m'avoir accueillie dans son groupe de recherche.

Je tiens à remercier vivement *Mr Laurent Foulloy*, Professeur à l'université de Savoie, et *Mr Henk Verbruggen*, Professeur à l'Université technique de Delft (Pays-Bas), pour l'honneur qu'ils m'ont fait en acceptant d'être les rapporteurs de cette thèse. Je remercie également, *Mr Hugues Bersini*, chargé de cours à l'Université libre de Bruxelles (Belgique), *Mr Serge Boverie*, chef de projet à Siemens Automotive (France), *Mr Mohamad Jamshidi*, Professeur à l'université du Nouveau Mexique (Etats-Unis), d'avoir accepté d'examiner ces travaux.

Je tiens à exprimer toute ma gratitude à mon Directeur de thèse, *Mr. André Tilli*, Professeur à l'I.N.S.A. de Toulouse, pour ses judicieux conseils, la confiance qu'il m'a témoignée, sans oublier ses qualités humaines.

J'adresse ma sympathie à toutes les personnes que j'ai pu côtoyer amicalement ces trois dernières années et qui ont toujours su maintenir une ambiance de travail chaleureuse. De peur d'en oublier certains, je ne m'aventurerais pas à tous les citer et, nul doute qu'ils sauront ici se reconnaître. Toutefois, j'accorde une mention spéciale à *Jamal*, pour son amitié et ses précieux conseils.

Je réserve une pensée toute particulière à mes parents et leur exprime ici toute ma gratitude pour leur soutien sans limite. Un grand merci à *Roger* pour sa présence et son aide de tous les jours. Qu'ils trouvent en ce mémoire le modeste témoignage de ma profonde reconnaissance.

Table des matières

Table des matières	5
Introduction	9
I Concepts de base	11
1 Notions de base de la logique floue	13
1.1 La théorie des ensembles flous	14
1.1.1 Différence entre ensemble flou et ensemble booléen	14
1.1.2 Variables linguistiques	14
1.1.3 Définitions	15
1.1.4 Opérations sur les ensembles flous	16
1.1.5 Le principe d'extension	17
1.2 Relations floues	18
1.2.1 Produit cartésien	18
1.2.2 Projection	19
1.2.3 Extension cylindrique	19
1.2.4 Composition de relations floues	20
1.3 Conclusion	20
2 La commande en logique floue	21
2.1 Un peu d'histoire	22
2.2 Acquisition de la connaissance et écriture de la base de règles	23
2.3 Propriétés de la base de règles	24
2.3.1 Continuité	24
2.3.2 Consistance	25
2.3.3 Complétude	26
2.4 Les différentes étapes de la commande floue	26
2.4.1 Mise en forme des entrées, normalisation	28
2.4.2 Fuzzification	29
2.4.3 Traitement des prémisses composées	30
2.4.4 Inférence floue	30

2.4.5	Agrégation des règles	31
2.4.6	Défuzzification	32
2.4.7	Dénormalisation	35
2.5	Structure de base d'un contrôleur flou: analogie structurelle avec les P.I.D.	35
2.6	Réglage, stabilité et robustesse d'un contrôleur flou	36
2.6.1	Réglage	36
2.6.2	Stabilité	37
2.6.3	Robustesse	39
2.7	Conclusion	39
 II Méthodologies		 41
1	Introduction	43
1.1	Commande floue multivariable: état de l'art	44
1.1.1	Loi de commande issue d'une information pauvre sur le processus	44
1.1.2	Loi de commande basée sur un modèle mathématique du processus	46
1.1.3	Loi de commande déduite d'un modèle flou du processus	47
1.2	Objectif du travail de thèse	48
2	Réduction de la complexité d'une base de règles	51
2.1	Elimination ou sélection de règles	52
2.1.1	Elimination de règles	52
2.1.2	Algorithme d'orthogonalisation des moindres carrés	52
2.2	Diminution du nombre de fonctions d'appartenance	60
2.2.1	Simplification de la base de règles par fusionnement d'ensembles flous similaires	60
2.2.2	Approximation du système flou initial	67
2.3	Conclusion	69
3	Vers la synthèse d'un contrôleur flou de complexité réduite	71
3.1	Réduction de la dimension du problème	72
3.1.1	Fusion mathématique	72
3.1.2	Fusion symbolique	81
3.2	Structuration de la loi de commande	86
3.2.1	Décentralisation	86
3.2.2	Hierarchisation	87
3.3	Combinaison de techniques	95
3.4	Conclusion	102

4	Auto-réglage du contrôleur flou	103
4.1	Apprentissage par modèle de référence	104
4.2	La procédure d'auto-réglage du contrôleur flou	106
4.2.1	Initialisation du contrôleur flou	106
4.2.2	Réglage des facteurs d'échelle	107
4.2.3	Optimisation des fonctions d'appartenance	108
4.3	Conclusion	113
5	Conclusion de la deuxième partie	115
 III Applications		 117
1	Application à un mélangeur	119
1.1	Présentation du système	120
1.2	Présentation du cahier des charges	121
1.3	Analyse qualitative du mélangeur	124
1.4	Réduction de la dimension du problème de commande	125
1.5	Structuration de la loi de commande	125
1.6	Analyse des résultats obtenus dans le cas d'une fusion mathématique	128
1.6.1	Evolution du critère quadratique C	128
1.6.2	Evolution des paramètres des contrôleurs flous	128
1.6.3	Résultats obtenus par rapport à la trajectoire de référence . . .	130
1.6.4	Comparaison avec une commande optimale	132
1.6.5	Analyse expérimentale de la robustesse	134
1.7	Analyse des résultats obtenus dans le cas d'une fusion symbolique .	135
1.7.1	Evolution du critère quadratique C	135
1.7.2	Evolution des paramètres du contrôleur flou hiérarchique . . .	136
1.7.3	Comparaison avec une commande optimale	137
1.7.4	Analyse expérimentale de la robustesse	139
1.8	Conclusion	140
2	Application à un bioréacteur	141
2.1	Présentation du procédé	142
2.2	Modélisation mathématique du procédé	142
2.2.1	Equations d'évolution du procédé	142
2.2.2	Modélisation des cinétiques de croissance et de dégradation . .	143
2.3	Présentation du cahier des charges	145
2.4	Définition d'une stratégie de commande floue	147
2.4.1	Analyse qualitative du procédé	148
2.4.2	Fusion mathématique et hiérarchisation	149
2.4.3	Auto-réglage du contrôleur flou	150
2.4.4	Analyse des résultats	153

2.4.5	Analyse expérimentale de la robustesse du contrôleur flou hiérarchique	155
2.5	Conclusion	157
Conclusion		159
Références bibliographiques		163
A	Calcul des gradients relatifs à un système flou de Sugeno d'ordre zéro	171
A.1	Contrôleur flou à une entrée	172
A.2	Contrôleur flou à deux entrées	173
A.3	Fonctions d'appartenance triangulaires	175
A.4	Fonctions d'appartenance bidimensionnelles	176

Introduction

Parmi les récents développements des techniques pour l'Automatique, l'introduction de nouveaux formalismes tels que la *logique floue*, a suscité un intérêt sans cesse croissant depuis les années soixante-dix. Il suffit de voir les nombreuses applications industrielles qui en découlent et de consulter l'abondante littérature sur le sujet pour s'en convaincre.

L'intérêt de la logique floue réside dans sa capacité à traiter l'imprécis, l'incertain et le vague. Elle est issue de la capacité de l'homme à décider et agir de façon pertinente malgré le *flou* des connaissances disponibles et a été introduite dans le but d'approcher le raisonnement humain à l'aide d'une représentation adéquate des connaissances. Aussi, le succès de la commande en logique floue trouve en grande partie son origine dans sa capacité à traduire une stratégie de contrôle d'un opérateur qualifié en un ensemble de règles linguistiques "si ... alors ..." facilement interprétables.

L'utilisation de la commande floue est particulièrement intéressante lorsqu'on ne dispose pas de modèle mathématique précis du processus à commander ou lorsque ce dernier présente de trop fortes non linéarités ou imprécisions. Cependant, si beaucoup de travaux relatifs à la commande de systèmes monovariables ont été reportés dans la littérature, on ne trouve, en comparaison, que très peu d'articles sur la commande floue multivariable. Ceci n'est guère surprenant étant donné que le nombre de règles est une fonction exponentielle des variables en entrée du contrôleur flou. Par conséquent, la synthèse de contrôleurs flous en vue de commander des systèmes multivariables, nécessite l'écriture d'un grand nombre de règles et le réglage d'un nombre impressionnant de paramètres. Les objectifs que nous nous sommes fixés sont donc, d'une part, la réduction de la base de règles, d'autre part, un outil d'aide à la conception de contrôleurs flous multivariables à travers la technique de descente du gradient.

Ce mémoire est constitué de trois grandes parties articulées comme suit :

La **première partie** introduit les concepts de base nécessaires à la bonne compréhension de cette thèse. Le *premier chapitre* est principalement consacré à la théorie des ensembles flous et traite également des relations floues. Ces quelques notions de base sont suffisantes pour aborder l'utilisation de la logique floue en commande, laquelle fait l'objet du *second chapitre*. Dans ce chapitre, le problème délicat de l'ac-

quisition des connaissances et de l'écriture des règles de commande floue est abordé. Les différentes étapes dans le traitement des règles floues y sont détaillées.

La **seconde partie** constitue le noyau dur de ce mémoire. Le *premier chapitre* dresse un rapide bilan de l'état actuel des recherches sur la commande floue multivariable et introduit la problématique et les objectifs du présent manuscrit. Le *second chapitre* est consacré aux principales méthodes permettant de réduire le nombre de règles d'une base donnée. Dans le *troisième chapitre*, le point de départ n'est plus la base de règles mais le procédé à commander et l'on cherche alors à définir un contrôleur flou tout en minimisant son nombre de règles. Le *quatrième chapitre* est consacré à un outil d'auto-réglage des contrôleurs flous à l'aide d'une technique de descente du gradient. Le *cinquième chapitre* conclut cette partie "Methodologies" et résume les différentes étapes que nous proposons, lesquelles permettent la synthèse d'un contrôleur flou de complexité réduite.

La **troisième partie** est consacrée à l'application de la méthodologie proposée dans cette thèse à des problèmes de commande multivariable. Dans le *premier chapitre* nous testons différentes techniques de réduction sur un exemple académique : un bac mélangeur multivariable à deux entrées et deux sorties. Dans le *second chapitre* nous nous intéressons à un problème d'actualité : un procédé biologique de traitement des eaux usées.

Bonne lecture ...

Première partie
Concepts de base

Chapitre 1

Notions de base de la logique floue

Ce chapitre n'a pas pour prétention de donner un état des lieux complet de la logique floue mais uniquement de fournir les quelques notions de base nécessaires à la bonne compréhension de cette thèse. Un lecteur déjà familier à ce domaine n'y apprendra certainement rien de plus.

Nous commençons ce chapitre par une présentation de la théorie des ensembles flous. Ainsi, le paragraphe 1.1.1 souligne la différence entre un ensemble booléen et un ensemble flou. Dans le paragraphe 1.1.2, nous pouvons trouver la définition d'une variable *linguistique*. Quelques définitions et opérations sur les ensembles flous sont respectivement données dans le paragraphe 1.1.3 et 1.1.4. Le principe d'extension, lequel constitue un des concepts de base de la théorie des ensembles flous et permet l'extension de concepts mathématiques aux ensembles flous, est abordé dans le paragraphe 1.1.5. Enfin, le second paragraphe traite des relations floues (paragraphe 1.2) et en particulier de la composition de relations floues (paragraphe 1.2.4) obtenue en utilisant les opérations de projection (paragraphe 1.2.2) et d'extension cylindrique (paragraphe 1.2.3).

1.1 La théorie des ensembles flous

Ce n'est qu'à partir de 1965 que L.A. Zadeh, professeur à l'université de Berkeley, jeta les bases de ce qu'il dénomma "fuzzy set" (ensemble flou), prenant ainsi en considération le problème posé par les connaissances imprécises ou vagues. La notion d'ensemble flou permet alors des graduations dans l'appartenance d'un élément à une classe, c'est à dire autorise un élément à appartenir plus ou moins fortement à cette classe.

1.1.1 Différence entre ensemble flou et ensemble booléen

Alors qu'un ensemble booléen est défini par sa fonction caractéristique f à valeurs 0 ou 1, un ensemble flou est défini par sa fonction d'appartenance μ à valeurs dans l'intervalle $[0,1]$.

Considérons l'exemple suivant :

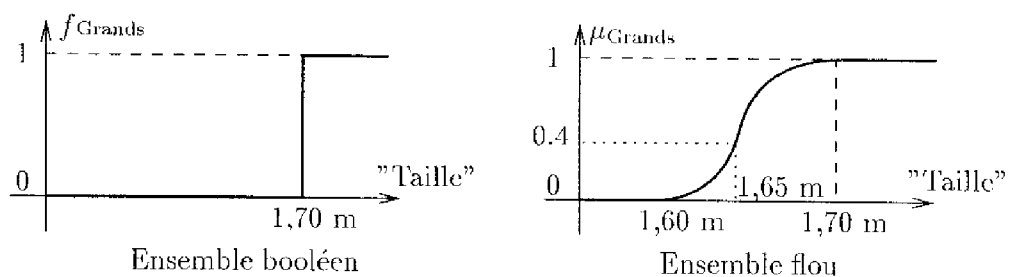


FIG. 1.1 – Représentation de "Grands" dans le cas booléen et flou

L'ensemble des tailles possibles d'un individu représente l'*univers de discours* de la variable "Taille". "Grands" est une *valeur linguistique* de cette variable.

Soit 1,65 m la taille de Jean. Dans le cas de l'ensemble booléen Jean n'appartient pas à la classe des "Grands". Dans le cas de l'ensemble flou Jean appartient à la classe des "Grands" avec un degré d'appartenance 0,4.

La notion d'ensemble flou évite l'utilisation arbitraire de limites rigides d'appartenance à des classes; il serait aberrant de considérer qu'un individu de 1,70 m est grand, mais qu'un individu de 1,695 m ne l'est pas.

1.1.2 Variables linguistiques

Une variable linguistique est représentée par un triplet (V, X, T_V) où V est la variable elle-même, X l'univers de discours et T_V l'ensemble des caractérisations floues de la variable V .

Considérons par exemple la variable *taille* définie sur l'ensemble des entiers positifs et caractérisée par les ensembles flous *petit*, *moyen* et *grand*. La variable *taille* est alors représentée par le triplet suivant : $\{taille, R^+, (petit, moyen, grand)\}$.

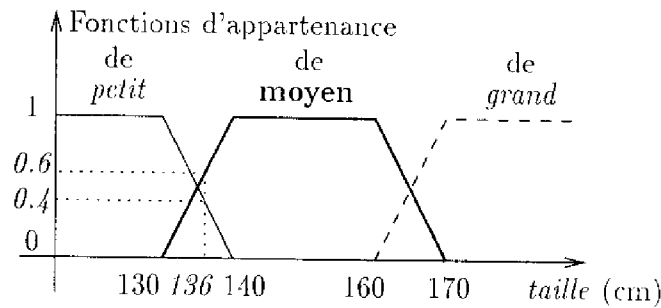


FIG. 1.2 – Une personne mesurant 136 cm est de taille petite avec un degré d'appartenance 0.4 et de taille moyenne avec un degré d'appartenance 0.6.

1.1.3 Définitions

Un ensemble flou A défini sur l'univers de discours X est noté :

$$A = \sum_{i=1}^n \frac{\mu_A(x_i)}{x_i} = \frac{\mu_A(x_1)}{x_1} + \dots + \frac{\mu_A(x_n)}{x_n} \quad \text{dans le cas discret,}$$

$$A = \int_X \frac{\mu_A(x)}{x} \quad \text{dans le cas continu.}$$

Caractéristiques d'un ensemble flou : ce sont essentiellement celles qui montrent dans quelle mesure l'ensemble flou diffère de l'ensemble booléen.

- Support: $\text{supp}(A) = \{x \in X \mid \mu_A(x) > 0\}$, c'est l'ensemble booléen des éléments de X qui appartiennent au moins un peu à l'ensemble flou.
- Hauteur: $h(A) = \sup_{x \in X} \{\mu_A(x)\}$, c'est le plus fort degré d'appartenance avec lequel un élément de X appartient à l'ensemble flou, ce dernier est dit normalisé si sa hauteur est de 1.
- Noyau: $\text{noy}(A) = \{x \in X \mid \mu_A(x) = 1\}$, c'est l'ensemble booléen de tous les éléments appartenant de façon absolue à l'ensemble flou.

Coupe de niveau α ou α -coupe : $\alpha\text{-coupe}(A) = A_\alpha = \{x \in X \mid \mu_A(x) \geq \alpha\}$, c'est l'ensemble booléen des éléments de X qui appartiennent à A avec un degré d'appartenance au moins égal à α .

Singleton flou $\{x\}$ de X : $\mu_{\{x\}}(x) = 1$ et $\mu_{\{x\}}(y) = 0$ pour tout $y \neq x$.

Ensembles flous convexes : Un ensemble flou A est convexe si

$$\forall x_1, x_2 \in X, \forall \lambda \in]0, 1], \mu_A(\lambda x_1 + (1 - \lambda)x_2) \leq \min(\mu_A(x_1), \mu_A(x_2))$$

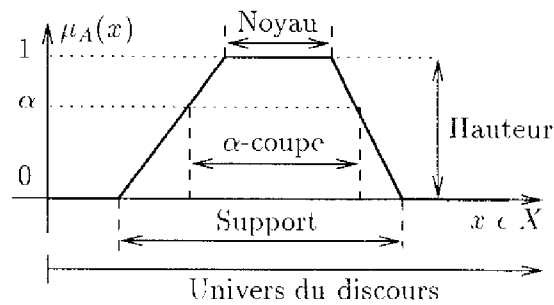
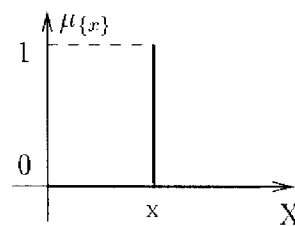
FIG. 1.3 – Support, hauteur, noyau, et α -coupe d'un ensemble flou

FIG. 1.4 – Singleton flou

Partition floue : N ensembles flous (A_1, \dots, A_N) définis sur l'univers de discours X forment une partition floue si

$$\forall x \in X, \sum_{i=1}^N \mu_{A_i}(x) = 1$$

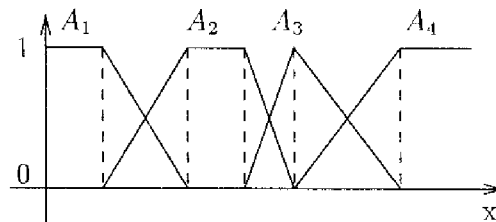


FIG. 1.5 – Exemple de partition floue

Une partition floue composée d'ensembles flous convexes normaux implique que pas plus de deux fonctions d'appartenance se recouvrent.

1.1.4 Opérations sur les ensembles flous

Nous définissons, ci-dessous, les notions d'intersection, d'union et de complémentation d'ensembles flous.

Soient A et B deux ensembles flous, μ_A et μ_B leur fonction d'appartenance respective et X leur univers de discours.

Intersection et Union : la généralisation des opérations booléennes aux ensembles flous n'est pas unique. Zadeh propose les opérateurs suivants :

$$\begin{aligned} \mu_{A \cap B}(x) &= \min(\mu_A(x), \mu_B(x)) \quad (\text{intersection}) \\ \mu_{A \cup B}(x) &= \max(\mu_A(x), \mu_B(x)) \quad (\text{union}) \end{aligned}$$

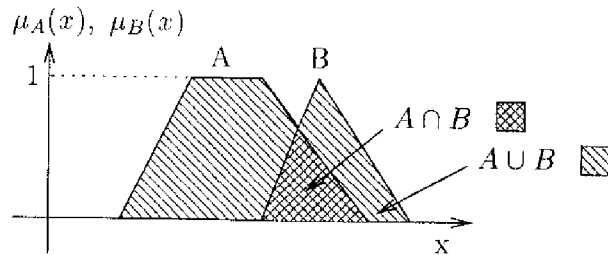


FIG. 1.6 – Exemple d'intersection et d'union d'ensembles flous (Zadeh)

Il existe d'autres opérateurs qui sont les normes triangulaires (t-normes) pour l'intersection et les conormes triangulaires (t-conormes) pour l'union.

	t-normes	t-conormes
Zadeh (1973)	$\min(x, y)$	$\max(x, y)$
Bandler et Kohout (1980)	$x.y$	$x + y - x.y$
Lukasiewicz, Giles (1976)	$\max(x + y - 1, 0)$	$\min(x + y, 1)$
Weber (1983)	$\begin{cases} x, & \text{si } y = 1 \\ y, & \text{si } x = 1 \\ 0, & \text{sinon} \end{cases}$	$\begin{cases} x, & \text{si } y = 0 \\ y, & \text{si } x = 0 \\ 1, & \text{sinon} \end{cases}$
Hamacher (1978) $\gamma > 0$	$\frac{x.y}{\gamma + (1 - \gamma)(x + y - x.y)}$	$\frac{x + y - (2 - \gamma)x.y}{1 - (1 - \gamma)x.y}$
Dubois et Prade (1986) $\alpha \in [0, 1]$	$\frac{x.y}{\max(x, y, \alpha)}$	$\frac{x + y - x.y - \min(x, y, 1 - \alpha)}{\max(1 - x, 1 - y, \alpha)}$

TAB. 1.1 – Principales t-normes et t-conormes

Complémentation : $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$ avec $A \cup \bar{A} \neq X$ et $A \cap \bar{A} \neq \emptyset$.

1.1.5 Le principe d'extension

Le principe d'extension a été introduit par L.A. Zadeh en 1975 et constitue l'un des concepts les plus importants de la théorie des ensembles flous. Il permet

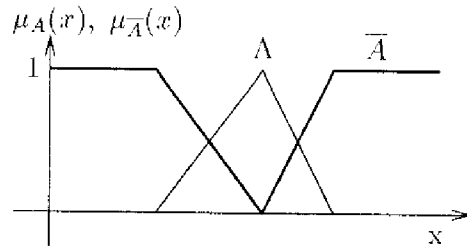


FIG. 1.7 - Exemple de complémentation d'un ensemble flou

l'extension d'une application f d'un premier ensemble de référence X vers un second ensemble de référence Y , éventuellement identique au premier. Il est alors possible de définir un ensemble flou B de Y à partir de la donnée d'un ensemble flou A de X :

$$\forall y \in Y, \quad \mu_B(y) = \begin{cases} \sup_{\{x \in X \mid y=f(x)\}} \mu_A(x) & \text{si } \{x \in X \mid y=f(x)\} \neq \emptyset \\ 0 & \text{sinon.} \end{cases}$$

1.2 Relations floues

Jusqu'à présent nous avons considéré des ensembles flous monodimensionnels. Lorsque ceux-ci sont multidimensionnels, leur fonction d'appartenance est aussi communément appelée relation floue.

Une relation floue R définie sur le produit cartésien $X_1 \times \cdots \times X_n$ est un ensemble flou (n -dimensionnel) et est noté :

$$R = \sum_{X_1 \times \cdots \times X_n} \frac{\mu_R(x_1, \cdots, x_n)}{(x_1, \cdots, x_n)} \quad \text{dans le cas discret,}$$

$$R = \int_{X_1 \times \cdots \times X_n} \frac{\mu_R(x_1, \cdots, x_n)}{(x_1, \cdots, x_n)} \quad \text{dans le cas continu.}$$

1.2.1 Produit cartésien

Considérons différents ensembles flous A_1, A_2, \cdots, A_n , respectivement définis sur X_1, X_2, \cdots, X_n . On peut définir, à partir de ces ensembles flous, un ensemble flou global multidimensionnel, $A = A_1 \times A_2 \times \cdots \times A_n$, considéré comme leur *produit cartésien*, de fonction d'appartenance :

$$\forall x = (x_1, \cdots, x_n) \in X, \quad \mu_A(x) = \min(\mu_{A_1}(x_1), \cdots, \mu_{A_n}(x_n))$$

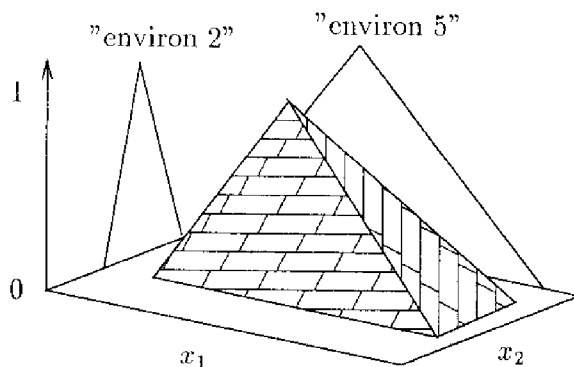


FIG. 1.8 – Produit cartésien de “environ 5” et “environ 2”

1.2.2 Projection

De façon inverse, à partir d'un ensemble flou multidimensionnel défini sur un univers complexe, il est possible, à l'aide d'une projection mathématique, d'obtenir des informations sur chacune des différentes composantes de cet univers.

Soit un ensemble flou A défini sur un univers $X = X_1 \times X_2$. La projection de A sur X_1 est définie par :

$$\forall x_1 \in X_1, \mu_{\text{proj}(A;X_1)}(x_1) = \sup_{x_2 \in X_2} [\mu_A(x_1, x_2)]$$

On définit de façon analogue la projection de A sur X_2 .

En particulier, si A est le produit cartésien des ensembles flous A_1 de X_1 et A_2 de X_2 , ses projections $\text{proj}(A;X_1)$ et $\text{proj}(A;X_2)$ sont respectivement A_1 et A_2 .

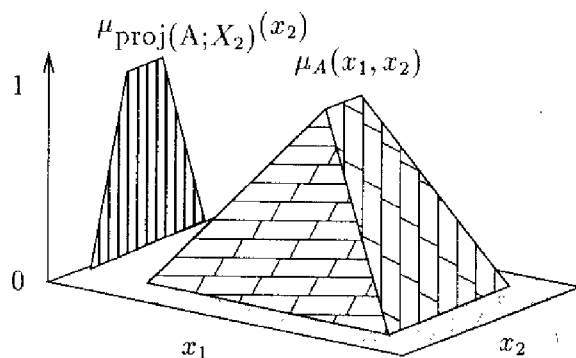


FIG. 1.9 – Exemple de projection d'un ensemble flou

1.2.3 Extension cylindrique

L'extension cylindrique de l'ensemble flou B de $X_a \times X_b \times \dots \times X_k$, avec $1 \leq a < b < \dots < k \leq n$, est l'ensemble flou $B^e = \text{extc}(B;X)$ de $X = X_1 \times X_2 \times \dots \times X_n$

de fonction d'appartenance :

$$\forall x = (x_1, \dots, x_n) \in X, \mu_{B^e}(x) = \mu_B(x_1, \dots, x_k)$$

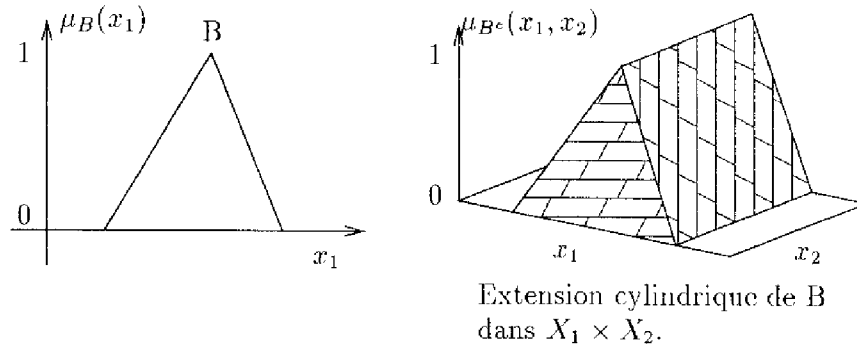


FIG. 1.10 - Exemple d'extension cylindrique

1.2.4 Composition de relations floues

Soit R une relation floue définie sur $X \times Y$ et A un ensemble flou de X ; l'ensemble flou B de Y est déduit de la composition de A par R :

$$\begin{aligned} B &= A \circ R \\ &= \text{proj} (R \cap \text{ext}_c(A, X \times Y); Y) \end{aligned}$$

En considérant l'extension cylindrique comme implicite, la composition de relations floues comporte deux phases : combinaison et projection. L.A. Zadeh propose l'opérateur de composition sup-min :

$$\forall y \in Y, \mu_B(y) = \sup_x \min(\mu_A(x), \mu_R(x, y))$$

1.3 Conclusion

Dans ce chapitre, nous nous sommes contentés d'introduire les notions de base de la logique floue, et ces connaissances sont suffisantes pour aborder l'utilisation de la logique floue en commande.

Chapitre 2

La commande en logique floue

Bien que la logique floue possède un champ d'application extrêmement vaste (commande, classification, aide à la décision, base de données imprécises, ...) nous nous intéressons ici qu'à son utilisation dans le cadre de la commande. Ce chapitre a donc pour objectif de présenter les principes de base de la commande en logique floue.

La commande est certainement le domaine de la logique floue où il existe le plus de réalisations effectives, en particulier industrielles, et il est intéressant d'en faire un rappel historique (paragraphe 2.1).

L'objectif de la commande floue est de traiter des problèmes de commande de processus, le plus souvent à partir des connaissances des experts ou d'opérateurs qualifiés travaillant sur le processus. L'acquisition de la connaissance et l'écriture des règles floues de commande font l'objet du paragraphe 2.2. Dans le paragraphe 2.3 sont citées les différentes propriétés d'une base de règles.

Les différentes étapes dans le traitement de ces règles sont exposées dans le paragraphe 2.4. La fuzzification, le traitement des prémisses composées, l'inférence floue, l'agrégation des règles et la défuzzification en sont les phases principales. La structure de base d'un contrôleur flou est donnée dans le paragraphe 2.5.

Enfin, dans un dernier paragraphe, nous abordons le problème du réglage (paragraphe 2.6.1), de la stabilité (paragraphe 2.6.2) et de la robustesse (paragraphe 2.6.3) d'un contrôleur flou.

2.1 Un peu d'histoire . . .

- 1965 L.A. Zadeh publie "Fuzzy Sets" [Zad65].
- 1965-1970 Tentatives pour modéliser des systèmes par des relations floues.
- 1968 "Fuzzy Algorithms" [Zad68]: L.A. Zadeh remarque qu'un humain sait garer une voiture plus facilement qu'un ordinateur manipulant les équations de la physique correspondante.
- 1971 Premières investigations de L.A. Zadeh dans l'utilisation des ensembles flous pour la représentation du langage naturel.
- 1973 Zadeh donne une première formalisation de règles floues "si . . . alors" et suggère une approche *système expert* de la commande: la loi de commande est remplacée par des règles floues. [Zad73]
- 1975 Mamdani et Assilian publient les premiers résultats en commande floue. [MA75]
- 1975-1980 Intense activité en commande floue en Europe. Applications à des processus industriels, au contrôle de trafic, . . .
- 1980-1987 Période de stagnation en Europe: les recherches s'arrêtent faute de crédit. Les japonais s'intéressent à la commande en logique floue.
- 1985 Sugeno publie l'application au pilotage d'une voiture en modèle réduit [SN85]. Première conférence internationale IFSA (International Fuzzy Systems Association).
- 1987 Conférence mondiale du flou à Tokyo au Japon. On découvre les travaux japonais et on visite le métro flou de Sendai.
- 1988 Poursuite des efforts japonais. De nombreuses applications de la commande floue, notamment en électroménager et pour le matériel audiovisuel, apparaissent. Création du pôle "Automatisation Intégrée" du Groupe de Recherche en Automatique du C.N.R.S. et du projet "Commande Symbolique et Neuromimétique".
- 1989 Etude réalisée par F. Grout, attaché pour la Science et la Technologie auprès de l'ambassade de France au Japon: "Intelligence artificielle au Japon - Effervescence sur le flou". Mission de scientifiques dirigée par H. Prade, directeur de recherche au C.N.R.S. à l'Institut de Recherche en Informatique à Toulouse dont le rapport s'intitule: "Le Boum du Flou au Japon".
- 1992 Première conférence IEEE (Institut of Electrical and Electronic Engineers) sur le flou à San-Diego aux Etats-Unis.
- 1992-1995 Projet européen "FALCON - Fuzzy ALgorithms for CONTROL".
- 1993 E.U.F.I.T.: "First European Congress on Fuzzy and Intelligent Techniques" à Aachen en Allemagne. Commande floue d'un hélicoptère sans pilote à l'Institut Technologique de Tokyo [SGB93].

- 1994 L'Observatoire Français des Techniques Avancées publie un rapport de synthèse sur la logique floue [OFT94].
- 1996-1998 Projet européen "FAMIMO - Fuzzy Algorithms for the control of Multi-Input Multi-Output processes".

2.2 Acquisition de la connaissance et écriture de la base de règles

L'algorithme de commande consiste en une collection de règles floues appelée base de règles. Plusieurs solutions peuvent être envisagées pour obtenir ces règles.

- L'extraction des connaissances d'opérateurs humains est certainement la méthode la plus utilisée pour la commande de systèmes complexes (cimenterie, four verrier, ...). Elle peut être obtenue directement sous forme de règles énoncées par des experts dans la commande du processus ou bien à partir d'un jeu de données entrées-sorties représentatif du comportement de l'opérateur humain. Dans le second cas, il s'agit de construire un modèle flou des actions de commande prises par l'opérateur [TS85].
- A l'image de la démarche usuelle en automatique, le contrôleur flou peut être obtenu à partir d'un modèle flou du processus (règles à conclusion polynômiale pour [TS85], modèle d'état flou pour [Ton80] ou [TW85]). L'inversion de ce modèle conduit directement à un contrôleur flou. Cependant, l'utilisation de ce modèle inverse en tant que contrôleur n'est envisageable que lorsque le système à commander est à minimum de phase. Dans le cas contraire, cela conduirait à une instabilité du système en boucle fermée. L'utilisation à la fois directe et indirecte [Bav96] (après inversion) du modèle flou du processus conduit à des lois de commande floue par modèle interne ou prédictive [BSV95].
- Les connaissances des automaticiens sur le comportement des processus vis-à-vis de certaines classes d'entrées (par exemple, les réponses temporelles) peuvent aussi être utilisées pour la réalisation de contrôleurs flous [Sug85, TM87]. Il existe également des règles d'équivalence entre les contrôleurs classiques de type P.I.D. et les contrôleurs flous [FG92]. Une table de règles très connue en régulation est celle de Mac Vicar-Whelan [VW76]. Ces contrôleurs sont des versions floues de contrôleurs usuels.

En commande, les règles floues utilisées sont généralement de la forme :

$$\text{SI } \underbrace{(X_1 \text{ est } A_1) \text{ et } (X_2 \text{ est } A_2)}_{\text{prémisse}} \text{ ALORS } \underbrace{(Y \text{ est } B)}_{\text{conclusion}}. \quad (2.1)$$

Lorsque les règles floues sont de sémantiques plus complexes, elles peuvent facilement s'écrire sous une forme plus simple du même type que précédemment (2.1). Ainsi, en pratique, des règles de la forme :

- "Si X_1 est A_1 Alors (Y est B_1 Sinon Y est B_2)" se décomposent en :

$$\left\{ \begin{array}{l} \text{"Si } X_1 \text{ est } A_1 \text{ Alors } Y \text{ est } B_1\text{"} \\ \text{OU} \\ \text{"Si } X_1 \text{ est } \bar{A}_1 \text{ Alors } Y \text{ est } B_2\text{"} \end{array} \right.$$

- Si " X_1 est A_1 Alors (Y est B_1 à moins que X_1 soit A_2)" se décomposent en :

$$\left\{ \begin{array}{l} \text{"Si } X_1 \text{ est } A_1 \text{ Alors } Y \text{ est } B_1\text{"} \\ \text{OU} \\ \text{"Si } X_1 \text{ est } A_2 \text{ Alors } Y \text{ est } \bar{B}_1\text{"} \end{array} \right.$$

- "Si X_1 est A_1 Alors (Y est B_1 Sinon (Y est B_2 Si X_1 est A_2))" se décomposent en :

$$\left\{ \begin{array}{l} \text{"Si } X_1 \text{ est } A_1 \text{ Alors } Y \text{ est } B_1\text{"} \\ \text{OU} \\ \text{"Si } X_1 \text{ est } \bar{A}_1 \text{ et } X_1 \text{ est } A_2 \text{ Alors } Y \text{ est } B_2\text{"} \end{array} \right.$$

- "Si X_1 est A_1 Alors (Y est B_1 Si X_1 est A_2)" s'écrivent aussi :

$$\text{"Si } X_1 \text{ est } A_1 \text{ et } X_1 \text{ est } A_2 \text{ Alors } Y \text{ est } B_1\text{"}$$

2.3 Propriétés de la base de règles

Une base de règles floues doit posséder certaines caractéristiques importantes afin d'assurer une commande correcte du système. La continuité, la consistance, et la complétude d'un ensemble de règles floues sont successivement définies [Jag95].

2.3.1 Continuité

Définition 2.1 *Un ensemble de règles floues "Si-Alors" est continu si toutes les règles de prémisses "adjacentes" ont des conclusions "adjacentes".*

La notion d'ensembles flous "adjacents" consiste à ordonner les ensembles flous sur leur univers de discours :

$$A_1 < A_2 < \dots < A_{i-1} < A_i < A_{i+1} < \dots$$

où A_{i-1} et A_i sont adjacents au même titre que A_i et A_{i+1} . Dans le cas d'une partition floue, les ensembles flous adjacents sont ceux qui se recouvrent (leur intersection est non nulle).

Les parties prémisses des règles sont dites "adjacentes" si elles contiennent les mêmes ensembles flous pour chacune des variables en entrée sauf pour l'une d'entre elles où les ensembles flous sont adjacents. Considérons, par exemple, l'ensemble suivant de règles:

$$R_k: \text{ Si } x_1 \text{ est } A_1^k \text{ et } x_2 \text{ est } A_2^k \text{ Alors } y \text{ est } B_k \quad (k = 1, \dots, r).$$

Les prémisses des règles R_k et $R_{k'}$, $k \neq k'$, sont adjacentes si l'une des deux conditions suivantes est vérifiée:

1. $A_1^k = A_1^{k'}$ et A_2^k et $A_2^{k'}$ sont adjacents.
2. $A_2^k = A_2^{k'}$ et A_1^k et $A_1^{k'}$ sont adjacents.

La base de règles est dite continue si, pour chacune des règles de prémisses adjacentes, les ensembles flous en sortie B_k et $B_{k'}$ le sont aussi.

2.3.2 Consistance

Un ensemble de règles floues "Si-Alors" est *consistant* s'il ne contient pas de contradictions.

Définition 2.2 *Un ensemble de règles floues "Si-Alors" est inconsistant s'il existe au moins deux règles de prémisses identiques mais de conclusions différentes.*

Un exemple très célèbre de base de règles inconsistante est celle utilisée pour la commande d'un robot devant éviter un obstacle, dans laquelle deux règles contradictoires coexistent:

$$\begin{cases} \text{Si } \textit{obstacle en face} \text{ Alors } \textit{aller à gauche}, \\ \text{Si } \textit{obstacle en face} \text{ Alors } \textit{aller à droite}. \end{cases}$$

En commande, si la partie prémisses des règles floues contient le connecteur "ET", il est très rare de trouver deux règles de prémisses identiques. Cependant, ce problème de consistance de la base de règles peut apparaître lorsque les connecteurs utilisés en partie prémisses sont des "OU" ou des opérateurs de complémentation [Jag95].

2.3.3 Complétude

Définition 2.3 Un ensemble de règles floues "Si-Alors" est complet si, quelle que soit la combinaison dans l'espace d'entrée, il existe une valeur de commande.

Une base de règles est *incomplète* s'il existe une situation de l'espace d'entrée pour laquelle aucune règle n'est activable. L'incomplétude de la base de règles peut conduire à une discontinuité indésirable dans la loi de commande. Une solution à ce problème est l'interpolation entre les règles floues [KII93a].

Soit la base de règles suivante :

$$R_k : \text{Si } x_1 \text{ est } A_1^k \text{ et } \dots \text{ et } x_n \text{ est } A_n^k \text{ Alors } y \text{ est } B_k \quad (k = 1, \dots, r).$$

Une mesure de complétude (\mathcal{C}) de la base de règles est donnée par [Jag95] :

$$\mathcal{C}(\underline{x}) = \sum_{k=1}^r \left\{ \prod_{i=1}^n \mu_{A_i^k}(x_i) \right\}$$

où $\underline{x} = (x_1, x_2, \dots, x_n)$ est le vecteur d'entrée.

- Si $\mathcal{C}(\underline{x}) = 0$, la base de règles est *incomplète*.
- Si $0 < \mathcal{C}(\underline{x}) < 1$, la base de règles est *sous-complète*.
- Si $\mathcal{C}(\underline{x}) = 1$, la base de règles est *strictement complète*.
- Si $\mathcal{C}(\underline{x}) > 1$, la base de règles est *sur-complète* (redondante).

La sous-complétude, la complétude stricte et la sur-complétude sont divers degrés de complétude de la base de règles.

La complétude d'une base de règles est stricte si :

1. pour chacune des variables d'entrée en partie prémisses, les ensembles flous correspondants forment une partition floue,
2. la base de règles énumère toutes les combinaisons possibles en partie prémisses.

2.4 Les différentes étapes de la commande floue

Un contrôleur flou est un système à base de connaissances particulier, utilisant un raisonnement en profondeur limitée, dans une procédure de chaînage avant des règles (activation des règles par les prémisses). Toutes les règles activables (prémisse caractérisée par un degré d'appartenance non nul) sont activées. On considère ensuite une "moyenne" sur le résultat de ces règles pour engendrer une décision finale.

On peut distinguer plusieurs étapes dans le traitement des règles. Un schéma représentatif peut être le suivant :

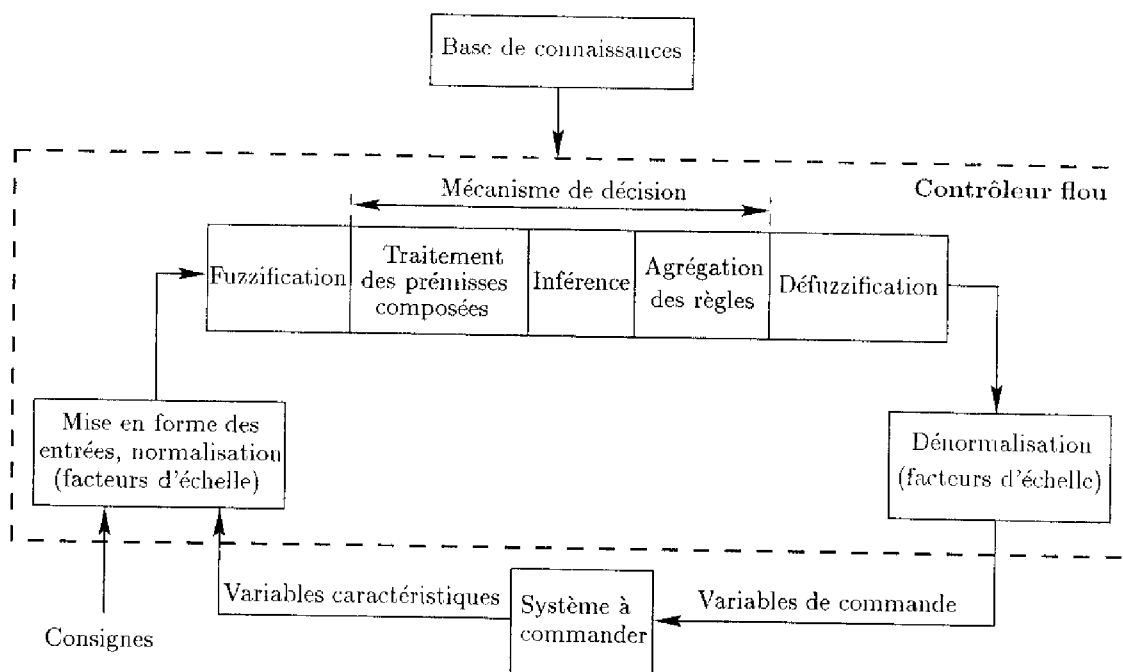


FIG. 2.1 - Structure de base d'un contrôleur flou

Les variables caractéristiques du système à commander et les consignes définissent les variables d'entrée du contrôleur flou. Les variables caractéristiques sont, en général, les grandeurs de sortie du processus et, le cas échéant, d'autres mesures déterminantes pour saisir l'évolution dynamique du processus. Les variables de sortie du contrôleur flou sont les commandes à appliquer au processus.

La base de connaissances est composée d'une base de données et d'une base de règles.

La base de données regroupe :

- les ensembles flous associés aux variables d'entrée et de sortie du contrôleur flou,
- les facteurs d'échelle (gains) en entrée (normalisation) et en sortie (dénormalisation).

La base de règles contient des règles de la forme :

“SI (X_1 est A_1 et X_2 est A_2) ALORS (Y est B)”

X_1 , X_2 et Y sont des grandeurs physiques caractéristiques du système et du problème de commande. A_1 et A_2 sont des labels linguistiques.

Suivant la nature de B , on parlera de :

- règles à conclusion symbolique (contrôleur de type Mamdani) : B est une valeur linguistique. Exemple : *Si l'erreur est "Négatif Moyen" et la variation de l'erreur est "Positif Petit" alors la commande est "Négatif Petit"*.
- règles à conclusion algébrique (contrôleur de type Sugeno) : B est une valeur numérique (singleton) ou une équation mathématique bien précise (non floue). Exemple : *Si l'erreur est "Négatif Moyen" et la variation de l'erreur est "Positif Petit" alors la commande est -0.3* . Lorsque B est une valeur numérique on parle de règles de Takagi-Sugeno "d'ordre zéro", sinon, de règles à conclusion polynômiale.

Bien que les sorties des contrôleurs flous de type Sugeno soient généralement des fonctions non linéaires statiques de leurs entrées, il ne faut pas oublier de mentionner les contrôleurs dits "flous dynamiques" de Sugeno, où B est un modèle dynamique, certain ou incertain, à temps continu ou discret. L'utilisation de tels contrôleurs permet d'étendre certains résultats de l'automatique classique à la commande floue.

On distingue classiquement trois parties dans la structure d'un contrôleur flou : la fuzzification, le mécanisme de décision et la défuzzification.

- La fuzzification est l'étape qui permet de transformer une grandeur mesurée sur le processus en un ensemble flou.
- Le mécanisme de décision permet de calculer l'ensemble flou associé à la commande.
- La défuzzification est l'étape qui permet de transformer l'ensemble flou, obtenu par le calcul précédent, en une grandeur de commande à appliquer au processus.

Les opérations de normalisation et de dénormalisation sont des étapes optionnelles.

Nous allons maintenant revenir en détail sur les différentes étapes dans le traitement des règles.

2.4.1 Mise en forme des entrées, normalisation

Cette première étape permet le traitement des variables d'entrée du contrôleur flou, par exemple, calcul d'erreurs (différences entre grandeurs mesurées et consignes) et de variations d'erreurs. L'utilisation de domaines normalisés (univers de discours compris entre -1 et 1) nécessite une transformation d'échelle. Celle-ci est réalisée par l'intermédiaire de facteurs d'échelle transformant les grandeurs physiques des entrées en des valeurs normalisées appartenant à l'intervalle $[-1, 1]$.

2.4.2 Fuzzification

C'est l'opération de *projection* de variables physiques réelles sur des ensembles flous caractérisant les valeurs linguistiques prises par ces variables. Deux cas peuvent se présenter selon que la mesure d'une variable physique réelle est précise (valeur numérique) ou pas (issue par exemple d'un capteur flou [FBM93]) :

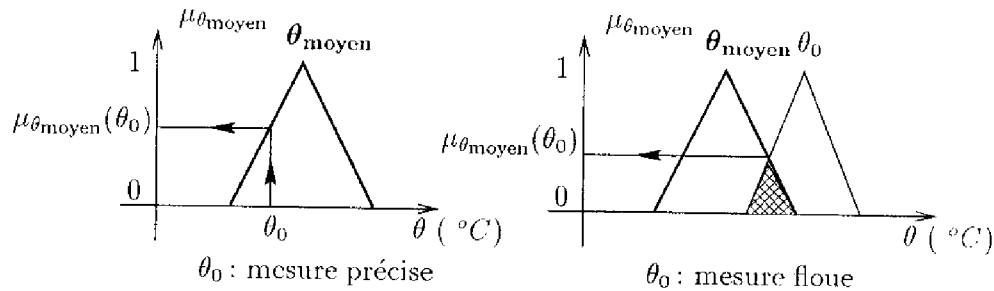


FIG. 2.2 - Conversion numérique-symbolique

Le choix de la forme des fonctions d'appartenance (triangulaires, trapézoïdales, exponentielles, gaussiennes, ...) est arbitraire. Des études comparatives ont montré, selon différentes formes de fonctions d'appartenance, des résultats pratiquement similaires en boucle fermée mais les formes triangulaires facilitent la programmation ce qui explique qu'elles soient le plus fréquemment utilisées. Quant au nombre de fonctions d'appartenance, il est généralement impair car elles se répartissent autour de zéro (3, 5 et 7 sont des valeurs courantes). Un exemple de fonctions d'appartenance triangulaires est donné dans la figure 2.3.

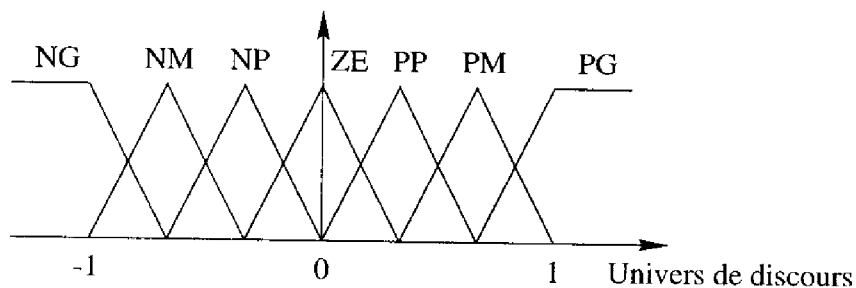


FIG. 2.3 - Exemple de fonctions d'appartenance triangulaires

NG, NM, ..., PG sont des valeurs linguistiques, avec :

- NG : "Négatif Grand"
- NM : "Négatif Moyen"
- NP : "Négatif Petit"

- ZE: "Zéro"
- PP: "Positif Petit"
- PM: "Positif Moyen"
- PG: "Positif Grand"

2.4.3 Traitement des prémisses composées

En général, les prémisses des règles vont comporter plusieurs clauses liées par des connecteurs "ET", "OU" et "NON". Dans la pratique, pour les opérations de conjonction et de disjonction, on a souvent recours, parmi les normes et conormes triangulaires, aux opérateurs min et max. Quant à la négation A^C d'un ensemble flou A , elle est caractérisée par $\mu_{A^C}(x) = 1 - \mu_A(x)$.

Des modificateurs linguistiques peuvent aussi être utilisés dans l'écriture de ces règles, par exemple :

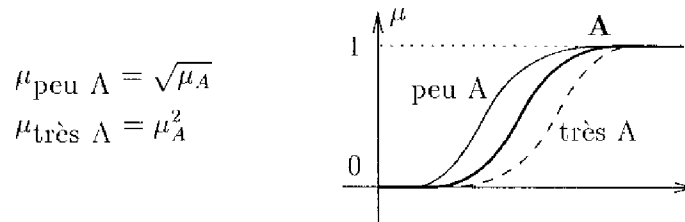


FIG. 2.A – Exemple de modificateurs linguistiques

2.4.4 Inférence floue

Elle repose sur l'utilisation d'un opérateur d'implication, lequel permet d'évaluer un degré de vérité d'une règle R de la forme "Si X est A alors Y est B ". En d'autres termes, cet opérateur quantifie la force de la liaison entre la prémisse et la conclusion de la règle.

Il existe de nombreux et différents opérateurs d'implication selon l'interprétation logique que l'on donne à l'implication "A implique B" ($A \rightarrow B$). On distingue ainsi l'implication classique où "A implique B" est définie par "Non A ou B" ($\bar{A} \cup B$) de l'implication dite conjonctive où "A implique B" est définie par "A et B" ($A \cap B$).

Les opérateurs les plus courants en commande sont de type conjonctif :

$$\begin{cases} \text{l'implication de Mamdani (1974):} & \mu_R(x, y) = \min(\mu_A(x), \mu_B(y)), \\ \text{l'implication de Larsen (1980):} & \mu_R(x, y) = \mu_A(x) \cdot \mu_B(y). \end{cases}$$

Soit la règle: "Si la cylindrée du véhicule est importante, alors sa consommation en carburant est élevée". Cette règle doit pouvoir être utilisée pour un véhicule donné dont on connaît précisément la cylindrée, qui n'est pas forcément typique de la caractérisation "importante", et doit également fournir une conclusion relative à sa consommation en carburant si la cylindrée du moteur n'est que relativement importante, par exemple. Dans la logique classique, le modus ponens ne permettrait d'obtenir une conclusion que si l'on savait exactement que la cylindrée du véhicule considéré est importante. Il faut donc modifier le modus ponens pour atteindre la souplesse de raisonnement souhaitée.

Le modus ponens trouve son équivalent flou dans le cadre du raisonnement approximatif sous la forme du *modus ponens généralisé* que nous présentons formellement ci-après.

	<i>Logique classique</i>	<i>Logique floue</i>
Fait	X est A	X est A'
Règle	Si X est A alors Y est B	Si X est A alors Y est B
Déduction	Y est B	Y est B'

La logique floue permet ainsi de déduire un nouveau fait B' caractérisé par :

$$\mu_{B'}(y) = \sup_{x \in X} T(\mu_{A'}(x), \mu_R(x, y))$$

pour une t-norme T appelée *opérateur de modus ponens généralisé*.

En commande, l'opérateur de Zadeh est couramment utilisé :

$$\mu_{B'}(y) = \sup_{x \in X} \min(\mu_{A'}(x), \mu_R(x, y))$$

Si A' est un singleton x_0 , alors : $\mu_{B'}(y) = \sup_{x \in X} \min(\mu_{A'}(x), \mu_R(x, y)) = \mu_R(x_0, y)$.

Pour l'implication de Mamdani, on obtient : $\mu_{B'}(y) = \min(\mu_{A'}(x_0), \mu_R(x_0, y))$.

2.4.5 Agrégation des règles

Selon le type de l'implication, classique ou conjonctive, l'opérateur utilisé pour agréger les règles est, respectivement, de type conjonctif ou disjonctif. Ainsi, en commande, l'implication étant généralement de type conjonctif, cela revient à considérer que les règles sont liées par un opérateur OU. En pratique, on utilise l'opérateur max :

$$\mu_{B'}(y) = \max_{i=1 \text{ à } n} \mu_{B'_i}(y)$$

Avant de décrire la prochaine étape consistant à transformer le résultat de l'agrégation des règles (ensemble flou décrit par la fonction d'appartenance $\mu_{B'}(y)$) en une

valeur précise de commande, considérons d'abord un exemple classique illustrant le raisonnement de Mamdani.

L'exemple illustré par la figure 2.5 considère les deux règles suivantes :

R_1 : Si (x_1 est ZE et x_2 est ZE) alors y est ZE

R_2 : Si (x_1 est PP et x_2 est PP) alors y est PP

La méthode de Mamdani repose sur l'utilisation de l'opérateur min pour la combinaison des prémisses et pour l'implication. Chaque règle est activée séparément et les conclusions sont agrégées pour définir l'ensemble flou associé à la variable de sortie y . L'agrégation des règles est réalisée par l'opérateur max.

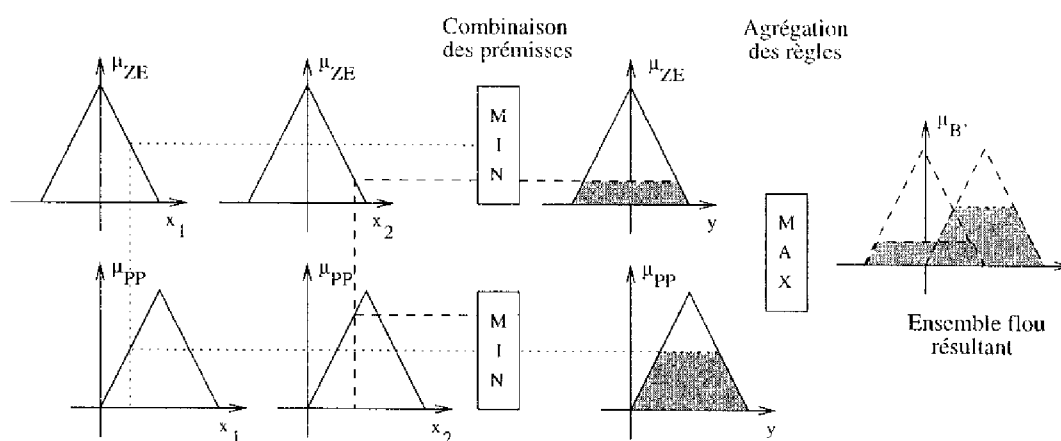


FIG. 2.5 – Illustration de la méthode de Mamdani

2.4.6 Défuzzification

La défuzzification consiste à transformer l'ensemble flou résultant de l'agrégation en une grandeur de commande précise. Là aussi il existe plusieurs méthodes [DHR93], parmi lesquelles :

- la méthode de la hauteur,
- le premier des maxima,
- le dernier des maxima,
- la moyenne des maxima,
- le centre de gravité,
- le centre des aires,

- le centre de la plus grande surface,
- le centre des maxima.

Les méthodes de défuzzification les plus utilisées en commande floue sont le centre de gravité, le centre des aires et le centre des maxima.

La méthode de la hauteur et ses variantes

La méthode de la hauteur consiste à choisir comme grandeur de commande la valeur du maximum.

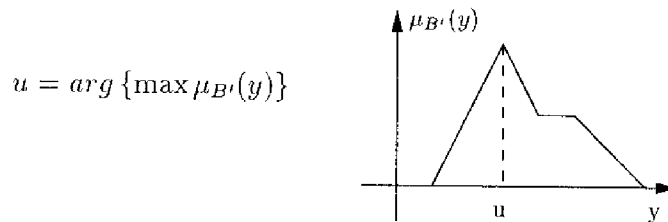


FIG. 2.6 – Défuzzification par la méthode de la hauteur

Dans le cas où la fonction d'appartenance $\mu_{B'}(y)$ a plus d'un maximum, on a le choix entre le premier, le dernier ou la moyenne des maxima. La moyenne des maxima est similaire à la méthode du premier ou du dernier des maxima et consiste tout simplement à en faire la moyenne.

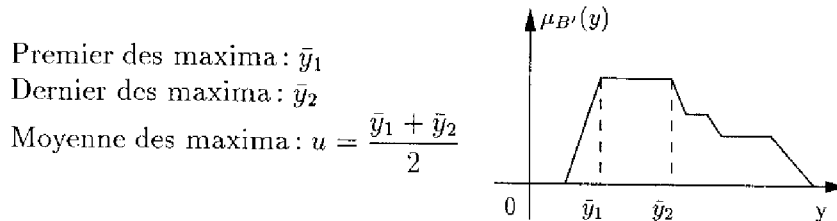


FIG. 2.7 – Variantes de la méthode de la hauteur

Ces méthodes nécessitent peu de calculs mais peuvent introduire des discontinuités dans la loi de commande, ce qui explique pourquoi elles sont si peu utilisées en commande floue.

Le centre de gravité

C'est la méthode de défuzzification la plus connue en commande floue. Cette méthode fournit intuitivement la valeur la plus représentative de l'ensemble flou issu de l'agrégation des règles. C'est aussi la méthode la plus coûteuse en temps calcul.

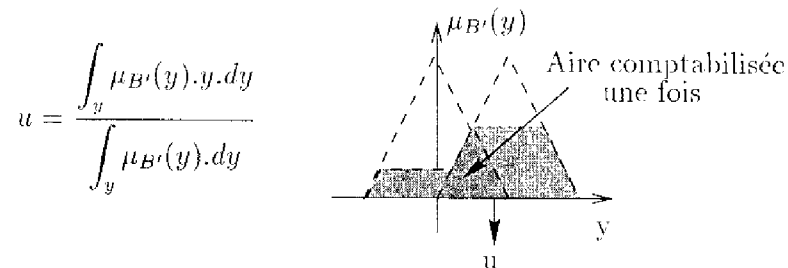


FIG. 2.8 – Défuzzification par un centre de gravité

Le centre des aires (ou centre des sommes)

Cette méthode est similaire à la précédente mais ne nécessite pas le calcul de $\mu_{B'}(y)$. L'idée est de considérer la contribution de chaque aire individuellement. L'ensemble B' est alors construit à partir de la somme de chaque aire. Ainsi, les aires qui se recouvrent, si elles existent, sont comptabilisées plus d'une fois.

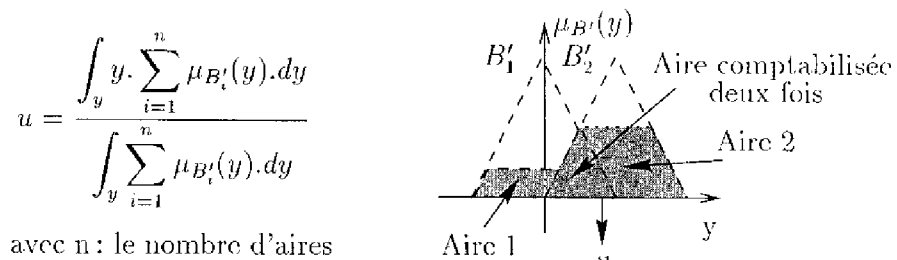


FIG. 2.9 – Défuzzification par un centre des aires

Le centre de la plus grande surface

Cette méthode consiste à calculer le centre de gravité de la plus grande surface.

Le centre des maxima

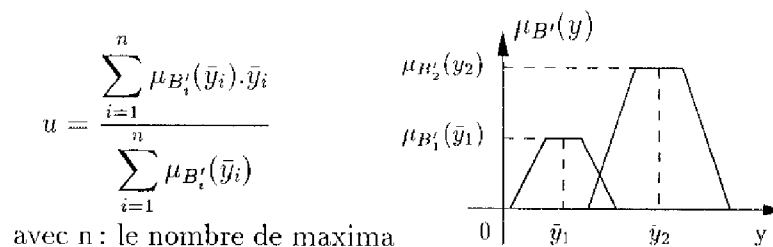


FIG. 2.10 – Défuzzification par un centre des maxima

2.5. Structure de base d'un contrôleur flou : analogie structurelle avec les P.I.D. 35

Cette méthode considère le maximum de chacune des contributions et en fait la moyenne pondérée.

Un cas particulier : méthode de Sugeno

Dans le cas de règles à conclusions polynômiales (règles de Takagi-Sugeno) du type :

R_i : Si (x_1 est A_1^i et x_2 est A_2^i et \dots x_n est A_n^i) Alors (y est $f_i(x_1, x_2, \dots, x_n)$)

la commande u est obtenue par une simple moyenne pondérée selon les niveaux d'activation α_i de chacune des règles R_i ($i = 1, \dots, r$) :

$$u = \frac{\sum_i^r \alpha_i \cdot f_i(x_1, x_2, \dots, x_n)}{\sum_i^r \alpha_i}$$

avec $\alpha_i = T(\mu_{A_1^i}(x_1), \mu_{A_2^i}(x_2), \dots, \mu_{A_n^i}(x_n))$

Pour la t-norme T , on choisit très souvent l'opérateur min ou le produit.

2.4.7 Dénormalisation

Cette dernière étape transforme les valeurs normalisées des variables de commande en des valeurs appartenant à leur domaine physique respectif.

2.5 Structure de base d'un contrôleur flou : analogie structurelle avec les P.I.D.

Pour des problèmes de régulation monovariante simples, les entrées du contrôleur flou sont généralement l'erreur ϵ (différence entre consigne et sortie du processus) et la variation de l'erreur $\Delta\epsilon$ (traduction de la dynamique du système).

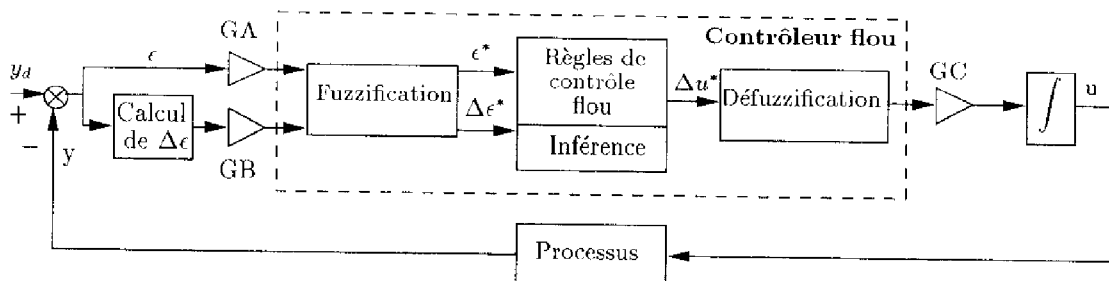


FIG. 2.11 -- Structure de base d'un contrôleur flou

GA, GB et GC sont des gains ou facteurs d'échelle.

Selon que la sortie du contrôleur flou concerne la commande ou sa variation, on obtiendra des équivalents structuraux non linéaires des contrôleurs classiques P.D., P.I., ou même des P.I.D., en augmentant le nombre d'entrées des contrôleurs.

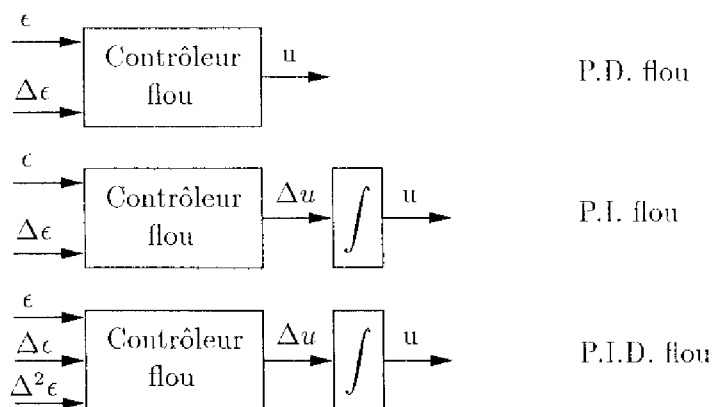


FIG. 2.12 - *Équivalents flous des contrôleurs usuels*

2.6 Réglage, stabilité et robustesse d'un contrôleur flou

2.6.1 Réglage

Un contrôleur flou possède de nombreux paramètres de réglage, ce qui peut, à priori, effrayer ses utilisateurs potentiels. En effet, contrairement aux contrôleurs classiques (notamment de type P.I.D. : 3 constantes à régler), le contrôleur flou possède un nombre bien plus conséquent de paramètres, et offre, par la même, davantage de degrés de liberté.

On peut citer parmi les choix et les réglages à faire :

- l'expression des règles,
- la définition des variables et des valeurs linguistiques (sous-ensembles flous pour chaque variable), avec leurs fonctions d'appartenance associées,
- la méthode d'implication,
- la méthode d'inférence,
- la méthode de défuzzification,

- les facteurs d'échelle (gains) sur les entrées et les sorties du contrôleur.

Cependant, on constate une certaine insensibilité du résultat au choix des méthodes d'implication, d'inférence et de défuzzification. Pour les autres paramètres, un réglage séquentiel est possible :

- facteurs d'échelle (effet global sur le comportement du système),
- fonctions d'appartenance,
- règles (effet local sur le comportement du système).

Le réglage par essais successifs de ces nombreux paramètres étant assez long et fastidieux, diverses techniques d'auto-réglage, d'optimisation et d'apprentissage ont été développées ces dernières années. On peut citer, à titre d'exemple, les techniques de la programmation mathématique, les réseaux neuronaux, les algorithmes génétiques.

2.6.2 Stabilité

L'analyse de la stabilité est a priori difficile dans une approche de type "système à base de connaissances" dont l'objectif est d'éviter l'utilisation de modèles mathématiques du procédé. Une modélisation floue du système bouclé est alors nécessaire pour l'analyse de la stabilité. Toutefois, si une caractérisation mathématique du processus à commander existe, le contrôleur flou étant, en général, un contrôleur non linéaire, certains résultats de la théorie des systèmes non linéaires peuvent être utilisés. On distingue alors deux grandes familles d'approches :

- *la méthode directe de Lyapunov* : cette méthode permet d'affirmer la stabilité asymptotique d'un état d'équilibre s'il existe une fonction de Lyapunov $V(x, t)$ définie positive telle que sa dérivée $\frac{dV(x, t)}{dt}$ soit définie négative.
- *l'approche entrée-sortie* : cette approche permet d'affirmer la stabilité d'un système si la sortie de celui-ci, en réponse à une entrée bornée, est elle-même bornée.

Nous choisissons ici de distinguer les différentes méthodes d'analyse de la stabilité selon que l'on dispose du modèle mathématique du processus ou pas. La liste des méthodes ici citées est non exhaustive et le lecteur peut se référer à [BO94, MT94] pour en avoir un aperçu plus complet.

Il n'existe pas de modèle mathématique du processus

Dans ce cas, le système en boucle fermée est modélisé par un système flou. Les méthodes d'analyse de la stabilité dépendent alors du type du modèle.

- Si le modèle du système en boucle fermée est un modèle symbolique (de type Mamdani ou modèle aux relations floues) du type temps discret, la stabilité n'est pas étudiée au sens classique du terme et de nouvelles définitions de la stabilité sont introduites. On parle alors de la stabilité d'un symbole flou [Ton78, KGN85, JL93].
- Dans le cas où le système en boucle fermée est modélisé par un système flou de type Sugeno, des propriétés de stabilité asymptotique peuvent être démontrées à partir de l'utilisation de fonctions de Lyapunov quadratiques [TS92].
- Dans le cas où le modèle du système en boucle fermée est un modèle hybride [MT97] (les ensembles flous en partie conclusion des règles dépendent des entrées) on utilise également des fonctions de Lyapunov quadratiques pour étudier la stabilité. Les modèles linguistiques et modèles de Sugeno sont deux cas particuliers de modèles hybrides.

Il existe un modèle mathématique du processus à commander

Dans ce cas, on peut faire appel à diverses méthodes d'analyse de la stabilité issues de la théorie des systèmes non linéaires. Ces méthodes diffèrent selon que le contrôleur flou a une expression mathématique simple ou pas.

- Si le contrôleur flou peut facilement se mettre sous forme analytique [MV94] l'on est ramené à l'étude de la stabilité asymptotique d'un système en boucle fermée non linéaire et on peut alors utiliser la méthode directe de Lyapunov.
- Pour des contrôleurs flous plus complexes, l'expression mathématique étant inexploitable, on doit alors se contenter d'une propriété de secteur géométrique [RM84, Opi93, Boh94]: le contrôleur flou est considéré comme une fonction mathématique non linéaire évoluant dans un secteur géométrique bien défini. On cherchera alors les conditions de stabilité absolue correspondantes à ce secteur, lesquelles peuvent s'exprimer soit dans un formalisme espace d'état (méthode directe de Lyapunov) soit dans un formalisme entrée-sortie [Vid93].

Les conditions les plus générales de stabilité sont obtenues par [Zam66] et ne nécessitent aucune hypothèse sur la nature du système (linéaire, non linéaire, variant, invariant, de dimension finie ou infinie). Cependant, bien que générales, ces conditions sont uniquement suffisantes et difficilement exploitables. C'est pourquoi le système est généralement supposé linéaire invariant de dimension finie bouclé par une non-linéarité (ici, le contrôleur flou). Selon les propriétés géométriques de cette non-linéarité, différents critères de stabilité

sont obtenus : critère du cercle, critère de conicité, lemme borné réel, lemme positif réel.

2.6.3 Robustesse

La robustesse, bien que maintes fois constatée, ne possède aucune démonstration rigoureuse. Son explication est à chercher dans l'aspect non linéaire de cette commande qui peut apparaître comme un contrôleur de type P.I.D. à gains variables ou pré-programmés, selon le point de fonctionnement. Toutefois, si le modèle du système en boucle fermée est un modèle "flou dynamique" de Sugeno, on peut se reporter aux travaux de l'automatique classique.

2.7 Conclusion

Dans ce chapitre, nous nous sommes intéressés à l'utilisation de la logique floue en commande. L'accent a particulièrement été mis sur les différentes étapes dans le traitement des règles d'un contrôleur flou.

Retenons, que l'intérêt majeur de la logique floue en commande réside dans sa capacité à traduire une stratégie de contrôle d'un opérateur qualifié en un ensemble de règles linguistiques facilement interprétables.

Les contrôleurs flous se distinguent selon la conclusion de leurs règles : symbolique (contrôleurs de type Mamdani) ou algébrique (contrôleurs de type Sugeno). Dans le cadre de notre étude, pour faciliter la réalisation pratique du contrôleur flou et pour limiter les temps de calcul, le contrôleur flou utilisé est de type Sugeno d'ordre zéro (singletons flous dans la partie conclusion des règles).

Nous disposons maintenant des notions élémentaires relatives à la commande en logique floue. Dans le chapitre suivant, on s'intéressera à la commande floue multivariable et on introduira la problématique abordée dans ce mémoire.

Deuxième partie

Méthodologies

Chapitre 1

Introduction

L'objectif de ce chapitre est, d'une part de dresser un rapide bilan de l'état actuel des recherches sur la commande floue multivariable (paragraphe 1.1), d'autre part, d'introduire les objectifs du présent mémoire (paragraphe 1.2).

Les problèmes de commande en logique floue de systèmes multivariables sont abordés différemment selon la richesse de l'information disponible sur le procédé. Nous avons choisi de classer les différentes approches selon les trois grands axes suivants :

1. Approche boîte noire (à partir de l'expertise) ou boîte grise (à partir d'une information pauvre sur le processus) (paragraphe 1.1.1),
2. Loi de commande floue hybride utilisant le modèle mathématique du processus (paragraphe 1.1.2),
3. Loi de commande issue d'un modèle flou du processus (paragraphe 1.1.3).

Cette synthèse des différentes approches de la commande floue multivariable n'est pas exhaustive mais permet de nous sensibiliser aux problèmes liés à l'explosion de la base de règles. En effet, bien que la commande en logique floue de processus multivariables semble être une approche prometteuse, la synthèse de tels contrôleurs reste difficile et non immédiate. Elle nécessite l'écriture d'un grand nombre de règles et le réglage d'un très grand nombre de paramètres et ce, d'autant plus que le nombre de variables du système est élevé. Ce problème de l'explosion de la base de règles constitue la motivation principale de cette thèse.

Les objectifs de ce travail de thèse sont introduits dans le paragraphe 1.2.

1.1 Commande floue multivariable : état de l'art

On se trouve de plus en plus souvent confrontés au problème de pilotage de systèmes multivariables. Ces derniers se révèlent difficiles à commander avec la théorie de l'automatique classique lorsqu'on ne dispose pas de modèle mathématique précis du processus ou lorsque ce dernier présente de trop fortes non linéarités ou imprécisions. La commande en logique floue de tels processus semble être une approche prometteuse mais, contrairement à la commande floue monovariable pour laquelle il existe une floraison d'articles, on ne trouve que très peu d'études sur la commande floue multivariable.

Dans le cas où le processus multivariable présente de faibles couplages, la synthèse d'un contrôleur flou multivariable peut se ramener à la conception de contrôleurs flous monovariables, chacun étant conçu pour une paire entrée-sortie. Dans le cas contraire, la synthèse des lois de commande est beaucoup plus délicate et reste un des problèmes ouverts de la commande floue.

On distingue ici trois grandes classes de méthodes. La première classe nécessite que très peu d'information sur le processus. La seconde classe d'approches est basée sur le modèle mathématique du processus et conduit à des lois de commande hybrides. Enfin, la dernière classe de méthodes est basée sur l'utilisation d'un modèle flou du processus à commander.

1.1.1 Loi de commande issue d'une information pauvre sur le processus

Analyse experte

La méthode basée sur la connaissance d'un expert est difficile à utiliser pour la commande multivariable. En effet, le nombre de règles nécessaires à la commande de tels systèmes augmente exponentiellement avec le nombre de variables. Faire citer toutes ces règles à un expert devient un exercice très difficile. Une première alternative consiste à identifier en ligne le comportement d'un opérateur qualifié travaillant sur le processus [TS85, Lee90]. Une seconde alternative consiste à analyser qualitativement le comportement du système à commander [RBT94]. Cette analyse qualitative souligne les différents couplages présents dans le processus et donne une première idée sur la stratégie de commande à adopter (structure du contrôleur flou, variables d'entrée et de sortie, règles de commande, etc). Le réglage des différents paramètres du contrôleur flou est alors effectué itérativement par une procédure d'essais et d'ajustements.

Pour des systèmes multivariables complexes ou de grande dimension, cette démarche experte devient très rapidement laborieuse et nécessite beaucoup de rigueur ... et de patience. Il est donc nécessaire de se tourner vers d'autres méthodes, plus automatiques, de conception de contrôleurs flous.

Commande floue adaptative

Il s'agit de régler automatiquement les paramètres du contrôleur dans l'objectif de satisfaire un critère - tel que la stabilité, ou la convergence vers une sortie désirée. La détermination de la loi d'ajustement est basée sur des théories bien définies telles que la méthode du gradient ou la théorie de la stabilité de Lyapunov [Wan94]. Cependant, il n'est pas possible d'assurer que l'erreur tend vers zéro, mais seulement qu'elle reste bornée.

Auto-réglage

Cette approche est très souvent confondue avec la commande adaptative. Elles ont comme objectif commun la convergence vers zéro de l'erreur e , différence entre la sortie désirée y_d et la sortie réelle y du processus. Cependant, si pour la commande adaptative cette erreur diminue asymptotiquement avec le temps t , pour ce qui est de l'auto-réglage, cette erreur diminue uniformément sur un intervalle de temps $[0, T]$ à chaque itération k .

L'objectif est donc ici

$$\lim_{k \rightarrow \infty} \| e_k(t) \| = 0 \text{ ou } \lim_{k \rightarrow \infty} \| e_k(t) \| < \gamma \quad t \in [0, T].$$

où k est le numéro d'itération, $\gamma > 0$ une erreur de tolérance pré-définie et $[0, T]$ l'intervalle de temps qui nous intéresse.

Dans [LN93a, LN93b], D.A. Linkens et J. Nie proposent une méthode d'apprentissage pour la construction de la base de règles de contrôleurs flous. Les processus à commander sont des systèmes multivariables à deux entrées et deux sorties, pouvant inclure des retards purs dans la commande. L'objectif est de construire deux bases de règles indépendantes et découplées, chacune correspondant à une boucle de commande. La structure du système de commande comporte quatre modules : le modèle de référence, l'algorithme d'apprentissage, le mécanisme de formation de la base de règles et le processus à commander. Cette approche tente d'émuler les capacités humaines d'apprentissage : le comportement du système n'est pas explicitement pris en compte. La détermination de la loi d'ajustement est davantage de nature heuristique et par conséquent nécessite peu de connaissance quantitative sur le processus à commander.

Méthodes systématiques

P. Viljamaa et H.N. Koivo [VK93] proposent une méthode plus systématique pour régler un contrôleur flou multivariable afin de commander un système linéaire stable à deux entrées et deux sorties. La structure de commande est définie a priori et le réglage des paramètres est basé sur les gains statiques et une connaissance approchée des constantes de temps et des retards dans le processus. Les gains statiques

peuvent être calculés à partir de réponses indicielles en boucle ouverte du système. Le contrôleur résultant a une partie proportionnelle et une partie intégrale.

Dans un second article, P. Viljamaa et H.N. Koivo [VK94] proposent une autre méthode systématique pour le réglage d'un contrôleur flou multivariable de type P.I., et ce afin de commander un système stable à deux entrées et deux sorties pouvant inclure des retards purs. Le contrôleur est composé de deux parties séparées, une partie proportionnelle et une partie intégrale. Chaque partie a sa propre base de règles. Le réglage des paramètres de la partie proportionnelle est basé sur les dérivées initiales des réponses indicielles et ceux de la partie intégrale sont réglés à partir des gains statiques du système. Dérivées initiales et gains statiques sont mesurés expérimentalement à partir de réponses indicielles en boucle ouverte.

Ces méthodes systématiques n'utilisent aucun modèle du processus mais, malheureusement, elles ne peuvent pas être étendues à des systèmes plus complexes.

1.1.2 Loi de commande basée sur un modèle mathématique du processus

Il s'agit de combiner méthodes classiques et systèmes de commande floue. Cette combinaison aboutit à des lois de commande floue hybrides.

A. Shakouri [SBLD82] utilise un modèle à espace d'état du système pour la synthèse d'un contrôleur flou. Le modèle en question n'est pas nécessairement précis et sert à prédire grossièrement le comportement du système. La commande dépend fortement de la qualité du modèle, ce qui peut conduire à des résultats insatisfaisants. Un algorithme de supervision floue considère donc l'imprécision du modèle et affine la commande.

Avec les méthodes classiques RNA (Rosenbrock-Nyquist Array) la commande de systèmes multivariables est réduite à la conception d'un ensemble de boucles de commande monovariante. Ces approches consistent à déterminer une matrice de pré-compensation multivariable $K_p(s)$ telle que le système résultant $Q(s) = G(s) K_p(s)$ soit diagonalement dominant. Une matrice de compensation diagonale $K_d(s)$ peut alors être calculée, chaque élément diagonal représentant une boucle de commande monovariante. Cependant, cette méthode conduit à des contrôleurs sensibles aux incertitudes de modélisation. Les contrôleurs flous étant supposés robustes de par leur structure à base de connaissance, C-L. Chen et P-C. Chen [CC91] proposent donc de remplacer la matrice de compensation diagonale $K_d(s)$ par autant de contrôleurs P.I.D. flous de type Sugeno qu'il y a de boucles de commande monovariante.

Toutes ces approches sont applicables si l'on dispose d'un modèle précis du système. L'idée de base est issue de l'automatique classique et la commande floue est utilisée afin de compenser des problèmes de robustesse inhérents à l'imprécision du modèle.

1.1.3 Loi de commande déduite d'un modèle flou du processus

De même qu'en automatique classique, le contrôleur flou peut être obtenu à partir d'un modèle flou du processus. Ce modèle est généralement déduit d'un ensemble de données entrées-sorties collectées sur le processus réel à commander. Sans vouloir rentrer dans les détails de l'obtention d'un modèle flou du processus, signalons simplement que celle-ci comporte généralement deux étapes : la recherche d'une partition de l'espace d'entrée (approche heuristique dans [SK86] ou technique de clustering dans [BK95]) puis l'identification de la partie conclusion des règles (méthode de descente du gradient, algorithme des moindres carrés, ...).

Pseudo-inversion du modèle flou

Ce type d'approche est basé sur la pseudo-inversion d'un modèle flou du processus.

Dans [SK86] M.Sugeno et G.T. Kang utilisent des règles floues de type Sugeno [TS85] pour modéliser le comportement global du système, et ce, avec un nombre minimum de règles, la partie conclusion étant une combinaison linéaire des variables d'entrée. Les règles du modèle flou sont appelées lois floues (lois décrivant le processus). Si le processus possède m sorties le modèle comportera m ensembles de lois floues (un ensemble de lois pour chaque sortie). La pseudo-inversion du modèle flou est réalisée en deux étapes : génération de la partie conclusion puis de la partie prémisse. La première phase consiste, sans tenir compte des prémisses des différentes lois, et en ne considérant que la partie conclusion, en une décomposition en sous-systèmes de m équations linéaires (une équation linéaire par sortie). La résolution d'un de ces sous-systèmes fournit la partie conséquence d'une règle de commande floue. La partie prémisse est le résultat de la conjonction des parties prémisses de l'ensemble des lois floues qui ont formé ce sous-système. L'inconvénient majeur de cette méthode est le manque de lisibilité des règles. La nature fonctionnelle de la partie conclusion des règles en est la cause principale.

Dans [Ray95], C. Raymond utilise un modèle flou linguistique de type Mamdani [Ray95]. La pseudo-inversion des règles floues est obtenue par permutation de la conséquence et d'une partie de la prémisse (partie relative aux variables de commande). Bien que la lisibilité des règles soit bien meilleure que précédemment, elle n'est pas garantie. Il faut imposer des contraintes sur les ensembles flous de chaque entrée et sortie. Le recouvrement des fonctions d'appartenance doit être limité si l'on veut que chaque label ait un sens physique. De plus, l'identification d'un tel modèle n'est pas triviale car un compromis est nécessaire entre la précision du modèle et le nombre de règles et, si les règles de commande résultantes sont consistantes (puisqu'elles sont déduites du modèle), leur complétude n'est pas assurée.

Découplage "flou"

Dans beaucoup de systèmes multivariables, il existe des interactions plus ou moins importantes entre les entrées et sorties, chaque entrée commandant plus d'une sortie et chaque sortie étant commandée par plus d'une entrée. Pour simplifier la synthèse de contrôleurs pour de tels systèmes, il est intéressant de découpler le processus de sorte que chaque entrée ne contrôle qu'une sortie et chaque sortie n'est contrôlée que par une entrée. Un système multivariable à m entrées et m sorties se ramène alors à m systèmes monovariables.

La plupart des techniques classiques de découplage souffrent d'un manque de robustesse car elles sont très souvent sensibles aux incertitudes du modèle. Une façon de résoudre le problème est d'utiliser un bloc de découplage flou, la robustesse étant une caractéristique inhérente aux systèmes de commande floue.

Dans [XL89], C-W. Xu et Y-Z. Lu étudient le problème de découplage dans un système aux relations floues à travers une approche par compensation cascade. Parce que le modèle du système et le contrôleur sont tous deux exprimés sous forme d'équations aux relations floues, le découplage est imprécis, ce qui permet de gagner en robustesse.

Ce type de découplage étant encore trop précis, C-W. Xu [Xu91] propose une nouvelle approche, l'approche linguistique. Le découplage se fait au niveau linguistique et non plus au niveau des relations floues.

Ces méthodes nécessitent un modèle flou du processus et restent complexes, spécialement la conception du bloc de découplage flou.

Commande prédictive

Le modèle flou du processus est utilisé pour prédire le comportement du système. Pour un état donné, plusieurs combinaisons de commande peuvent être appliquées au modèle du processus pour avoir les sorties prédites. La combinaison qui optimise le critère de performance prédit est appliquée au système. Néanmoins, la conception d'un schéma de commande prédictif coûte cher en temps-calcul et nécessite l'identification d'un modèle.

1.2 Objectif du travail de thèse

Cette étude bibliographique soulève un certain nombre de problèmes ci-dessous énumérés:

1. Mise en œuvre d'une loi de commande floue relativement difficile,
2. Temps de calculs considérables,
3. Espace mémoire nécessaire important.

Ces problèmes sont la conséquence directe de l'explosion combinatoire de la base de règles. En effet, Si l'on souhaite garder la propriété de complétude de la base de règles, le nombre de règles croît exponentiellement avec le nombre de variables.

A partir de ces constatations, il s'avère impératif, si l'on veut commander des systèmes multivariables complexes, de réduire cette base de règles. Nous distinguons dans ce mémoire deux cas de figure [LT96a, LT] :

1. La base de règles existe déjà et il est nécessaire de diminuer sa complexité (chapitre 2).
2. La base de règles reste à écrire, la synthèse d'un contrôleur flou de complexité réduite est donc à réaliser (chapitre 3).

Il faut cependant noter que l'on ne dispose généralement pas d'une base de règles et qu'à ce jour, aucune méthode systématique ne permet d'en faire la synthèse tout en assurant un nombre limité de règles.

Nous nous plaçons donc dans le second cas de figure, tout en sachant qu'une fois la synthèse du contrôleur flou réalisée, une ultime étape de réduction reste possible (premier cas de figure) [BLTV96, LBT96].

Pour résumer, les objectifs de cette thèse sont, d'une part la **synthèse d'un contrôleur flou de complexité réduite**, d'autre part un outil d'**aide à la conception de contrôleurs flous multivariables** à travers des techniques d'apprentissage (chapitre 4).

Chapitre 2

Réduction de la complexité d'une base de règles

L'objectif de ce chapitre est de présenter les principales méthodes de réduction de la complexité d'une base de règles.

Nous supposons ici que la base de règles est disponible et que l'on désire diminuer sa complexité.

Nous pouvons distinguer deux grands groupes d'approches.

Le premier groupe d'approches (paragraphe 2.1) consiste à éliminer ou à sélectionner certaines règles. La sélection des règles les plus significatives revient à appliquer à deux reprises l'algorithme d'orthogonalisation des moindres carrés (paragraphe 2.1.2).

Le second groupe d'approches (paragraphe 2.1) consiste à diminuer le nombre de labels linguistiques. Dans le paragraphe 2.2.1 une base de règles est simplifiée par fusionnement d'ensembles flous similaires. Dans le paragraphe 2.2.2 le système flou initial est approximé par un autre système flou contenant un nombre moins important de fonctions d'appartenance.

2.1 Élimination ou sélection de règles

2.1.1 Élimination de règles

Si n est le nombre de variables systèmes apparaissant dans la partie prémisse des règles et m le nombre de valeurs linguistiques que peuvent prendre chacune de ces variables, alors une base de règles strictement complète peut être obtenue par énumération de toutes les situations possibles en partie condition des règles. La base de règles correspondante contiendra donc m^n règles différentes. Cette explosion combinatoire du nombre de règles requiert non seulement un espace mémoire conséquent mais nécessite également des temps de calcul importants ce qui peut entraîner des problèmes pour une réalisation en temps réel. Il est donc impératif si l'on veut commander des systèmes de grande dimension de diminuer ce nombre de règles.

L'approche la plus directe et la plus simple pour réduire le nombre de règles consiste à éliminer de la base de règles celles qui ne sont jamais ou que très pauvrement activées.

Les règles dont la partie prémisse est physiquement irréalisable sont tout a fait naturellement ôtées de la base de règles puisque celles-ci ne seront jamais activées. De même, les règles ayant un degré d'activation toujours inférieur à un seuil donné sont éliminées de la base de règles, ces règles ayant que très peu de poids sur le calcul des variables de la partie conclusion.

Cette approche ne conduit pas à une réduction importante du nombre de règles mais peut être considérée comme un premier écrémage de celles-ci.

2.1.2 Algorithme d'orthogonalisation des moindres carrés

Hohensolm et Mendel [HM94] appliquent à deux reprises l'algorithme d'orthogonalisation des moindres carrés (algorithme OLS Orthogonal Least-Squares) et ce, afin de réduire une base de règles.

Tout d'abord, rappelons qu'un système flou de Sugeno d'ordre zéro peut facilement se mettre sous forme mathématique.

Formulation mathématique d'un système flou de Sugeno d'ordre zéro.

Les règles considérées sont des règles de Takagi-Sugeno d'ordre zéro :

$$\text{Règle } l : \text{SI } x_1 \text{ est } F_1^l \text{ et } \dots \text{ et } x_n \text{ est } F_n^l \text{ ALORS } y \text{ est } \theta_l. \quad (2.1)$$

$\underline{x} = (x_1, x_2, \dots, x_n)^T \in U$ est l'ensemble des variables linguistiques en entrée, y est la variable de sortie.

Les fonctions d'appartenance F_i^l ($i = 1, 2, \dots, n$, $l = 1, 2, \dots, M$) définies en entrée sont gaussiennes :

$$\mu_{F_i^l}(x_i) = a_i^l \exp \left[-\frac{1}{2} \left(\frac{x_i - \bar{x}_i^l}{\sigma_i^l} \right)^2 \right] \quad (2.2)$$

a_i^l , \bar{x}_i^l et σ_i^l sont des paramètres ajustables et sont respectivement la hauteur, le centre et la largeur de la gaussienne. D'autres types de fonctions d'appartenance peuvent être choisies.

Les fonctions d'appartenance en sortie sont des singletons flous :

$$\theta_l \in R, \quad l = 1, 2, \dots, M.$$

La combinaison ET des prémisses est réalisée par l'opérateur produit :

$$\mu_A(\underline{x}) = \mu_{F_1^l \times \dots \times F_n^l}(\underline{x}) = \mu_{F_1^l}(x_1) \times \dots \times \mu_{F_n^l}(x_n) \quad (2.3)$$

Le système de logique floue résultant s'écrit alors :

$$y = f(\underline{x}) = \frac{\sum_{l=1}^M \theta_l \left[\prod_{i=1}^n a_i^l \exp \left(-\frac{1}{2} \left(\frac{x_i - \bar{x}_i^l}{\sigma_i^l} \right)^2 \right) \right]}{\sum_{l=1}^M \left[\prod_{i=1}^n a_i^l \exp \left(-\frac{1}{2} \left(\frac{x_i - \bar{x}_i^l}{\sigma_i^l} \right)^2 \right) \right]} \quad (2.4)$$

L'idée de base est de considérer le système flou de Sugeno d'ordre zéro comme étant une combinaison linéaire de fonctions non linéaires appelées *fonctions floues de base* et notées FFB.

Systèmes flous (2.4) définis comme une combinaison de fonctions floues de base

Définition 2.1 Une fonction floue de base (FFB) est définie par :

$$p_j(\underline{x}) = \frac{\prod_{i=1}^n \mu_{F_i^j}(x_i)}{\sum_{j=1}^M \left(\prod_{i=1}^n \mu_{F_i^j}(x_i) \right)}, \quad j = 1, 2, \dots, M \quad (2.5)$$

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'appartenance gaussiennes.

Le système flou (2.4) est alors équivalent à une combinaison linéaire de FFB :

$$f(\underline{x}) = \sum_{j=1}^M p_j(\underline{x})\theta_j \quad (2.6)$$

où $\theta_j \in \mathbb{R}$ sont des constantes.

Notons qu'une fonction floue de base correspond à une règle floue SI...ALORS (2.1) et est calculée comme suit :

1. Calcul de la pseudo fonction floue de base (p-FFB) pour chacune des M règles :

$$(\text{p-FFB})_j = \prod_{i=1}^n \left(\mu_{F_i'}(x_i) \right) \quad , \quad j = 1, \dots, M.$$

La pseudo fonction floue de base pour la règle j ($\text{p-FFB})_j$ correspond au degré d'activation de la $j^{\text{ème}}$ règle.

2. Calcul de la fonction floue de base pour chaque règle :

$$(\text{FFB})_j = p_j(\underline{x}) = \frac{(\text{p-FFB})_j}{\sum_{i=1}^M (\text{p-FFB})_i} \quad , \quad j = 1, \dots, M.$$

La fonction floue de base de la règle j ($\text{FFB})_j$ correspond au degré relatif d'activation de la $j^{\text{ème}}$ règle.

Si nous fixons tous les paramètres de $p_j(\underline{x})$ (les fonctions d'appartenance en entrée sont fixes), alors les seuls paramètres libres sont les θ_j et dans ce cas, $f(\underline{x})$ de (2.6) est linéaire par rapport à ses paramètres. On peut alors utiliser des méthodes d'estimation linéaire de paramètres très efficaces, comme par exemple, l'algorithme d'orthogonalisation des moindres carrés de Gram-Schmidt.

Algorithme d'orthogonalisation des moindres carrés [Wan94]

De façon à saisir le comportement de l'algorithme d'orthogonalisation des moindres carrés, considérons la combinaison linéaire de FFB (2.6) comme un cas particulier du modèle de régression linéaire suivant :

$$d(t) = \sum_{j=1}^M p_j(t)\theta_j + e(t) \quad (2.7)$$

où $d(t)$ est la sortie du système, θ_j sont des paramètres réels, $p_j(t) = p_j(\underline{x}(t))$ sont des fonctions fixes appelées régresseurs et $e(t)$ est un signal d'erreur.

Soient N paires E-S (Entrées-Sortie) données :

$$[\underline{x}^0(t), d^0(t)], \quad t = 1, \dots, N \quad (2.8)$$

L'objectif est de trouver une combinaison linéaire de fonctions floues de base $f(\underline{x})$ telle que $[f(\underline{x}^0(t)) - d^0(t)]$ soit minimisé.

Le modèle de régression linéaire (2.7) peut s'écrire sous forme matricielle :

$$\underline{d} = P\underline{\theta} + \underline{\varepsilon} \quad (2.9)$$

c'est à dire :

$$\begin{bmatrix} d(1) \\ d(2) \\ \vdots \\ d(N) \end{bmatrix} = \begin{bmatrix} p_1(1) & p_2(1) & \cdots & p_M(1) \\ p_1(2) & p_2(2) & \cdots & p_M(2) \\ \vdots & \vdots & & \vdots \\ p_1(N) & p_2(N) & \cdots & p_M(N) \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_M \end{bmatrix} + \begin{bmatrix} \varepsilon(1) \\ \varepsilon(2) \\ \vdots \\ \varepsilon(N) \end{bmatrix}$$

L'objectif de l'algorithme d'orthogonalisation des moindres carrés de Gram-Schmidt est la décomposition orthogonale de P , c'est à dire, $P = WA$, où W est une matrice orthogonale ($W^T W = I$), et A une matrice triangulaire supérieure d'éléments diagonaux égaux à 1. En substituant $P = WA$ dans l'équation (2.9) nous obtenons $\underline{d} = WA\underline{\theta} + \underline{\varepsilon} = W\underline{g} + \underline{\varepsilon}$, où $\underline{g} = A\underline{\theta}$. La solution $\underline{\theta}$ peut facilement être obtenue par substitution ou directement par $\underline{\theta} = A^{-1}.\underline{g}$, la matrice A étant inversible. La seule différence entre l'algorithme original de Gram-Schmidt et celui présenté ci-dessous est qu'ici l'algorithme d'orthogonalisation ne décompose pas toute la matrice P , mais ne considère et ne sélectionne que les colonnes dominantes de P .

Une étape préliminaire à l'application de l'algorithme consiste à initialiser les paramètres a_i^j , \bar{x}_i^j et σ_i^j des fonctions floues de base.

Initialisation des fonctions floues de base : les N $p_j(\underline{x})$ (2.5) sont initialisés à partir des N paires E-S (2.8). Les a_i^j , \bar{x}_i^j et σ_i^j ($i = 1, \dots, n$, $j = 1, 2, \dots, N$) correspondent aux paramètres des fonctions d'appartenance gaussiennes. Les hauteurs sont choisies égales à 1 : $a_i^j = 1$. Les centres des gaussiennes sont équivalentes aux valeurs des entrées des N paires E-S données : $\bar{x}_i^j = x_i^0(j)$. Les largeurs des gaussiennes (σ_i^j) sont choisies de sorte que les N FFB recouvrent uniformément tout l'espace d'entrée défini par les N paires E-S (2.8).

L'algorithme d'orthogonalisation des moindres carrés sélectionne alors, parmi les N fonctions floues de base précédemment initialisées, les M_S plus significatives :

- **Etape 1** : Choix du premier vecteur de base w_1 .

Pour $1 \leq i \leq N$, calculer

$$\underline{w}_j^{(i)} = \underline{p}_i \quad , \quad g_1^{(i)} = \frac{\left(\underline{w}_1^{(i)}\right)^T \underline{d}^0}{\left(\underline{w}_1^{(i)}\right)^T \underline{w}_1^{(i)}} \quad (2.10)$$

$$[\text{err}]_1^{(i)} = \frac{\left(g_1^{(i)}\right)^2 \left(\underline{w}_1^{(i)}\right)^T \underline{w}_1^{(i)}}{\left(\underline{d}^0\right)^T \underline{d}^0} \quad (2.11)$$

avec $\underline{p}_i = \begin{pmatrix} p_i(\underline{x}^0(1)) \\ \vdots \\ p_i(\underline{x}^0(N)) \end{pmatrix}$ donné par la méthode de détermination des FFB initiaux.

Trouver (i_1) tel que

$$[\text{err}]_1^{(i_1)} = \max_{1 \leq i \leq N} \left([\text{err}]_1^{(i)}\right) \quad (2.12)$$

et sélectionner le premier vecteur de base \underline{w}_1 et le premier élément g_1 du vecteur solution des moindres carrés.

$$\underline{w}_1 = \underline{w}_1^{(i_1)} = \underline{p}_{i_1} \quad , \quad g_1 = g_1^{(i_1)} \quad (2.13)$$

- **Etape k**, $2 \leq k \leq M_S$: Choix du vecteur de base suivant \underline{w}_k .

Pour $1 \leq i \leq N$, $i \neq i_1, \dots, i \neq i_{k-1}$, calculer

$$\alpha_{jk}^{(i)} = \frac{\underline{w}_j^T \underline{p}_i}{\underline{w}_j^T \underline{w}_j} \quad , \quad 1 \leq j < k \quad (2.14)$$

$$\underline{w}_k^{(i)} = \underline{p}_i - \sum_{j=1}^{k-1} \alpha_{jk}^{(i)} \underline{w}_j \quad , \quad g_k^{(i)} = \frac{\left(\underline{w}_k^{(i)}\right)^T \underline{d}^0}{\left(\underline{w}_k^{(i)}\right)^T \underline{w}_k^{(i)}} \quad (2.15)$$

$$[\text{err}]_k^{(i)} = \frac{\left(g_k^{(i)}\right)^2 \left(\underline{w}_k^{(i)}\right)^T \underline{w}_k^{(i)}}{\left(\underline{d}^0\right)^T \underline{d}^0} \quad (2.16)$$

Trouver (i_k) tel que

$$[\text{err}]_k^{(i_k)} = \max_{1 \leq i \leq N, i \neq i_1, \dots, i \neq i_{k-1}} \left([\text{err}]_k^{(i)}\right) \quad (2.17)$$

et sélectionner le $k^{\text{ième}}$ vecteur de base \underline{w}_k et le $k^{\text{ième}}$ élément g_k du vecteur solution des moindres carrés.

$$\underline{w}_k = \underline{w}_k^{(i_k)} \quad , \quad g_k = g_k^{(i_k)} \quad (2.18)$$

- Dernière étape: Résoudre le système triangulaire

$$A^{(M_S)} \underline{\theta}^{(M_S)} = \underline{g}^{(M_S)} \quad (2.19)$$

où

$$A^{(M_S)} = \begin{bmatrix} 1 & \alpha_{1,2}^{i_2} & \alpha_{1,3}^{i_3} & \cdots & \alpha_{1,M_S}^{i_{M_S}} \\ & \ddots & \alpha_{2,3}^{i_3} & \cdots & \alpha_{2,M_S}^{i_{M_S}} \\ & 0 & \ddots & & \cdots \\ & & & \ddots & \alpha_{M_S-1,M_S}^{i_{M_S}} \\ & & & & 1 \end{bmatrix} \quad (2.20)$$

$$\underline{g}^{(M_S)} = \begin{bmatrix} g_1^{(M_S)} \\ \vdots \\ g_{M_S}^{(M_S)} \end{bmatrix} \quad , \quad \underline{\theta}^{(M_S)} = \begin{bmatrix} \theta_1^{(M_S)} \\ \vdots \\ \theta_{M_S}^{(M_S)} \end{bmatrix} \quad (2.21)$$

La combinaison finale de FFB est

$$f(\underline{x}) = \sum_{j=1}^{M_S} p_{i_j}(\underline{x}) \theta_j^{(M_S)} \quad (2.22)$$

où les $p_{i_j}(\underline{x})$ sont les FFB sélectionnées, les i_j étant déterminés lors des étapes précédentes.

Remarque 2.1 Les $[err]_k^{(i)} = \frac{(g_k^{(i)})^2 (\underline{w}_k^{(i)})^T \underline{w}_k^{(i)}}{(\underline{x}^{(i)})^T \underline{x}^{(i)}}$ représentent les ratios de réduction d'erreur dus à $\underline{w}_k^{(i)}$. Ainsi, ne seront sélectionnées, pour la combinaison finale de FFB, que les FFB ayant les ratios de réduction d'erreur les plus importants.

Nous allons maintenant exposer la méthode d'orthogonalisation des moindres carrés qu'appliquent à deux reprises Hohensohn et Mendel pour réduire une base de règles.

Réduction de la base de règles par application à deux reprises de l'algorithme d'orthogonalisation des moindres carrés [HM94].

Les fonctions floues de base sont déduites des règles du système flou (2.1). Un système contenant M règles contiendra M FFB. L'algorithme d'orthogonalisation des moindres carrés permet de sélectionner $M_S < M$ FFB et de calculer le vecteur de paramètres correspondant

$$\underline{\theta}^{(M_S)} = (\theta_1^{(M_S)}, \dots, \theta_{M_S}^{(M_S)})^T$$

tel que

$$f(\underline{x}) = \sum_{j=1}^{M_S} p_{i_j}(\underline{x}) \theta_j^{(M_S)}$$

approxime un vecteur d'apprentissage \underline{d} donné.

Cependant, le système flou (2.4) n'a pas été totalement réduit puisque le dénominateur de toutes les FFB sélectionnées considère toutes les fonctions d'appartenance, y compris celles apparaissant dans les $M - M_S$ règles non sélectionnées. La solution proposée par Hohensohn et Mendel [HM94] pour réduire la complexité du système flou initial est l'application en deux étapes de l'algorithme d'orthogonalisation des moindres carrés :

- [1.] $N = N_1$ et déterminer $M = N_1$ FFB initiales.

$$a_i^j = 1 \quad (2.23)$$

$$\bar{x}_i^j = x_i^0(j) \quad (2.24)$$

$$\sigma_i^j = \frac{\left[\max_{j=1,2,\dots,N} (x_i^0(j)) - \min_{j=1,2,\dots,N} (x_i^0(j)) \right]}{M_S} * n \quad (2.25)$$

pour $i = 1, \dots, n, j = 1, 2, \dots, N$ et M_S le nombre de FFB de la combinaison finale.

Garder une trace des paramètres en partie prémisse $(a_i^j, \bar{x}_i^j, \sigma_i^j)$.

- [2.] Appliquer l'algorithme d'orthogonalisation des moindres carrés pour sélectionner parmi $M = N_1$ les M_S règles les plus significatives.

Seules les M_S règles sélectionnées ont un intérêt ici, et seulement leurs paramètres en partie prémisses des règles sont conservés. Le système triangulaire (2.19) n'est pas résolu lors de la première application de l'algorithme.

Remarque 2.2 *Il n'est pas nécessaire de prédéterminer M_S . La première application de l'algorithme peut se terminer au $M_S^{\text{ième}}$ pas lorsque*

$$1 - \sum_{j=1}^{M_S} [\text{err}]_j < \rho \quad (2.26)$$

où ρ est un seuil prédéterminé et $0 < \rho < 1$.

- [3.] Ne conserver que les paramètres des M_S FFB sélectionnées et effacer le reste.

On pose $M = M_S$, afin de réduire le nombre de règles. Les FFB sont maintenant normalisées par

$$\sum_{j=1}^{M_S} \left(\prod_{i=1}^n \mu_{F_i}^j(x_i) \right)$$

Seuls les vecteurs p_i des M_S règles sont nécessaires. Notons que le nombre de vecteurs p_i est réduit, mais pas leur dimension qui est toujours de $N_1 \times 1$.

- [4.] Appliquer l'algorithme d'orthogonalisation des moindres carrés une seconde fois pour obtenir les paramètres $\underline{\theta}^{(M_S)}$.

L'algorithme d'orthogonalisation des moindres carrés utilise maintenant les M_S FFB sélectionnées. L'objectif ici n'est plus de chercher les FFB les plus significatives mais seulement d'orthogonaliser l'ensemble des M_S FFB à l'aide des équations (2.10), (2.14) et (2.15), puis de déterminer le vecteur de paramètres $\underline{\theta}^{(M_S)}$ à partir des équations (2.19) à (2.21).

Remarque 2.3 *La deuxième application de l'algorithme est nettement plus rapide car $M_S \ll M$. Il n'est pas nécessaire lors de la seconde application de l'algorithme d'utiliser les mêmes données d'apprentissage que lors de la première application. On peut choisir $N = N_2 \neq N_1$. L'ensemble de données entrées-sortie peut être beaucoup plus important ($N_2 \gg N_1$), ce qui peut considérablement améliorer les résultats.*

Nous disposons maintenant d'un système flou de Sugeno d'ordre zéro réduit, ne contenant plus que M_S règles. Le temps de calcul étant proportionnel au nombre d'ensembles flous c'est à dire $n.M_S$, cela représente une réduction du temps calcul de $\frac{M_S}{N}$. Cette méthode reste cependant mathématiquement très lourde.

2.2 Diminution du nombre de fonctions d'appartenance

2.2.1 Simplification de la base de règles par fusionnement d'ensembles flous similaires

Setnes [Set95] propose d'utiliser des mesures de similarité pour simplifier une base de règles :

- les fonctions d'appartenance similaires à un ensemble "universel" U défini par

$$\mu_U(x) = 1 \quad \forall x \in X \quad (2.27)$$

sont éliminées de la base de connaissance,

- les ensembles flous similaires sont fusionnés (voir Figure 2.1).

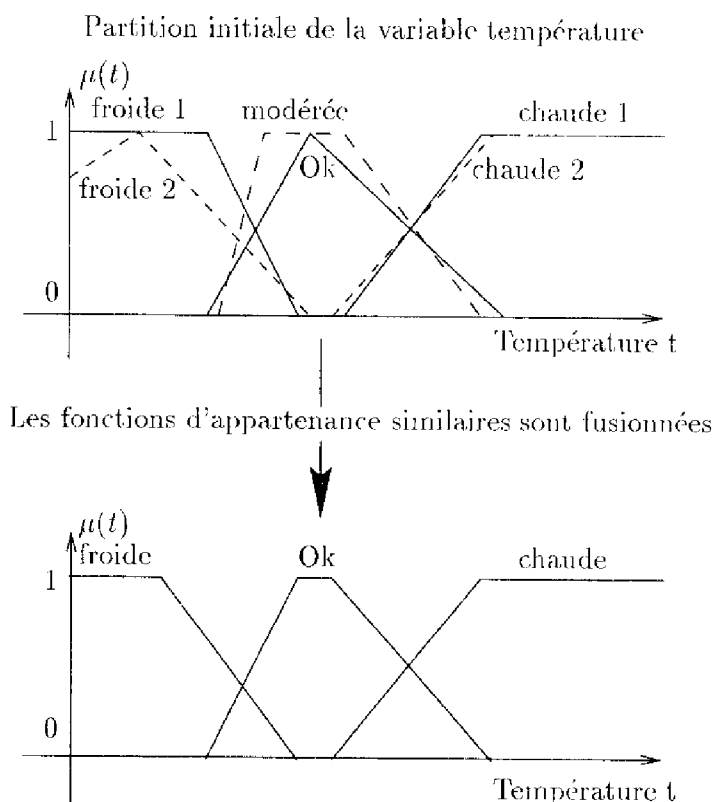


FIG. 2.1 – La fusion de fonctions d'appartenance similaires facilite l'association de labels linguistiques aux fonctions d'appartenance résultantes, augmentant ainsi l'interprétation sémantique des règles.

Cette simplification de la base de règles peut entraîner une réduction du nombre de règles: les règles de prémisses égales sont fusionnées suite à une combinaison de leur partie conclusion (voir Figure 2.2).

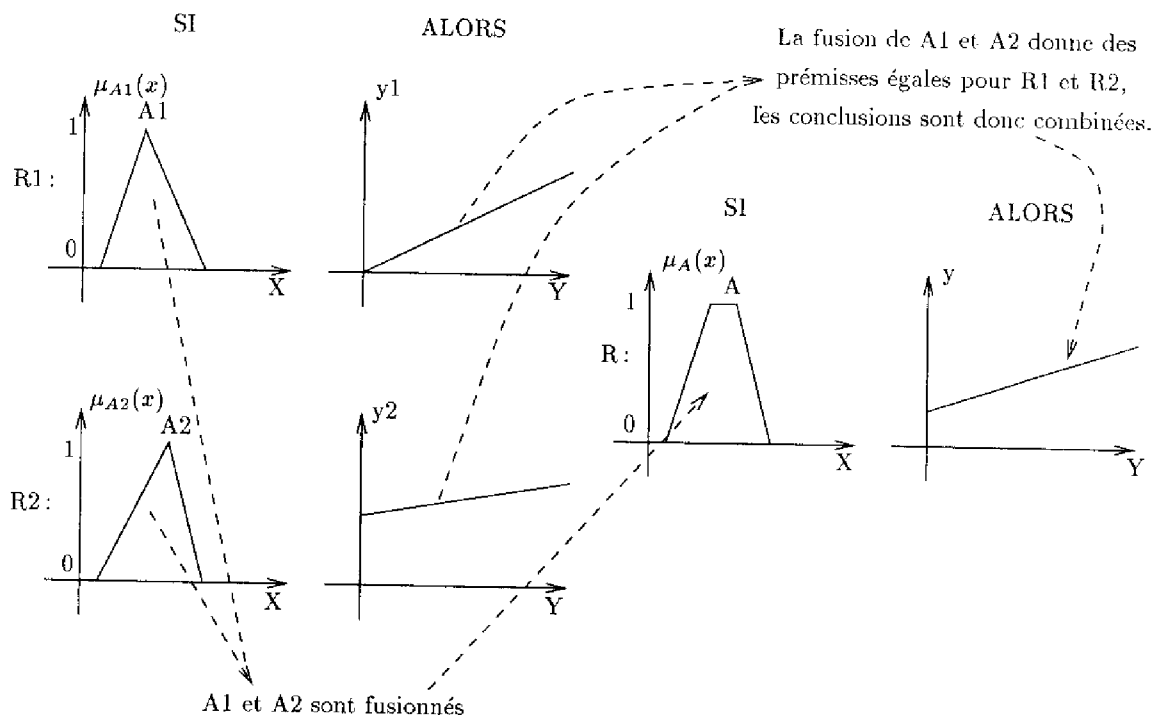


FIG. 2.2 - La fusion d'ensembles flous similaires peut conduire à des prémisses égales, permettant ainsi de réduire le nombre de règles.

La définition des mesures de similarité étant basée sur la notion de cardinalité d'un ensemble flou nous rappelons ici quelques définitions.

La cardinalité d'un ensemble flou \$A\$ sur un univers de discours fini \$X\$ est définie par :

$$|A| = \sum_{x \in X} \mu_A(x), \tag{2.28}$$

Lorsque le nombre d'éléments dans \$X\$ n'est pas fini, \$|A|\$ peut ne pas exister. Cependant, si \$A\$ a un support fini

$$|A| = \sum_{x \in \text{supp}(A)} \mu_A(x) \tag{2.29}$$

La *cardinalité relative* est alors définie par :

$$\|A\| = \frac{|A|}{|X|}. \tag{2.30}$$

L'algorithme complet de la méthode de Setnes est illustré dans la figure 2.3.

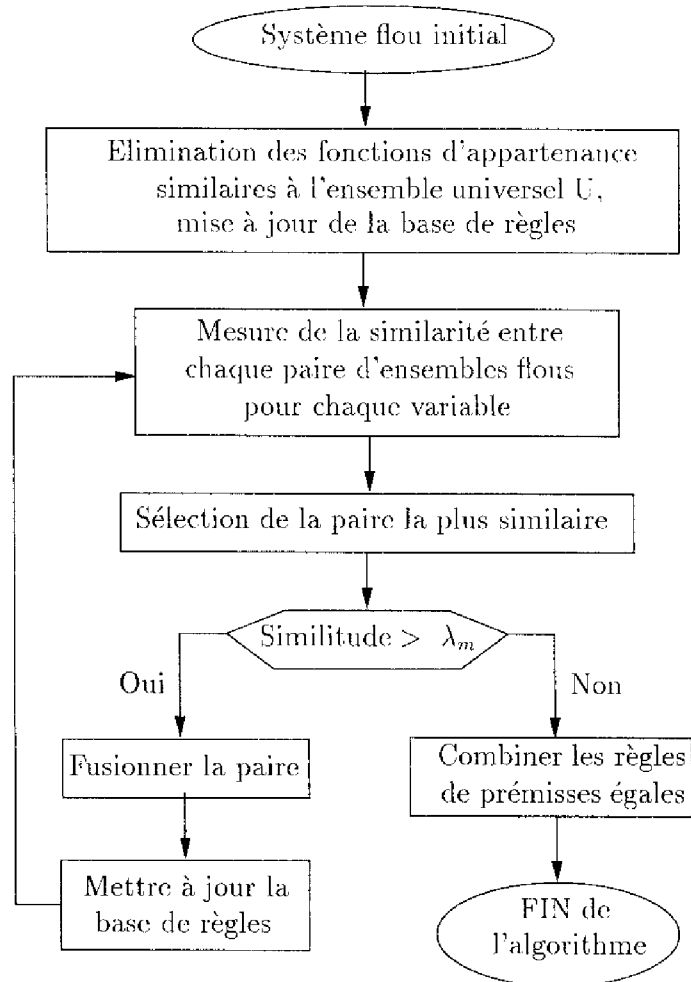


FIG. 2.3 – *Algorithme de simplification-réduction de la base de règles*

Les différentes étapes de l'algorithme sont maintenant détaillées.

- [1.] Définir un seuil de similarité λ_m pour la fusion de deux fonctions d'appartenance : si $S_1(A, B) \geq \lambda_m$ alors A et B sont candidats pour la fusion. $\lambda_m = 0.8$ par défaut.

La première mesure de similarité utilisée par Setnes est basée sur l'intersection, l'union et la cardinalité :

$$S_1(A, B) = \frac{\| \min(A, B) \|}{\| \max(A, B) \|} \quad (2.31)$$

où A et B sont deux ensembles flous et les opérateurs min et max modélisent respectivement l'intersection (\cap) et l'union (\cup). Cette mesure $S_1(A, B) \in [0, 1]$ fournit le degré de recouvrement des deux ensembles flous A et B . Ce degré est le résultat de la division de la surface définie par l'intersection des deux ensembles flous par la surface définie par leur union. $S_1 = 1$ correspond à des ensembles flous égaux et $S_1 = 0$ à des ensembles flous non similaires (c'est à dire de recouvrement nul).

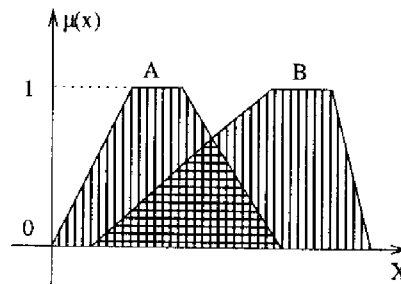


FIG. 2.4 – La mesure de similitude S_1 peut être vue comme le degré de recouvrement des surfaces définies par les fonctions d'appartenance.

- [2.] Définir un seuil de similarité λ_r pour l'élimination des fonctions d'appartenance similaires à l'ensemble universel U : si $S_2(A, U) \geq \lambda_r$, alors A est éliminé. $\lambda_r = 0.9$ par défaut.

La seconde mesure de similarité est basée sur la différence symétrique et la cardinalité :

$$S_2(A, B) = 1 - \|A \square B\| \quad (2.32)$$

où $A \square B$ appelée *différence symétrique* est un ensemble flou dont les éléments appartiennent à A mais pas à B ou appartiennent à B mais pas à A :

$$\mu_{A \square B} = \max(\min(\mu_A(x), 1 - \mu_B(x)), \min(\mu_B(x), 1 - \mu_A(x))), \forall x \in X. \quad (2.33)$$

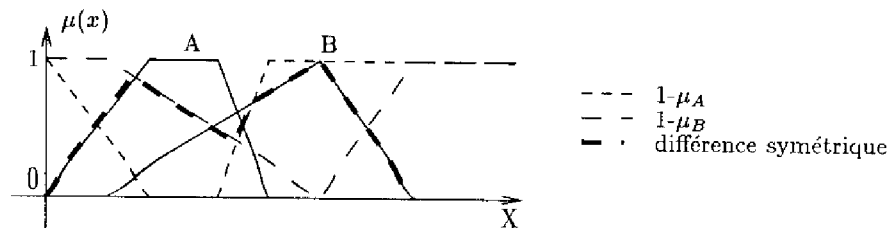


FIG. 2.5 – Différence symétrique entre les ensembles flous A et B

La figure 2.5 illustre l'ensemble flou *différence symétrique* entre A et B (notons au passage que ce n'est pas un ensemble flou convexe).

- [3.] Pour chaque variable du domaine, comparer les fonctions d'appartenance définies pour cette variable avec l'ensemble universel U . Éliminer les ensembles flous A ayant un degré de similarité $S_2(A, U) \geq \lambda_r$. Mettre à jour la base de règles.

Les fonctions d'appartenance similaires à l'ensemble universel U défini par $\mu_U(x) = 1, \forall x \in X$ donneront toujours $\mu(x) \approx 1$ quelle que soit la valeur d'entrée x . De telles fonctions d'appartenance n'ont aucune influence sur le calcul de la sortie. Elles peuvent donc être éliminées de la base de règles sans pour autant altérer le système flou initial (voir Figure 2.6).

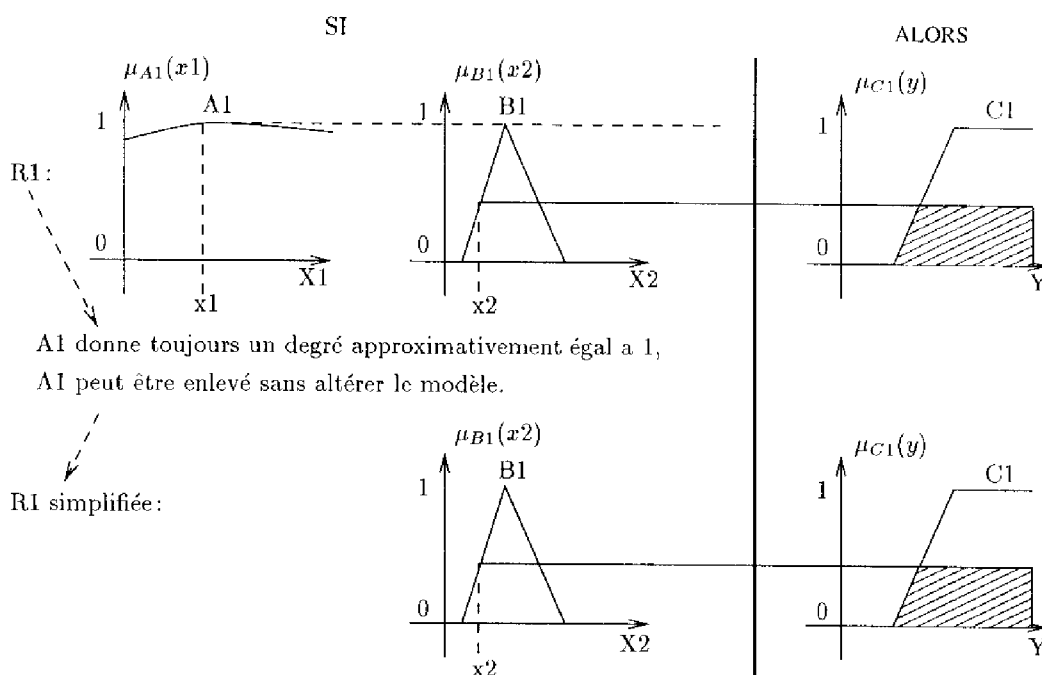


FIG. 2.6 – La fonction d'appartenance $A1$ donne $\mu_{A1}(x) = 1, \forall x \in X1$. Elle ne contribue donc pas au calcul de la sortie et peut être enlevée, donnant une règle plus simple.

- [4.] Mesurer la similitude entre toutes les fonctions d'appartenance, deux à deux, pour chaque variable de la prémisse. Sélectionner la paire de fonctions d'appartenance ayant le plus fort degré de similitude.
- [5.] Si $S_1(A, B) \geq \lambda_m$ alors A et B sont fusionnés sinon aller à 8.

Fusionner une paire de fonctions d'appartenance consiste à remplacer les deux ensembles flous par une nouvelle fonction d'appartenance représentative des deux ensembles (voir figure 2.1).

Les ensembles flous sont généralement paramétrés. A titre d'exemple, la figure 2.7 illustre la paramétrisation d'ensembles flous dans la *Fuzzy Toolbox*

de Matlab : une fonction d'appartenance est décrite par un vecteur de dimension (5×1) , où le premier élément indique le type de fonction d'appartenance (1=trapézoïdale, 2=exponentielle, 3=sigmoïdale) et les quatre derniers éléments indiquent le début du support, le début du noyau, la fin du noyau et la fin du support de l'ensemble flou. La fusion de deux ensembles flous revient donc à déterminer les quatre paramètres du vecteur décrivant la nouvelle fonction d'appartenance.

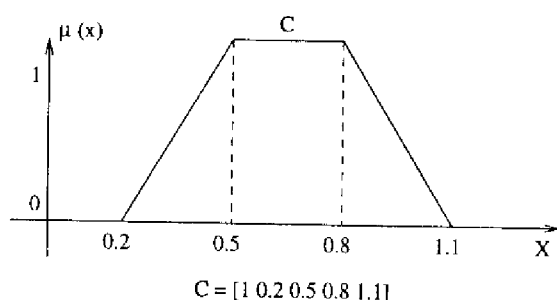


FIG. 2.7 – Représentation d'une fonction d'appartenance dans la Fuzzy Toolbox de Matlab

Setnes choisit de définir sa nouvelle fonction d'appartenance en faisant la moyenne des paramètres des deux ensembles flous A et B à fusionner. Une illustration est donnée dans la figure 2.8.

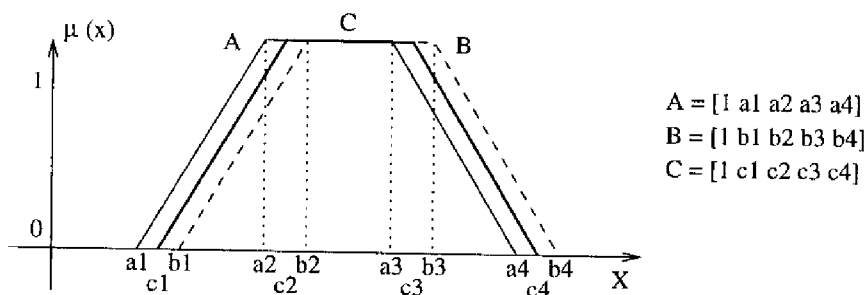


FIG. 2.8 – Fusion de deux fonctions d'appartenance

Remarque 2.4 Lors de la fusion de deux ensembles flous, il faut faire attention à ce que cela ne crée pas de 'trous' dans la base de règles.

Remarque 2.5 Seulement deux fonctions d'appartenance sont fusionnées par itération, mais il est possible qu'une d'elles ou les deux aient été créés par la fusion d'autres ensembles flous lors d'une itération précédente. Pour que la nouvelle fonction d'appartenance prenne en compte, à coefficient égal, tous les

ensembles flous impliqués, son vecteur de paramètres est défini par :

$$C_p = \frac{n_A A_p + n_B B_p}{n_A + n_B}, \quad (2.34)$$

où C_p est le vecteur de paramètres de la nouvelle fonction d'appartenance, et A_p et B_p les vecteurs de paramètres des fonctions d'appartenance fusionnées. n_A et n_B sont égaux aux nombres de fonctions d'appartenance précédemment combinées pour créer A et B respectivement.

- [6.] Mettre à jour la base de règles.

Les systèmes flous considérés sont des systèmes flous de Sugeno d'ordre un. La conclusion $y^i = f^i(x)$ dans la $i^{\text{ème}}$ règle est une combinaison linéaire des variables intervenant en partie prémisses :

$$f^i(x_1, x_2, \dots, x_n) = p_0^i + p_1^i x_1 + p_2^i x_2 + \dots + p_n^i x_n \quad (2.35)$$

Une fois les deux ensembles flous fusionnés, une nouvelle fonction d'appartenance les remplace. En ce qui concerne la partie conclusion des règles affectées, soit on garde les mêmes paramètres ($p_0^i, p_1^i, p_2^i, \dots, p_n^i$) soit on les ré-estime à l'aide d'un algorithme des moindres carrés en utilisant un jeu de données constitué des mesures entrées-sortie issues du système flou initial.

- [7.] Aller à 4.
- [8.] Combiner les règles de prémisses égales.

Il est possible, après fusion, que deux règles aient leur partie prémisses identique, leur partie conclusion peut alors être combinée. Pour les systèmes flous de Sugeno, où les conclusions sont des modèles locaux linéaires, cela peut être réalisé en faisant la moyenne des deux conséquences pour former une nouvelle conclusion (voir figure 2.2). Cela conduit à une *réduction de la base de règles*. Cependant, deux règles ayant une partie prémisses identique mais une partie conclusion très différente peut être révélateur d'une incohérence de la base de règles (base de règles inconsistante).

- [9.] FIN de l'algorithme.

Cette approche est très intéressante lorsque les fonctions d'appartenance du système flou initial ont un fort degré de recouvrement, la lisibilité de la base de règles étant alors considérablement améliorée. Cependant, cette méthode ne conduit pas forcément à une réduction de la base de règles. En effet, il n'est pas assuré que l'on trouve des règles de prémisses égales.

2.2.2 Approximation du système flou initial

Dreier [Dre94a, Dre94b] propose d'approximer un système flou multi-entrées mono-sortie, linéaire ou non linéaire, par un système flou de Sugeno d'ordre zéro.

Dans un premier temps, un jeu de données entrées-sortie est déduit du système flou à simplifier. Des règles de type Takagi-Sugeno d'ordre zéro sont alors générées afin d'approximer au mieux le précédent jeu de données. Les règles obtenues sont de la forme :

SI (x_1 est A_{j1} et x_2 est $A_{j2} \dots$ et x_n est A_{jn}) ALORS (la sortie y est S_j)

La méthode de Dreier est la suivante.

A chaque variable d'entrée x_i sont associées m_i fonctions d'appartenance. Le nombre de règles est donc de

$$r = \prod_{i=1}^n m_i. \quad (2.36)$$

Les fonctions d'appartenance peuvent être de formes quelconques. Cependant, on choisit ici des ensembles flous triangulaires symétriques et ce, pour plusieurs raisons : mise en œuvre aisée et temps de calcul moins important. Les centres des triangles sont répartis régulièrement sur tout l'univers de discours. La base d'un triangle commence et se termine aux valeurs centrales des ensembles adjacents. Ainsi, les fonctions d'appartenance forment une partition floue, c'est à dire vérifient

$$\sum_{j=1}^r \mu_{A_j}(x) = 1 \quad \forall x. \quad (2.37)$$

Les r règles couvrent toutes les combinaisons possibles des fonctions d'appartenance dans la partie prémisse. Pour le $k^{\text{ème}}$ point (défini par la $k^{\text{ème}}$ donnée), la $j^{\text{ème}}$ règle est

R_j : SI $x_1(k)$ est A_{j1} et $x_2(k)$ est A_{j2} et \dots et $x_n(k)$ est A_{jn} ALORS $y(k)$ est S_j

Le degré d'activation de la $j^{\text{ème}}$ règle est déterminé par la composition min

$$\mu_{R_{k_j}} = \min(\mu_{A_{j1}}(x_1(k)), \mu_{A_{j2}}(x_2(k)), \dots, \mu_{A_{jn}}(x_n(k))) \quad (2.38)$$

ou la combinaison produit

$$\mu_{R_{k_j}} = \mu_{A_{j1}}(x_1(k)) * \mu_{A_{j2}}(x_2(k)) * \dots * \mu_{A_{jn}}(x_n(k)). \quad (2.39)$$

Le nombre de données entrées-sortie est q . Soit X_i le vecteur contenant les valeurs de l'entrée x_i aux q points et Y les q valeurs de sortie correspondantes :

$$X_i = (x_i(1), \dots, x_i(k), \dots, x_i(q))^T \quad (2.40)$$

$$Y = (y(1), \dots, y(k), \dots, y(q))^T. \quad (2.41)$$

Au point k , la sortie y est donnée par

$$y(k) = \frac{\sum_{j=1}^r \mu_{R_{kj}} \cdot S_j}{\sum_{j=1}^r \mu_{R_{kj}}} \quad (2.42)$$

Soit la matrice de coefficients C définie par

$$C_{kj} = \frac{\mu_{R_{kj}}}{\sum_{j=1}^r \mu_{R_{kj}}} \quad (2.43)$$

A partir des équations (2.42) et (2.43) on peut écrire

$$Y = C.S \quad (2.44)$$

où $S = (S_1, \dots, S_j, \dots, S_r)^T$ et C est une matrice de dimension $(q \times r)$.

Pour résoudre (2.44), on peut être tenté de pré-multiplier (2.44) par C^T , puis inverser $C^T * C$. On reconnaît là la forme matricielle de la méthode des moindres carrés, méthode minimisant l'erreur entre la sortie désirée et la sortie réelle. Malheureusement, rien ne garantit l'activation de toutes les règles. Cela signifie que C peut avoir une colonne nulle et la matrice $C^T * C$ peut ainsi être une matrice singulière. La décomposition en valeurs singulières peut ne pas aboutir à une pseudo-inverse de C . C'est pourquoi la méthode de réduction modale est utilisée.

Soit

$$W = C^T * C. \quad (2.45)$$

Si W est non singulière, alors ses valeurs propres, Λ , et ses vecteurs propres, rangés par colonne dans la matrice modale Z , satisfont

$$W.Z = Z.\Lambda \quad (2.46)$$

et vérifient la propriété suivante

$$Z^T.W.Z = Z^T.Z.\Lambda = \text{diag}(\Lambda). \quad (2.47)$$

Les vecteurs propres colonnes de Z correspondent aux éléments diagonaux de la matrice des valeurs propres. Maintenant si W est singulière ou mal conditionnée, alors quelques unes des valeurs propres sont nulles ou proches de zéro. Dreier définit une valeur propre de zéro comme étant une valeur propre dont l'amplitude est

inférieure à 10^{-5} fois la plus grande des valeurs propres. Si les colonnes de Z qui correspondent à ces valeurs propres nulles ou presque nulles sont ôtées, et que la matrice modale réduite est appelée Z_r , alors la matrice des valeurs propres réduite Λ_r est définie par

$$\text{diag}(\Lambda_r) = Z_r^T \cdot W \cdot Z_r \quad (2.48)$$

avec $\text{diag}(\Lambda_r)$ inversible.

La matrice Z_r est reliée aux singletons physiques S et aux singletons modaux V par

$$S = Z_r \cdot V \quad (2.49)$$

Pré-multiplier l'expression (2.44) par C^T puis remplacer $C^T * C$ par W (2.45).

Pré-multiplier par Z_r^T , et utiliser (2.48) et (2.49) pour obtenir

$$Z_r^T \cdot C^T \cdot Y = \text{diag}(\Lambda_r) \cdot V \quad (2.50)$$

L'équation (2.50) peut être résolue par rapport à V puisque $\text{diag}(\Lambda_r)$ est inversible. Une fois V connu, S est déterminé à travers l'équation (2.49). Quelques uns des singletons physiques de S peuvent être nuls ou presque nuls. Les règles qui y sont associées peuvent être éliminées puisqu'elles ne contribuent pas au résultat final Y .

Cette méthode se distingue d'autres types d'approches de modélisation du fait que c'est une méthode non itérative. Elle suppose que le système flou initial est un système multi-entrées mono-sortie mais peut être étendue à des systèmes multivariables. Il suffit alors de décomposer le système flou multivariable en autant de sous-systèmes multi-entrées mono-sortie qu'il y a de sorties dans le système flou initial.

2.3 Conclusion

Dans ce chapitre, nous avons exposé les principales techniques de réduction du nombre de règles d'une base donnée.

Dans le chapitre suivant, le point de départ n'est plus la base de règles mais le procédé à commander. Nous allons donc, maintenant, chercher à faire la synthèse d'un contrôleur flou tout en se préoccupant de la minimisation de son nombre de règles.

Chapitre 3

Vers la synthèse d'un contrôleur flou de complexité réduite

Nous supposons ici, contrairement au chapitre précédent (chapitre 2), que la base de règles n'est pas disponible et que la synthèse d'un contrôleur flou de complexité réduite est à réaliser.

Aussi, dans le cadre de ce chapitre, nous présentons différentes techniques conduisant à un contrôleur flou de complexité réduite.

La fusion mathématique (paragraphe 3.1.1) ou symbolique (paragraphe 3.1.2) des variables d'entrée du contrôleur flou permet de réduire la dimension du problème de commande et avec lui le nombre de règles nécessaires.

Une autre voie intéressante est la structuration de la loi de commande, décentralisée (paragraphe 3.2.1) ou hiérarchisée (paragraphe 3.2.2), laquelle facilite considérablement l'écriture de la base de règles.

Une combinaison des techniques ci-dessus citées est également possible, par exemple, réduction de la dimension et hiérarchisation de la loi de commande (paragraphe 3.3).

3.1 Réduction de la dimension du problème

La réduction de la dimension du problème de commande consiste à diminuer le nombre de variables n en entrée du contrôleur flou. Cette réduction peut être obtenue par fusion mathématique des variables d'entrée ou encore par fusion symbolique, laquelle repose sur l'utilisation de fonctions d'appartenance multidimensionnelles.

3.1.1 Fusion mathématique

Cette méthode consiste à combiner mathématiquement des variables avant de les fournir en entrée du contrôleur flou. Ces variables sont généralement agrégées linéairement.

Dans la figure 3.1 plusieurs cas sont examinés afin d'illustrer la méthode. Les coefficients de fusion a, b, c et d sont des réels positifs dont les valeurs découlent de considérations physiques ou d'une connaissance experte sur le processus à commander. On considère, dans chacun des cas de la figure 3.1, que les variables en entrée du contrôleur flou sont représentées par $m = 5$ valeurs linguistiques : NM pour *Négatif Moyen*, NP pour *Négatif Petit*, ZE pour *Zéro*, PP pour *Positif Petit* et PM pour *Positif Moyen*. Ces notations seront couramment utilisées dans la suite du mémoire.

Considérons le cas (b) où $n = 3$ variables (x_1, x_2 et x_3). Une base de règles complète comporterait donc $k = m^n = 5^3 = 125$ règles de la forme :

$$R_{ijk} : \text{Si } x_1 \text{ est } F1^i \text{ et } x_2 \text{ est } F2^j \text{ et } x_3 \text{ est } F3^k \text{ alors } u \text{ est } C_{ijk} \quad (3.1)$$

pour $i, j, k = 1$ à 5 . Les indices i, j et k correspondent respectivement au $i^{\text{ème}}$, $j^{\text{ème}}$ et $k^{\text{ème}}$ ensemble flou, lesquels sont ordonnés comme suit $\{NM, NP, ZE, PP, PM\}$.

Les variables x_1, x_2 et x_3 peuvent être combinées linéairement de différentes manières :

- fusion mathématique de x_1 et x_2 : $\begin{cases} X1 &= a.x1 + b.x2 \\ X2 &= x3 \end{cases}$
- fusion mathématique de x_1 et x_3 : $\begin{cases} X1 &= a.x1 + b.x3 \\ X2 &= x2 \end{cases}$
- fusion mathématique de x_2 et x_3 : $\begin{cases} X1 &= a.x2 + b.x3 \\ X2 &= x1 \end{cases}$
- fusion mathématique de x_1, x_2 et x_3 : $X = a.x1 + b.x2 + c.x3$

Le nombre de règles sera ainsi réduit de 125 à 25 si deux variables sont fusionnées ou encore de 125 à 5 si les trois variables sont combinées :

$$\left\{ \begin{array}{l} R_1: \text{ Si } X \text{ est NM alors } u \text{ est } C_1, \\ R_2: \text{ Si } X \text{ est NP alors } u \text{ est } C_2, \\ R_3: \text{ Si } X \text{ est ZE alors } u \text{ est } C_3, \\ R_4: \text{ Si } X \text{ est PP alors } u \text{ est } C_4, \\ R_5: \text{ Si } X \text{ est PM alors } u \text{ est } C_5. \end{array} \right.$$

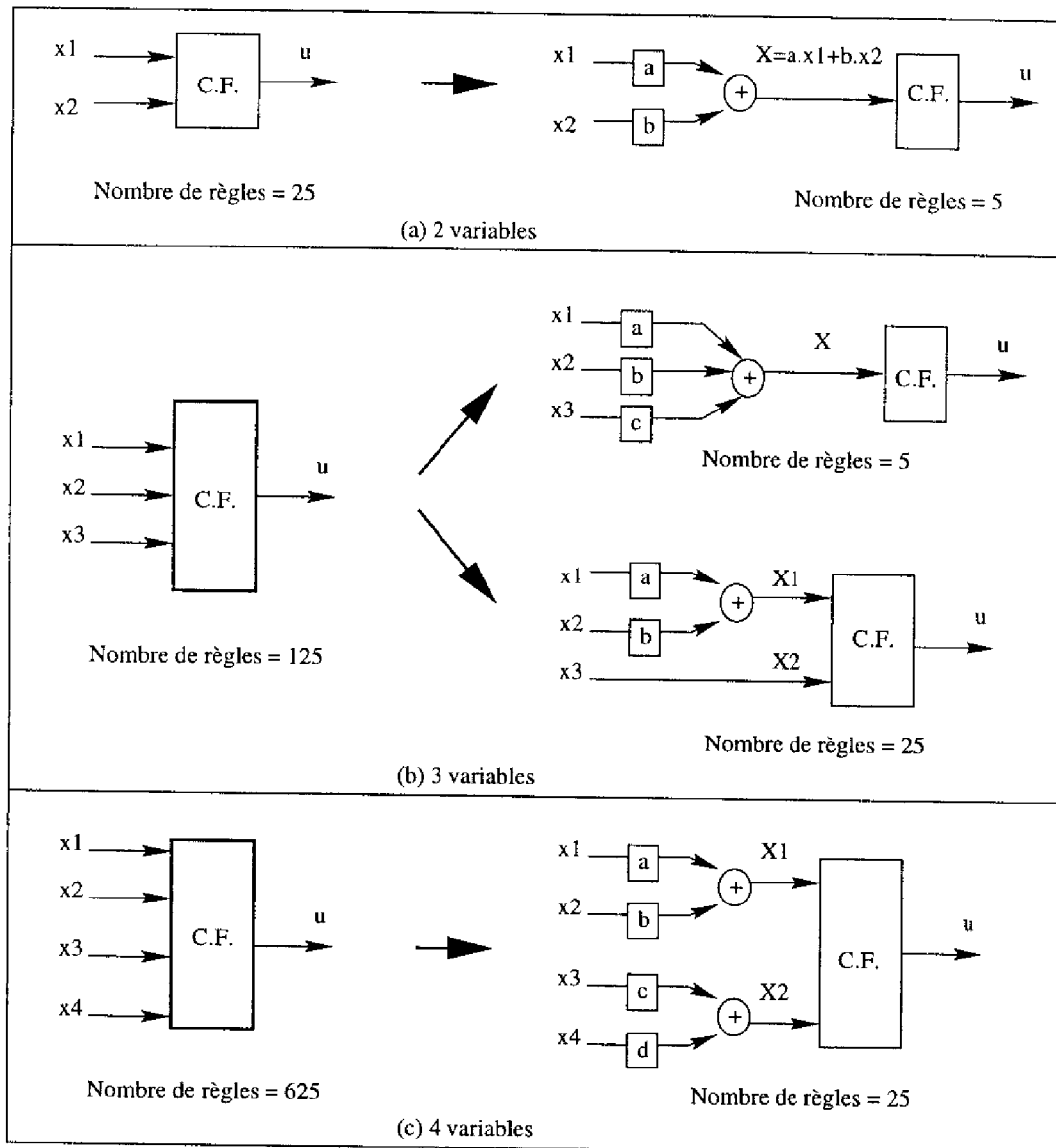


FIG. 3.1 - Fusion mathématique pour $n = 2, 3, 4$ variables et $m = 5$ labels linguistiques par variable.

La réduction du nombre de règles est optimale si l'on peut fusionner toutes les variables d'entrée en une seule variable adjointe. Dans ce cas, le nombre de règles est égal au nombre de labels linguistiques définis pour cette variable. Il est cependant évident que toutes ces variables ne peuvent pas être fusionnées arbitrairement, toute combinaison de variables devant garder un sens physique. Pour des problèmes de régulation de systèmes d'ordre n , et pour chacune des variables régulées, c'est généralement l'erreur (différence entre la variable et sa valeur de consigne) et ses $(n - 1)$ dérivées successives que l'on fusionne en une autre variable adjointe représentant une "situation" pour cette variable.

Nous allons maintenant rappeler quelques concepts de base relatifs à la théorie des systèmes à structure variable (commande en mode glissant) de façon à justifier la fusion mathématique proposée pour des problèmes de régulation : combinaison de l'erreur et de ses dérivées successives.

Rappels sur la commande en mode glissant

Soit le système non linéaire d'ordre n défini par

$$\dot{x}^{(n)} = f(\underline{x}, u) \quad (3.2)$$

où $\underline{x} = \begin{bmatrix} x \\ \dot{x} \\ \vdots \\ x^{(n-1)} \end{bmatrix}$ est le vecteur d'état et u la variable de commande.

L'erreur de poursuite $\underline{e} = \underline{x}_d - \underline{x}$ représente la différence entre le vecteur d'état désiré et le vecteur d'état courant et est définie par le vecteur

$$\underline{e} = \begin{bmatrix} e \\ \dot{e} \\ \vdots \\ e^{(n-1)} \end{bmatrix} \quad (3.3)$$

Le principe de base d'un système à structure variable consiste à :

1. définir l'erreur généralisée S comme une fonction linéaire des coordonnées du vecteur d'erreur de poursuite \underline{e} (3.3)

$$S(\underline{x}) = \sum_{i=1}^n c_i \cdot e^{(i-1)} \quad (3.4)$$

La surface définie par l'équation $S(\underline{x}) = 0$ est appelée *surface de glissement* dans l'espace d'état.

2. générer une loi de commande telle que le système revient sur la surface de glissement et y reste.

La loi de commande est souvent choisie comme suit :

$$u = M \cdot \text{sgn}(S) \quad \text{avec} \quad \text{sgn}(S) = \begin{cases} +1 & \text{si } S > 0 \\ 0 & \text{si } S = 0 \\ -1 & \text{si } S < 0 \end{cases} \quad (3.5)$$

Une telle loi de commande satisfait les conditions de mode glissant suivantes et est représentée dans la figure 3.2 :

1. Si $e = 0$ alors $S = 0$,
2. Si $e \neq 0$ alors $S \cdot \dot{S} < 0$,
3. $S > 0$ si on est d'un côté de la surface de glissement et $S < 0$ si on se trouve du côté opposé.

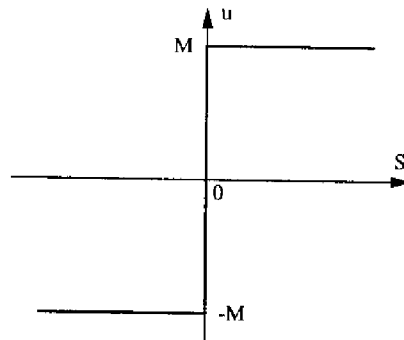


FIG. 3.2 - Loi de commande en mode glissant de type "bang-bang"

Par souci de clarté, nous considérons maintenant un système du 2^{ème} ordre : $\underline{x} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$ La surface de glissement $S = 0$ est une droite définie par :

$$S = c_1 \cdot e + c_2 \cdot \dot{e} \quad (3.6)$$

ce qui s'écrit aussi

$$S = \dot{e} + \lambda e \quad \text{avec} \quad \lambda = \frac{c_1}{c_2} > 0 \quad (3.7)$$

Par conséquent, sur la droite de glissement, le système se comporte comme un système du 1^{er} ordre de constante de temps $\frac{1}{\lambda}$.

La loi de commande (3.5) présente cependant un inconvénient majeur (voir la figure 3.2) : à chaque changement de signe de S , l'action de commande est très forte et change de signe. La commande est dite "bang-bang".

Pour éviter de si brusques changements dans la loi de commande, celle-ci est adoucie par l'introduction d'une zone de commutation contenant la surface de glissement (voir la figure 3.3). Cette zone de commutation permet d'obtenir une transition plus douce entre les deux valeurs de saturation de la précédente loi de commande.

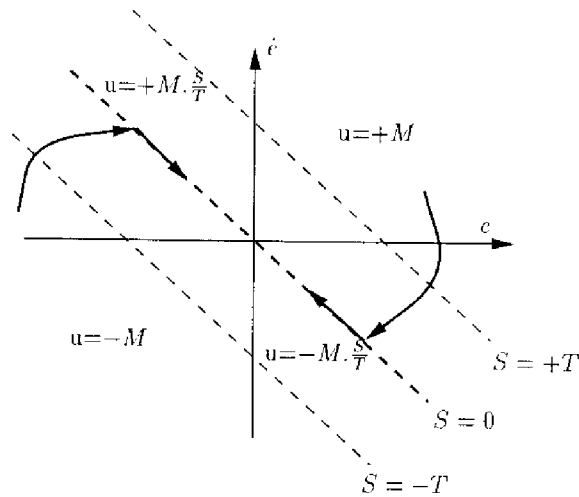


FIG. 3.3 – Introduction d'une zone de commutation, la commande en mode glissant.

A l'intérieur de la zone de commutation, la commande est proportionnelle à S et en dehors de cette zone la commande est saturée. Cette loi de commande est illustrée dans la figure 3.4 et est définie par :

$$u = M \cdot \text{sat}(S) \text{ avec } \text{sat}(S) = \begin{cases} \text{sgn}(\frac{S}{T}) & \text{si } |\frac{S}{T}| \geq 1 \\ \frac{S}{T} & \text{si } |\frac{S}{T}| < 1 \end{cases} \quad (3.8)$$

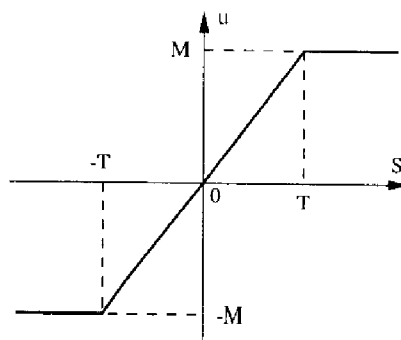


FIG. 3.4 – Version pratique de la loi de commande en mode glissant

Il est alors possible d'obtenir l'équivalent flou d'un système à structure variable réalisant une commande en mode glissant.

Contrôleur flou équivalent à un contrôleur en mode glissant [Pal92]

De la même façon que pour les systèmes à structure variable, pour un système du $n^{\text{ème}}$ ordre, on définit une erreur généralisée $S = \sum_{i=1}^n c_i \cdot e^{(i-1)}$. S est donc le résultat de la fusion d'un signal d'erreur et de ses $(n - 1)$ dérivées successives. Par conséquent, au lieu d'avoir à résoudre un problème de commande du $n^{\text{ème}}$ ordre, nous sommes confrontés à un problème plus simple du premier ordre dont l'objectif est de stabiliser S autour de zéro.

Soit la base de règles : $\left\{ \begin{array}{l} R_1 : \text{“Si } S \text{ est NM alors } u \text{ est NM”} \\ R_2 : \text{“Si } S \text{ est NP alors } u \text{ est NP”} \\ R_3 : \text{“Si } S \text{ est ZE alors } u \text{ est ZE”} \\ R_4 : \text{“Si } S \text{ est PP alors } u \text{ est PP”} \\ R_5 : \text{“Si } S \text{ est PM alors } u \text{ est PM”} \end{array} \right.$

condensée sous forme d'une table de règles :

S	NM	NP	ZE	PP	PM
u	NM	NP	ZE	PP	PM

Les fonctions d'appartenance de S et u , ainsi que la sortie du contrôleur flou sont représentées dans la figure 3.5.

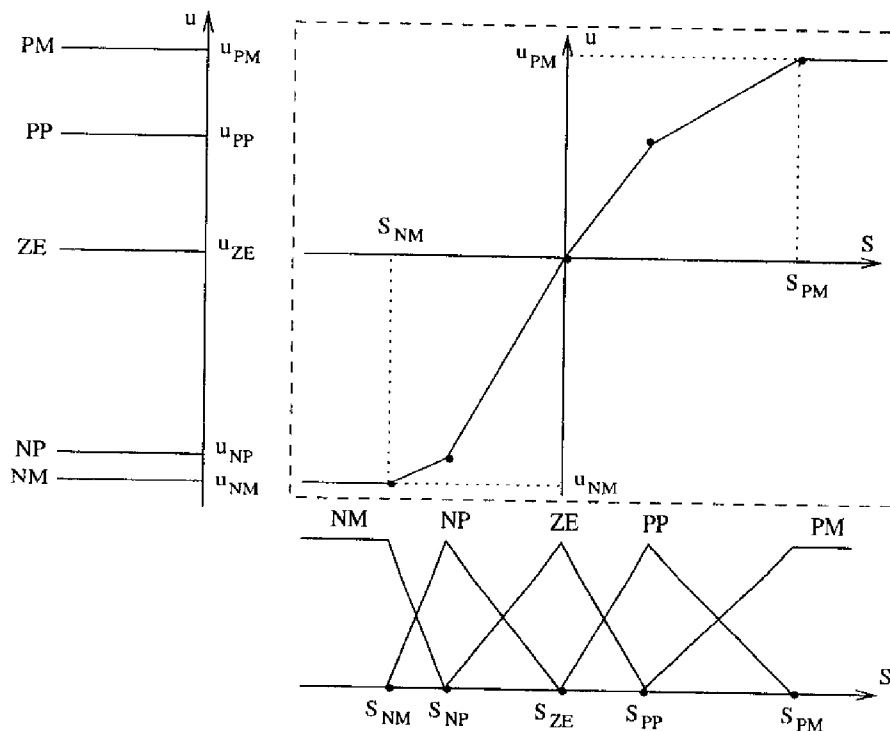


FIG. 3.5 – Contrôleur flou équivalent à un contrôleur en mode glissant : un exemple.

Ainsi, un contrôleur flou peut réaliser une commande en mode glissant. La seule différence avec la version classique de la loi de commande en mode glissant (3.8) est que, à l'intérieur de la zone de commutation, u est en général une fonction non linéaire de S , que l'on peut rendre, évidemment, linéaire, si l'on veut.

Illustration sur le pendule inversé [LT97b]

Une représentation du pendule inversé est donnée dans la figure 3.6. Un chariot surmonté d'une tige se déplace horizontalement sur une courroie actionnée par un moteur à courant continu.

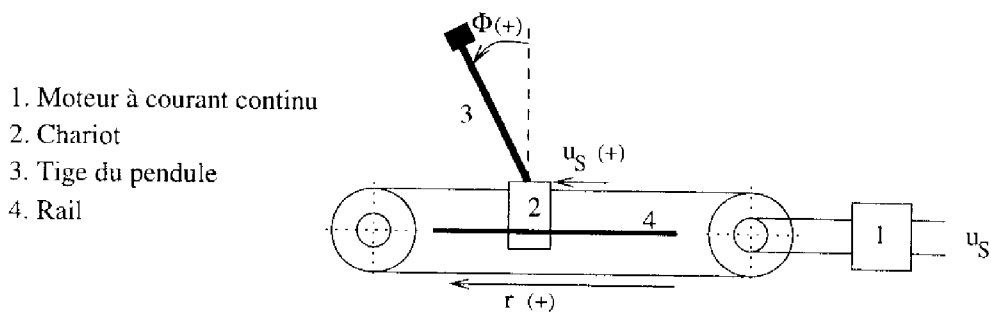


FIG. 3.6 - Description schématique du pendule inversé

Les variables principales sont :

- la position angulaire de la tige Φ ,
- la vitesse angulaire de la tige $\dot{\Phi}$,
- la position horizontale du chariot r ,
- la vitesse de déplacement du chariot \dot{r} .

Soit le vecteur d'état

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} r \\ \Phi \\ \dot{r} \\ \dot{\Phi} \end{bmatrix} \quad (3.9)$$

les équations différentielles $\dot{\underline{x}} = f(\underline{x}, u_S)$ modélisant le pendule inversé sont de la forme :

$$\begin{aligned} \dot{x}_1 &= x_3 \\ \dot{x}_2 &= x_4 \\ \dot{x}_3 &= \frac{a_{32} \cdot \sin(x_2) \cdot \cos(x_2) + a_{33} \cdot x_3 + a_{34} \cdot x_4 \cdot \cos(x_2) + a_{35} \cdot x_4^2 \cdot \sin(x_2) + b_3 \cdot u_S}{1 - a_{45} \cdot (\sin(x_2))^2} \\ \dot{x}_4 &= \frac{a_{42} \cdot \sin(x_2) + a_{43} \cdot x_3 \cdot \cos(x_2) + a_{44} \cdot x_4 + a_{45} \cdot x_4^2 \cdot \cos(x_2) \cdot \sin(x_2) + b_4 \cdot \cos(x_2) \cdot u_S}{1 - a_{45} \cdot (\sin(x_2))^2} \end{aligned} \quad (3.10)$$

Le lecteur intéressé par l'établissement du modèle et par l'expression des coefficients a_{ij} et b_i pour $i = 3, 4$ et $j = 2, \dots, 5$ est invité à consulter [Ami92].

L'objectif de la stratégie de régulation est, d'une part, de maintenir la tige du pendule en position verticale haute, d'autre part, d'amener le chariot vers une position donnée r_c .

Les variables d'entrée du contrôleur flou sont la position et une grandeur proportionnelle à la vitesse angulaire de la tige, Φ et $\Delta\Phi$, l'erreur en position e donnée par $e = r - r_c$ et une grandeur proportionnelle à la vitesse de déplacement du chariot $\Delta e = \Delta r$. La variable de sortie du contrôleur flou est une tension u_S permettant de commander un moteur à courant continu.

La fusion mathématique des variables d'entrée du contrôleur flou est réalisée comme suit :

$$\begin{cases} X_\Phi = a\Phi + b\Delta\Phi & (a > 0, b > 0) \\ X_e = ce + d\Delta e & (c > 0, d > 0) \end{cases} \quad (3.11)$$

Ainsi, si X_e est nul, cela signifie que le chariot a atteint sa position de référence ($e = 0$ et $\Delta e = 0$), ou encore qu'il se déplace vers celle-ci ($ce = -d\Delta e$). Le raisonnement est identique pour X_Φ . Si X_Φ est nul, la position angulaire du pendule est stabilisée ou s'approche de zéro. Par conséquent, la stabilisation de X_Φ et X_e permet d'amener le pendule vers une position de référence et assure le maintien de la tige du pendule en position verticale haute. Plus la valeur absolue de X_Φ , respectivement, X_e , est importante, plus la position horizontale, respectivement, angulaire, du pendule est critique. Ces deux variables, X_Φ et X_e , représentent respectivement la *position angulaire critique* et la *position horizontale critique* du pendule.

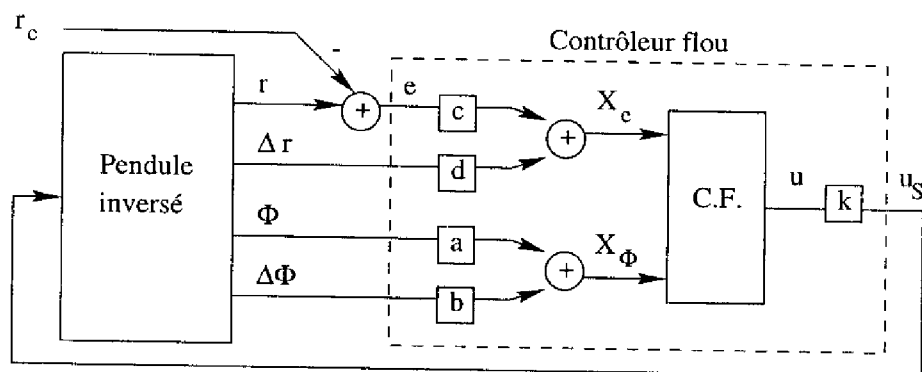


FIG. 3.7 - Contrôleur flou basé sur une fusion mathématique

Ce problème de commande est loin d'être trivial car deux variables sont à réguler mais l'on ne dispose que d'une seule action de commande. La dynamique de Φ étant beaucoup plus rapide que celle de r , la stratégie adoptée est la suivante : nous cherchons dans un premier temps à équilibrer le pendule (position verticale haute)

puis à l'amener progressivement vers sa position de référence en le déséquilibrant du "bon côté", c'est à dire vers celui qui entraînera une valeur de commande telle que le pendule se déplacera vers r_e .

Nous pouvons alors facilement écrire les cinq règles suivantes afin de commander le pendule à la fois horizontalement et verticalement :

$$\left\{ \begin{array}{l} R_1: \text{ Si } X_\Phi \text{ est Négatif alors } u \text{ est Négatif,} \\ R_2: \text{ Si } X_\Phi \text{ est Positif alors } u \text{ est Positif,} \\ R_3: \text{ Si } X_\Phi \text{ est Zéro et } X_e \text{ est Négatif alors } u \text{ est Négatif,} \\ R_4: \text{ Si } X_\Phi \text{ est Zéro et } X_e \text{ est Zéro alors } u \text{ est Zéro,} \\ R_5: \text{ Si } X_\Phi \text{ est Zéro et } X_e \text{ est Positif alors } u \text{ est Positif.} \end{array} \right. \quad (3.12)$$

Ces cinq règles traduisent bien que l'objectif prioritaire est la stabilisation verticale du pendule: X_e n'est considéré que lorsque X_Φ est nul. Examinons maintenant la 3^{ème} règle (R_3): "Si X_Φ est Zéro et X_e est Négatif alors u est Négatif". Cette commande négative entraîne X_Φ positif. La seconde règle R_2 est alors activée: $u_s = k.u$ devient positif afin d'équilibrer le pendule et la position du chariot augmente comme souhaité. Le raisonnement est similaire pour la 5^{ème} règle.

Les cinq règles précédentes (3.12) peuvent s'écrire de façon plus compacte sous forme de table :

X_Φ		N	Z	P
	N		N	
X_e	Z	N	Z	P
	P		P	

TAB. 3.1 - Base de règles du contrôleur flou (C.F.) $\rightarrow u$

La simulation du pendule inversé est réalisée sous MATLAB à partir des équations non linéaires (3.10). Ces équations différentielles sont intégrées numériquement par une méthode de Runge-Kutta. Les valeurs numériques des coefficients a_{ij} et b_i pour $i = 3, 4$ et $j = 2, \dots, 5$ sont :

$$\begin{array}{llll} a_{32} = -1.93 & a_{33} = -1.99 & a_{34} = 0.009 & a_{35} = 0.063 \\ a_{42} = 36.99 & a_{43} = 6.26 & a_{44} = -0.174 & a_{45} = -0.1972 \\ b_3 = 0.5128 & b_4 = -1.61 & & \end{array} \quad (3.13)$$

Les limitations physiques du pendule inversé ici considéré sont $r \in [-0.5, +0.5]$ mètres et $u_s \in [-10, +10]$ volts.

Les ensembles flous en entrée sont des triangles (N, Z et P) régulièrement répartis sur l'univers de discours normalisé $[-1, +1]$. Les ensembles flous en sortie sont des singletons régulièrement répartis sur $[-1, +1]$. Les coefficients de fusion a, b, c et d sont respectivement égaux à 12, 2.5, 0.7 et 2. Le gain en sortie k est égal à 5.

La réponse temporelle, pour une consigne en position de 30 cm, est donnée dans la figure 3.8.

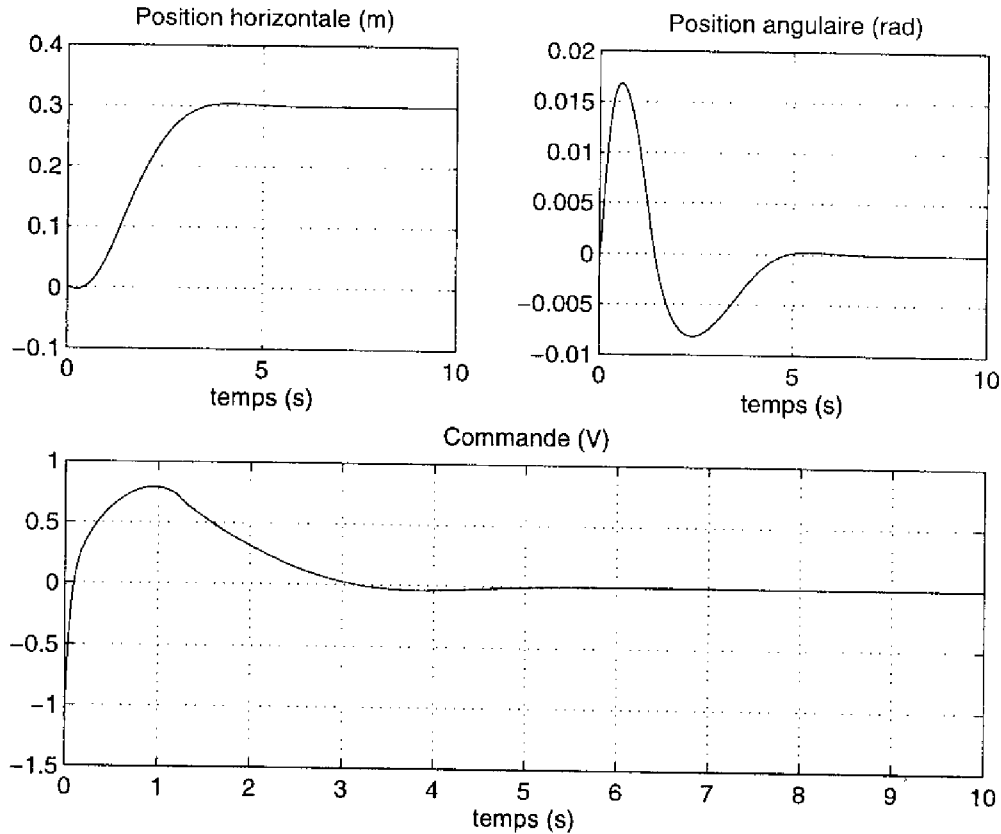


FIG. 3.8 – Réponse temporelle à une consigne r_c de 30 cm

Les performances obtenues avec ce contrôleur flou contenant seulement cinq règles sont tout à fait satisfaisantes. De plus, les règles sont très lisibles et facilement interprétables par un non-spécialiste en automatique.

3.1.2 Fusion symbolique

L'autre méthode, pour réduire la dimension du problème de commande, est fondée sur l'utilisation d'ensembles flous multidimensionnels [Ben93, FGB94, JGBF95]: les variables en entrée du contrôleur flou sont combinées de façon symbolique. Ainsi, plutôt que de définir un ensemble de termes linguistiques pour chacune des variables en entrée, on définit un ensemble de termes linguistiques pour un groupe de variables d'entrée, chacun de ces termes linguistiques étant alors représenté par un ensemble flou multidimensionnel.

Supposons que les variables d'entrée v_1, v_2, \dots, v_n soient fusionnées symboliquement. Si $v = (v_1, v_2, \dots, v_n)$ prend m valeurs linguistiques, alors il est nécessaire de définir m ensembles flous de dimension n , chacun d'eux décrivant un des termes linguistiques.

Une méthode simple pour construire des ensembles flous multidimensionnels se base sur le maillage de Delaunay [GH89].

Maillage de Delaunay

Soit \mathcal{E}_n l'espace euclidien affine de dimension n et \mathcal{P} un ensemble de m points $\{P_1, P_2, \dots, P_m\}$ de \mathcal{E}_n . Le diagramme de Voronoï associé à \mathcal{P} est une collection $\{V_1, V_2, \dots, V_m\}$ de polyèdres convexes recouvrant \mathcal{E}_n où V_i contient tous les points de \mathcal{E}_n qui ont P_i comme plus proche voisin dans l'ensemble \mathcal{P} .

$$\forall p \in \mathcal{E}_n, p \in V_i \Rightarrow \forall p_{j \neq i} \in \mathcal{P}, \delta(p, p_i) \leq \delta(p, p_j) \quad (3.14)$$

où δ est la distance euclidienne :

$$\delta(u, v) = \sqrt{\sum_{i=1}^n (u_i - v_i)^2} \quad \text{avec} \quad u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \quad \text{et} \quad v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \quad (3.15)$$

Le dual géométrique du diagramme de Voronoï est appelé *maillage de Delaunay* et est obtenu en reliant par des segments de droite tous les points P_i pour lesquels les polyèdres de Voronoï associés ont une face commune.

Le maillage de Delaunay se compose alors de n -simplexes partitionnant l'enveloppe convexe de \mathcal{P} . Un n -simplexe est caractérisé par ses $n+1$ sommets; Ainsi, dans le cas d'un espace euclidien affine de dimension 2, le domaine défini par l'ensemble des points de \mathcal{P} est partitionné à l'aide de triangles. On parle alors de *triangulation de Delaunay*. Une illustration du cas bidimensionnel est donnée dans la figure 3.9.

Construction d'ensembles flous multidimensionnels

Soit $\mathcal{P} = \{P_1, P_2, \dots, P_m\}$ un ensemble de m points définis dans l'espace euclidien affine de dimension n

$$P_i = \begin{bmatrix} x_{i,1} \\ x_{i,2} \\ \vdots \\ x_{i,n} \end{bmatrix} \in \mathcal{E}_n \quad (3.16)$$

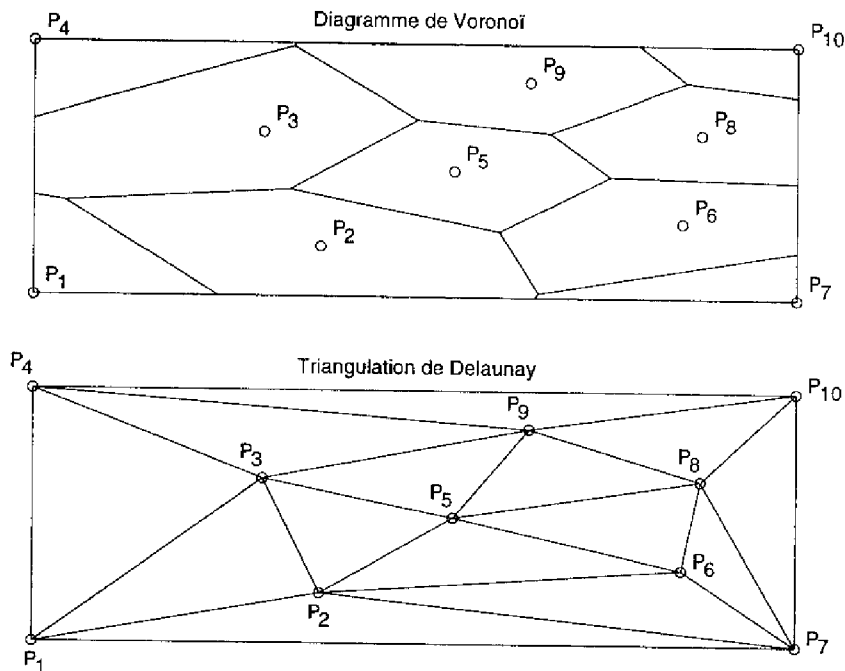


FIG. 3.9 – La triangulation de Delaunay est le dual géométrique du diagramme de Voronoï

L'ensemble flou décrivant un terme linguistique A est défini à partir de la triangulation de Delaunay du nuage de points \mathcal{P} comme suit:

1. Construction du "vecteur modal" associé au terme linguistique A :

$$\mathcal{M}_A = \begin{bmatrix} \mu_A(P_1) \\ \vdots \\ \mu_A(P_m) \end{bmatrix} \quad (3.17)$$

avec, pour $i = 1, \dots, m$, $\mu_A(P_i) = 1$ si P_i est décrit par le terme linguistique A , $\mu_A(P_i) = 0$ dans le cas contraire.

2. Sur chaque n -simplexe défini par la triangulation de Delaunay, la fonction d'appartenance $\mu_A(p)$ est construite selon une interpolation linéaire entre les $n + 1$ sommets du n -simplexe :

$$\mu_A(p) = a_1.x_1 + a_2.x_2 + \dots + a_n.x_n + a_{n+1} \quad (3.18)$$

Les coefficients a_1, \dots, a_{n+1} sont calculés à partir des $n + 1$ sommets $P_i = (x_{i,1}, x_{i,2}, \dots, x_{i,n})^T \in \mathcal{E}_n$ caractérisant le n -simplexe, par résolution de l'équation :

$$A = C^{-1}.B$$

$$\text{où } \mathcal{A} = \begin{pmatrix} a_1 \\ \vdots \\ a_{n+1} \end{pmatrix}, \mathcal{C} = \begin{pmatrix} x_{1,1} & \cdots & x_{1,n} & 1 \\ \vdots & & \vdots & \vdots \\ x_{n+1,1} & \cdots & x_{n+1,n} & 1 \end{pmatrix} \text{ et } \mathcal{B} = \begin{pmatrix} \mu_A(P_1) \\ \vdots \\ \mu_A(P_{n+1}) \end{pmatrix}.$$

Afin d'illustrer la construction d'un ensemble flou à partir du maillage de Delaunay, considérons la triangulation de Delaunay de 5 points caractéristiques définis dans l'espace bidimensionnel $X_1 \times X_2$. Ce maillage est représenté dans la figure 3.10. Soit A un terme linguistique tel que les points P_3 , P_4 et P_5 décrivent complètement ce label ($\mu_A(P_3) = \mu_A(P_4) = \mu_A(P_5) = 1$) et les autres points P_1 et P_2 ne décrivent absolument pas ce terme ($\mu_A(P_1) = \mu_A(P_2) = 0$): $\mathcal{M}_A = [0 \ 0 \ 1 \ 1 \ 1]^T$.

La fonction d'appartenance associée au terme linguistique A est :

- sur le n-simplexe (P_3, P_4, P_5) , $\mu_A(p) = 1$,
- sur le n-simplexe (P_1, P_3, P_4) , $\mu_A(p) = x_1 + x_2$,
- sur le n-simplexe (P_2, P_3, P_5) , $\mu_A(p) = -x_1 + x_2 + 1$,
- sur le n-simplexe (P_1, P_2, P_3) , $\mu_A(p) = 2x_2$.

Elle est représentée dans la figure 3.10.

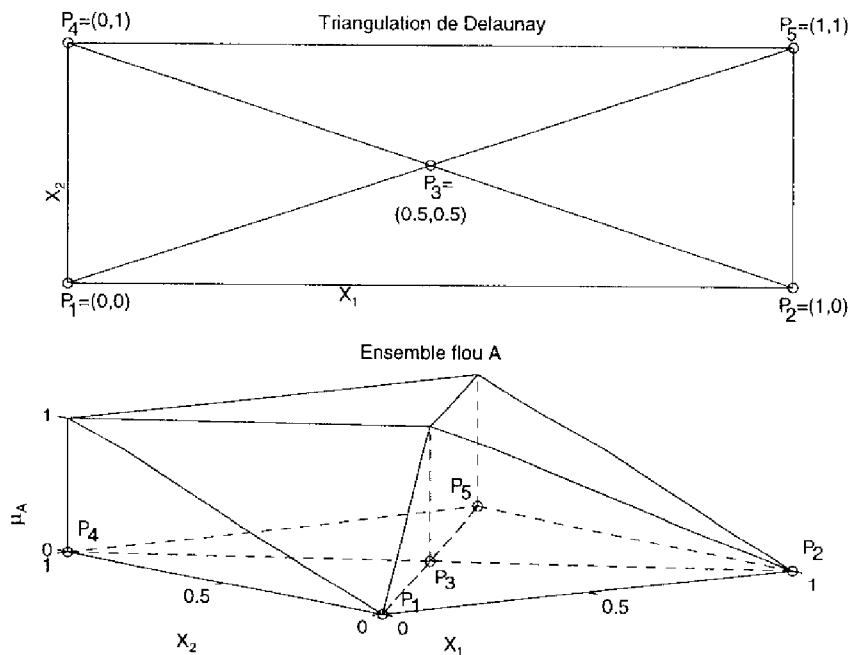


FIG. 3.10 – La fonction d'appartenance est définie à partir du maillage de Delaunay.

Illustration sur le pendule inversé

Plutôt que de combiner mathématiquement les variables Φ , $\Delta\Phi$ et e , Δe , ces dernières sont fusionnées symboliquement. Les règles (3.12) s'écrivent alors de la façon suivante :

$$\left\{ \begin{array}{l} R_1: \text{ Si } (a\Phi, b\Delta\Phi) \text{ est } \textit{Négatif} \text{ alors } u \text{ est } \textit{Négatif}, \\ R_2: \text{ Si } (a\Phi, b\Delta\Phi) \text{ est } \textit{Positif} \text{ alors } u \text{ est } \textit{Positif}, \\ R_3: \text{ Si } (a\Phi, b\Delta\Phi) \text{ est } \textit{Zéro} \text{ et } (ce, d\Delta e) \text{ est } \textit{Négatif} \text{ alors } u \text{ est } \textit{Négatif}, \\ R_4: \text{ Si } (a\Phi, b\Delta\Phi) \text{ est } \textit{Zéro} \text{ et } (ce, d\Delta e) \text{ est } \textit{Zéro} \text{ alors } u \text{ est } \textit{Zéro}, \\ R_5: \text{ Si } (a\Phi, b\Delta\Phi) \text{ est } \textit{Zéro} \text{ et } (ce, d\Delta e) \text{ est } \textit{Positif} \text{ alors } u \text{ est } \textit{Positif}. \end{array} \right. \quad (3.19)$$

Les labels linguistiques *Négatif*, *Zéro* et *Positif* apparaissant dans la partie pré-misse des règles (3.19) sont représentés par des ensembles flous bidimensionnels. Ces derniers sont construits à partir d'une triangulation de Delaunay d'un ensemble de points $\mathcal{P} = \{P_1, P_2, \dots, P_9\}$ régulièrement répartis sur l'univers de discours normalisé $[-1,1] \times [-1,1]$. Ils sont représentés dans la figure 3.11.

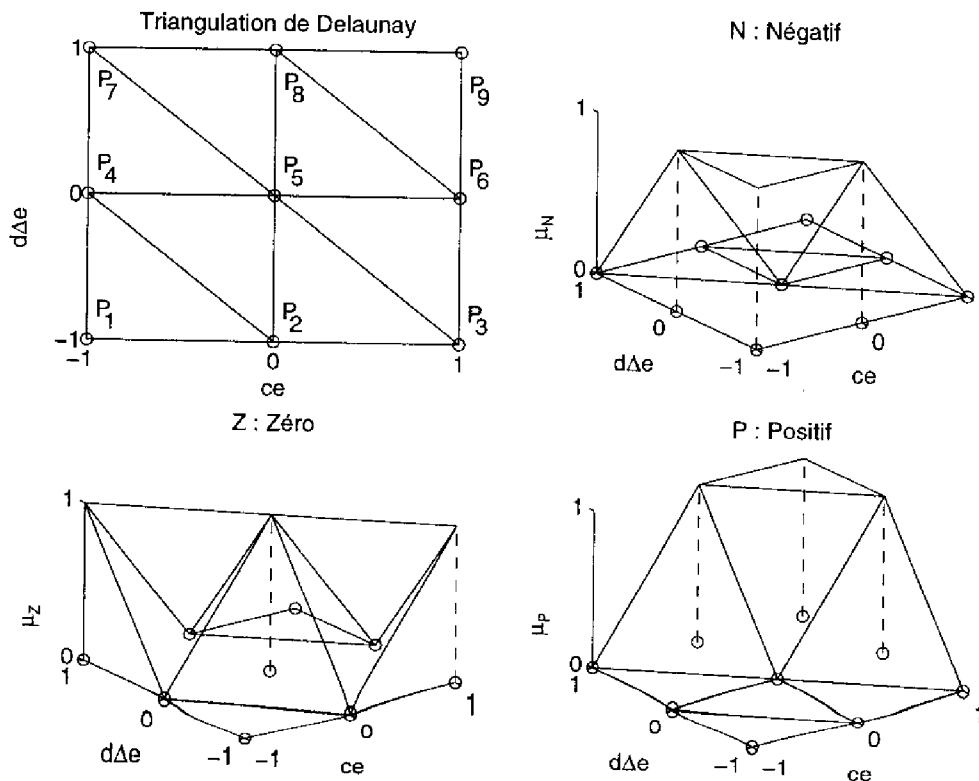


FIG. 3.11 - Les fonctions d'appartenance sont définies à partir d'une triangulation de Delaunay.

Les vecteurs modaux associés aux ensembles flous N, Z et P sont respectivement

$$\mathcal{M}_N = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathcal{M}_Z = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad \text{et} \quad \mathcal{M}_P = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}. \quad (3.20)$$

Remarquons que $\mathcal{M}_N + \mathcal{M}_Z + \mathcal{M}_P = I = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]^T$, autrement dit, pour chacun des points P_i de \mathcal{P} il ne correspond qu'un et un seul label linguistique. Les ensembles flous N, Z et P forment une partition floue :

$$\forall p \in [-1,1] \times [-1,1], \mu_N(p) + \mu_Z(p) + \mu_P(p) = 1 \quad (3.21)$$

Les résultats étant identiques à ceux obtenus avec le contrôleur flou basé sur la fusion mathématique (les valeurs des gains a, b, c, d et k sont inchangées) voir la figure 3.8.

3.2 Structuration de la loi de commande

Nous allons tâcher d'exploiter la connaissance disponible sur le procédé à commander afin de structurer la loi de commande floue. Grâce à la définition de structures de commande particulières, décentralisées ou hiérarchiques, le problème de la synthèse d'un contrôleur flou (CF) multivariable peut être décomposé en plusieurs sous-problèmes. Ainsi, la synthèse d'un contrôleur flou multivariable se décompose en un ensemble de sous-problèmes dont les objectifs sont la synthèse de CFs monovariables dans le cas d'une structure décentralisée, ou encore la détermination d'un ensemble de CFs de plus faible dimension dans le cas d'une structure hiérarchique.

3.2.1 Décentralisation

Un système multivariable se distingue des processus monovariables par la présence de couplages entre les entrées et les sorties du procédé : *une entrée peut agir sur plus d'une sortie, et une sortie peut dépendre de plus d'une entrée*. S'il est possible de mettre en évidence des couplages forts reliant chacun des couples entrée-sortie du processus et si, pour chacun de ces couples, les couplages croisés, couplages reliant chacune des entrées des autres couples à la sortie du couple considéré, sont négligeables, alors la synthèse du CF multi-entrées multi-sorties se ramène à la synthèse d'une collection de CFs mono-entrée mono-sortie, chacun d'eux correspondant à une paire entrée-sortie. Le problème de commande multivariable se décompose alors en un ensemble de problèmes monovariables beaucoup plus simples à résoudre.

3.2.2 Hiérarchisation

Contrôleur flou hiérarchique

Raju et al. [RZK91, RZ90] proposent de hiérarchiser la base de règles de sorte que le nombre de règles augmente linéairement et non plus exponentiellement avec le nombre de variables.

L'ensemble de règles est construit de façon hiérarchique: les variables en entrée du contrôleur flou ne sont plus traitées en parallèle mais sont réparties selon différents niveaux de raisonnement. Le problème de commande est donc résolu séquentiellement.

Les règles du premier niveau sont de la forme :

Si (x_1 est F_1^1 et x_2 est $F_2^1 \dots$ et x_{n_1} est $F_{n_1}^1$) **alors** (la sortie u_1 est C_1)

Les règles du niveau i ($i > 1$) sont de la forme :

Si (x_{N_i+1} est F_1^i et \dots et $x_{N_i+n_i}$ est $F_{n_i}^i$ et u_{i-1} est G_{i-1}) **alors** (la sortie u_i est C_i)

où

$$N_i = \sum_{j=1}^{i-1} n_j \leq n \quad (3.22)$$

avec n_j le nombre de variables intervenant au niveau j , et u_{i-1} la sortie du niveau précédent ($i - 1$) considérée comme variable d'entrée pour le niveau i .

A chaque niveau i ($i > 1$), une ou plusieurs variables sont ajoutées à la sortie du niveau précédent pour le développement de l'ensemble des règles du niveau i .

Théorème 3.1 *Soit n le nombre de variables, L le nombre de niveaux de la structure hiérarchique et n_i le nombre de variables apparaissant dans le niveau i , y compris la variable de sortie du niveau $i - 1$ (si $i > 1$), alors le nombre total de règles k est*

$$k = \sum_{i=1}^L m^{n_i} \quad (3.23)$$

où m est le nombre de valeurs linguistiques que peuvent prendre chacune des variables, et

$$n_1 + \sum_{i=2}^L (n_i - 1) = n \quad (3.24)$$

Si le nombre de variables intervenant dans chaque niveau est constant, $n_i = t$ est un entier constant ($i = 1, 2, \dots, L$), le nombre de niveaux étant alors égal à

$$L = \frac{n-t}{t-1} + 1 \quad (3.25)$$

le nombre total de règles est donné par

$$k = \left(\frac{n-t}{t-1} + 1\right).m^t \quad (3.26)$$

Ainsi, grâce à la structure hiérarchique de la base de règles, le nombre de règles assurant la stricte complétude de cette dernière, est une **fonction linéaire du nombre n de variables** et non plus une fonction exponentielle de n comme ce serait le cas pour une base de règles classique.

Raju et al. démontrent alors que *le nombre total de règles est minimal si, dans chacun des niveaux successifs, une et une seule variable supplémentaire est considérée* (voir la figure 3.12).

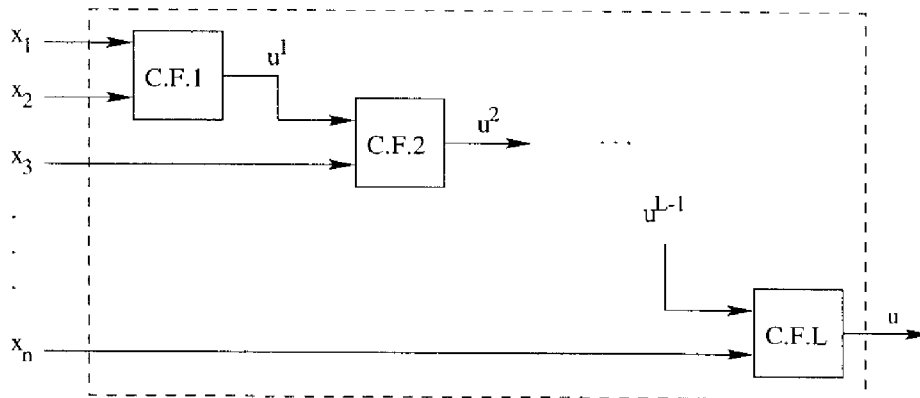


FIG. 3.12 - Contrôleur flou hiérarchique

Théorème 3.2 Dans une structure hiérarchique contenant n variables, si $m \geq 2$ et $n_i \geq 2$ (m étant le nombre de valeurs linguistiques pour chaque variable et n_i le nombre de variables intervenant dans le niveau i), le nombre total de règles atteindra sa valeur minimale pour $n_i = t = 2$ et sa valeur maximale pour $n_i = n_1 = n$.

En effet, le nombre total de règles dans la structure hiérarchique sera réduit si nous divisons chaque niveau contenant au moins trois variables en deux niveaux, l'un des deux contenant deux variables. En répétant ce procédé pour tous les niveaux, il est clair que le nombre total de règles atteindra sa valeur minimale si chaque niveau contient seulement deux variables. Un raisonnement inverse permet de montrer que le nombre total de règles atteindra sa valeur maximale si $n_i = n_1 = n$, celui-ci augmentant lorsque nous combinons deux niveaux en un seul niveau.

La structure hiérarchique diminue considérablement le nombre de règles. De plus, elle permet de rajouter ou d'enlever une variable très facilement, c'est à dire sans altérer les autres règles de la base. Il faut également noter que le procédé adopté dans la structure hiérarchique est similaire au type de raisonnement d'un opérateur : une première décision est prise sur la base d'un premier groupe de variables puis est modifiée en fonction de la valeur des autres variables. Il est en effet difficile de décider d'une action en considérant "d'un seul coup" toutes les variables entrant en jeu.

La hiérarchisation du traitement des variables reste néanmoins une étape non triviale et nécessite une connaissance experte sur le procédé à commander ou encore une analyse qualitative du processus. Il n'existe donc aucune règle générale pour les hiérarchiser. Il semble cependant plus naturel de placer les variables les plus "importantes" en bas de la hiérarchie puisque le dernier contrôleur flou constitue la chaîne la plus directe. En effet, ce sont les dernières variables qui influencent le plus la décision finale.

Retenons que la structure hiérarchique composée de contrôleurs flous à deux entrées (figure 3.12) est celle qui offre une réduction maximale du nombre de règles. C'est donc vers ce type de structure que nous nous orientons.

Illustration sur le pendule inversé

La structure du contrôleur flou hiérarchique est représentée dans la figure 3.13.

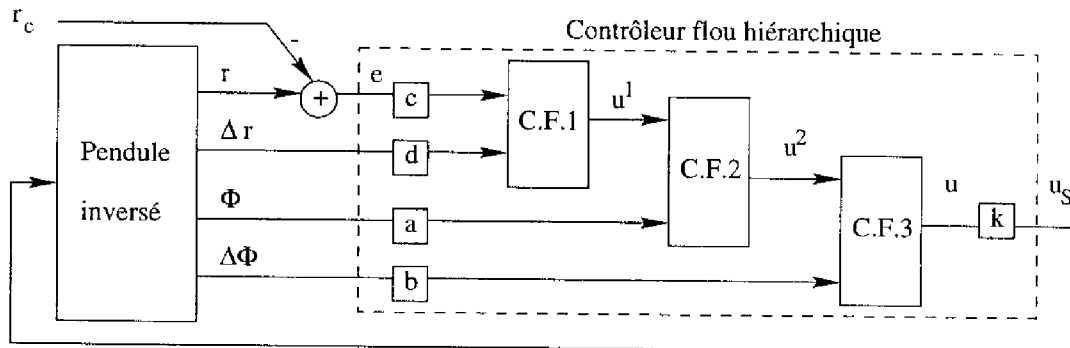


FIG. 3.13 - *Contrôleur flou hiérarchique*

Les variables en entrée du contrôleur flou sont ici traitées selon leur dynamique croissante. Le contrôleur flou hiérarchique est composé de trois contrôleurs flous connectés en série. Les bases de règles correspondantes sont représentées dans les tables 3.2, 3.3 et 3.4. Le nombre total de règles est de $9 + 5 + 9 = 23$ règles.

L'objectif du premier contrôleur flou C.F.1 est d'amener le chariot vers sa position de référence r_c . La première action u^1 consiste donc à déséquilibrer le pendule dans le "bon sens". Ce déséquilibre doit avoir pour conséquence le déplacement du chariot dans la direction souhaitée.

$d\Delta e$		N	Z	P
ce	N	N	N	Z
	Z	N	Z	P
	P	Z	P	P

TAB. 3.2 - Base de règles du C.F.1 $\rightarrow u^1$

L'objectif du second contrôleur flou C.F.2 est d'équilibrer le pendule si celui-ci ne l'est pas encore. La première décision de commande u^1 est donc conservée si la position angulaire du pendule est *zéro*, mais si le pendule n'est pas équilibré, la nouvelle commande u^2 est telle que le pendule converge vers une position verticale haute.

$a\Phi$		N	Z	P
u^1	N		N	
	Z	N	Z	P
	P		P	

TAB. 3.3 - Base de règles du C.F.2 $\rightarrow u^2$

Quant au 3^{ème} contrôleur flou C.F.3 il a pour objectif d'affiner la commande précédente en considérant une variable supplémentaire $\Delta\Phi$, la vitesse angulaire du pendule.

$d\Delta\Phi$		N	Z	P
u^2	N	N	N	Z
	Z	N	Z	P
	P	Z	P	P

TAB. 3.4 - Base de règles du C.F.3 $\rightarrow u$

Ainsi, si $\Delta\Phi$ est *zéro*, il n'y a aucune raison de modifier la précédente commande u^2 . De même, si $\Delta\Phi$ est *négatif* (respectivement *positif*) et u^2 est *négatif* (respectivement *positif*) alors la commande précédente n'a pas à être révisée puisqu'elle équilibre le pendule. Par contre, si la commande u^2 est *zéro* et que le pendule a tendance à se déséquilibrer alors il faut choisir une commande en conséquence. Considérons maintenant le cas plus délicat où u^2 est *négatif* et $\Delta\Phi$ est *positif*. La commande précédente a été choisie soit pour équilibrer le pendule ("Si $a\Phi$ est N alors u^2 est N") soit pour le déséquilibrer vers la gauche ("Si $a\Phi$ est Z et u^1 est N alors u^2 est N"). Dans les deux cas, puisque $\Delta\Phi$ est *positif*, il est inutile d'agir. En effet, dans le 1^{er} cas de figure, le pendule va converger naturellement vers sa position

verticale haute, et, dans le second cas, il va se déséquilibrer vers la gauche comme souhaité. Le raisonnement est similaire pour le cas u^2 négatif et $\Delta\Phi$ positif.

La réponse temporelle, pour une consigne en position de 30 cm, est représentée dans la figure 3.14.

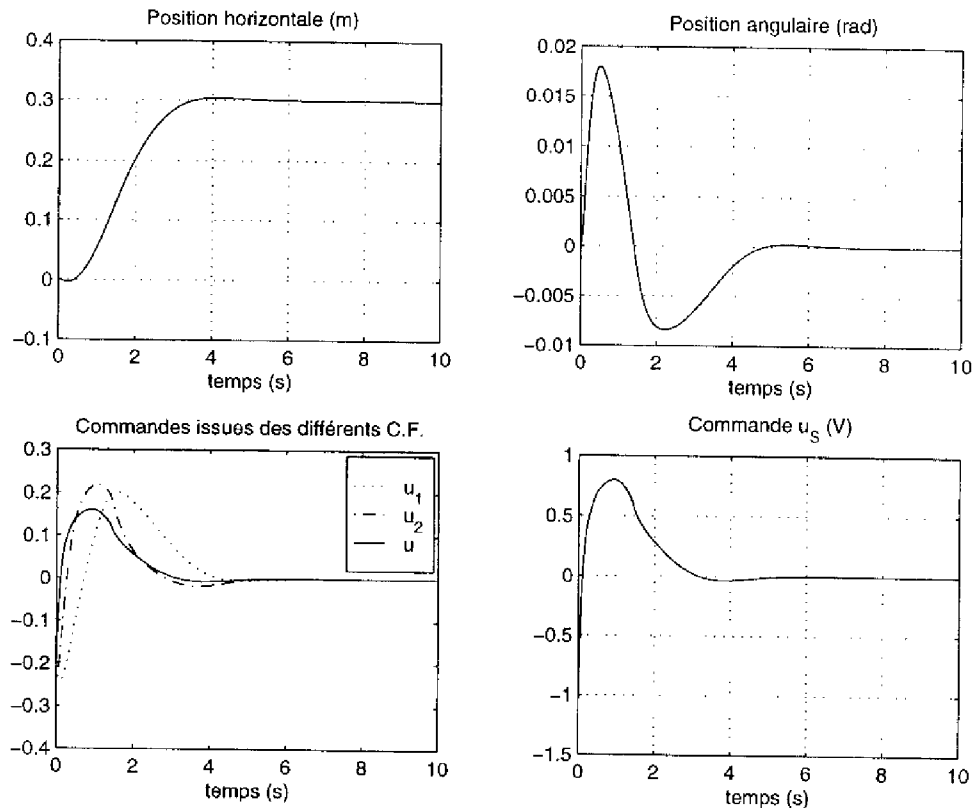


FIG. 3.14 – Réponse temporelle à une consigne r_c de 30 cm

Base de règles multi-niveaux

L'idée de base ici exploitée est que, même un système très complexe de grande dimension peut avoir un comportement qui dépend localement que d'un sous-ensemble des variables le régissant, ce sous-ensemble variant selon le domaine de l'espace d'état considéré.

L'espace d'état est divisé en plusieurs domaines et à chaque domaine correspond une "base de règles locale". Celle-ci ne considère qu'une partie des variables du système et par conséquent contient un plus petit nombre de règles. Elle est appelée *sous-base de règles*. Dans les niveaux supérieurs (un ou plusieurs) appelés *méta-niveaux*, des méta-règles décident quelles sous-bases de règles activer.

Tous les niveaux, excepté le dernier, sont des méta-niveaux. L'espace d'état global est divisé en domaines et ces domaines forment une partition de X .

$$X = \cup_{i=1}^p D_i \quad (3.27)$$

où les D_i sont les domaines. Chacun de ces domaines peut de nouveau être partitionné :

$$D_i = \cup_{j=1}^{p_i} D_{ij} \quad (3.28)$$

etc ...

Au niveau inférieur, on trouve les sous-bases de règles. Cependant, au lieu d'avoir en partie prémisses des règles $X = (x_1, x_2, \dots, x_n)$, on aura $X_{ijk\dots}$, un sous-ensemble de variables appartenant à X .

Un exemple très simple est la base de règles à deux niveaux suivante :

1. Méta-niveau (R_0) :

Si x_1 est A_1^1 et x_2 est A_2^1 **alors** considérer le domaine D_1 .
Si x_1 est A_1^2 et x_2 est A_2^2 **alors** considérer le domaine D_2 .

2. 2^{ème} niveau (R_1 et R_2) :

- Sous-base de règles R_1 pour le domaine D_1 :
Si x_3 est A_3^1 et x_4 est A_4^1 **alors** u est B_1 .
Si x_3 est A_3^2 et x_4 est A_4^2 **alors** u est B_2 .
- Sous-base de règles R_2 pour le domaine D_2 :
Si x_5 est A_5^1 **alors** u est B_3 .
Si x_5 est A_5^2 **alors** u est B_4 .

Dans la forme la plus simple de cette méthode [SGB93], les domaines forment une partition booléenne de l'espace d'état, de sorte que pour une situation donnée ne correspond qu'un seul et unique domaine. Par conséquent, une seule sous-base de règles sera activée. Cependant, la partition de l'espace d'état peut être floue [KH93b], permettant une plus grande flexibilité dans le choix des sous-bases de règles à activer mais soulevant par la même occasion la question suivante : *comment combiner les sous-bases de règles faisant intervenir différents sous-ensembles de variables ?* Kóczy propose de définir un nouveau sous-ensemble de variables incluant toutes celles intervenant dans les sous-bases de règles ayant été activées. Toutes les règles sont alors représentées avec ce même sous-ensemble en prenant leur extension cylindrique, formant ainsi une nouvelle sous-base de règles. Une autre façon, plus simple, d'aborder le problème est de considérer séparément chacune des sous-bases de règles activées. Plusieurs conclusions sont alors calculées et la valeur de commande finale est obtenue par leur moyenne pondérée par le degré d'activation des différentes sous-bases de règles.

Illustration sur le pendule inversé

Nous considérons de nouveau le problème du pendule inversé. La base de règles comporte alors trois niveaux :

1. Méta-niveau (R_0) :

$$\left\{ \begin{array}{l} \text{Si } a\Phi \text{ est N alors considérer le domaine } D_1, \\ \text{Si } a\Phi \text{ est Z alors considérer le domaine } D_2, \\ \text{Si } a\Phi \text{ est P alors considérer le domaine } D_3. \end{array} \right. \quad (3.29)$$

2. 2^{ème} niveau (R_1, R_2 et R_3) :

• Sous-base de règles R_1 pour le domaine D_1 :

$$\left\{ \begin{array}{l} \text{Si } b\Delta\Phi \text{ est N alors } u \text{ est N,} \\ \text{Si } b\Delta\Phi \text{ est Z alors } u \text{ est N,} \\ \text{Si } b\Delta\Phi \text{ est P alors } u \text{ est Z.} \end{array} \right. \quad (3.30)$$

• Sous-base de règles R_2 pour le domaine D_2 :

$$\left\{ \begin{array}{l} \text{Si } b\Delta\Phi \text{ est N alors } u \text{ est N,} \\ \text{Si } b\Delta\Phi \text{ est Z alors considérer le domaine } D_{21}, \\ \text{Si } b\Delta\Phi \text{ est P alors } u \text{ est P.} \end{array} \right. \quad (3.31)$$

• Sous-base de règles R_3 pour le domaine D_3 :

$$\left\{ \begin{array}{l} \text{Si } b\Delta\Phi \text{ est N alors } u \text{ est Z,} \\ \text{Si } b\Delta\Phi \text{ est Z alors } u \text{ est P,} \\ \text{Si } b\Delta\Phi \text{ est P alors } u \text{ est P.} \end{array} \right. \quad (3.32)$$

3. 3^{ème} niveau (R_{21}) : sous-base de règles R_{21} dont l'objectif est de déséquilibrer le pendule du "bon côté" afin de conduire le chariot vers sa position de référence.

$d\Delta e$	N	Z	P
ce	N	N	Z
	Z	N	P
	P	Z	P

TAB. 3.5 – Commande u correspondante au domaine D_{21}

La base de règles complète comporte alors $3+(3+3+3)+9 = 21$ règles. Remarquons qu'au 2^{ème} niveau on a au maximum 2 sous-bases de règles activées : R_1 et R_2 ou R_2 et R_3 . Notons μ_i le degré d'activation de la sous-base de règles R_i :

$$\mu_1 = \mu_N(a\Phi) \quad \mu_2 = \mu_Z(a\Phi) \quad \mu_3 = \mu_P(a\Phi) \quad \mu_{21} = \mu_Z(b\Delta\Phi) \quad (3.33)$$

Notons u_1, u_2, u_3 et u_{21} les commandes calculées au niveau des domaines D_1, D_2, D_3 et D_{21} respectivement. La commande u_2 est calculée comme suit, avec -1 la valeur du singleton N et $+1$ celle du singleton P :

$$u_2 = \frac{\mu_N(b\Delta\Phi).(-1) + \mu_{21}.u_{21} + \mu_P(b\Delta\Phi).(1)}{\mu_N(b\Delta\Phi) + \mu_{21} + \mu_P(b\Delta\Phi)} \quad (3.34)$$

Quant à la commande finale u , elle est obtenue soit directement par la sous-base de règle activée si celle-ci est unique, soit par une moyenne pondérée des deux sous-bases de règles activées R_i et R_j avec $j = i + 1$:

$$\begin{cases} u = u_i & \text{si une seule sous-base de règles } R_i \text{ est activée,} \\ u = \frac{\mu_i.u_i + \mu_j.u_j}{\mu_i + \mu_j} & \text{si deux sous-base de règles } R_i \text{ et } R_j \text{ sont activées.} \end{cases} \quad (3.35)$$

La réponse temporelle, pour une consigne en position de 30 cm, ainsi que le degré d'activation des différentes sous-bases de règles, sont représentés dans la figure 3.15.

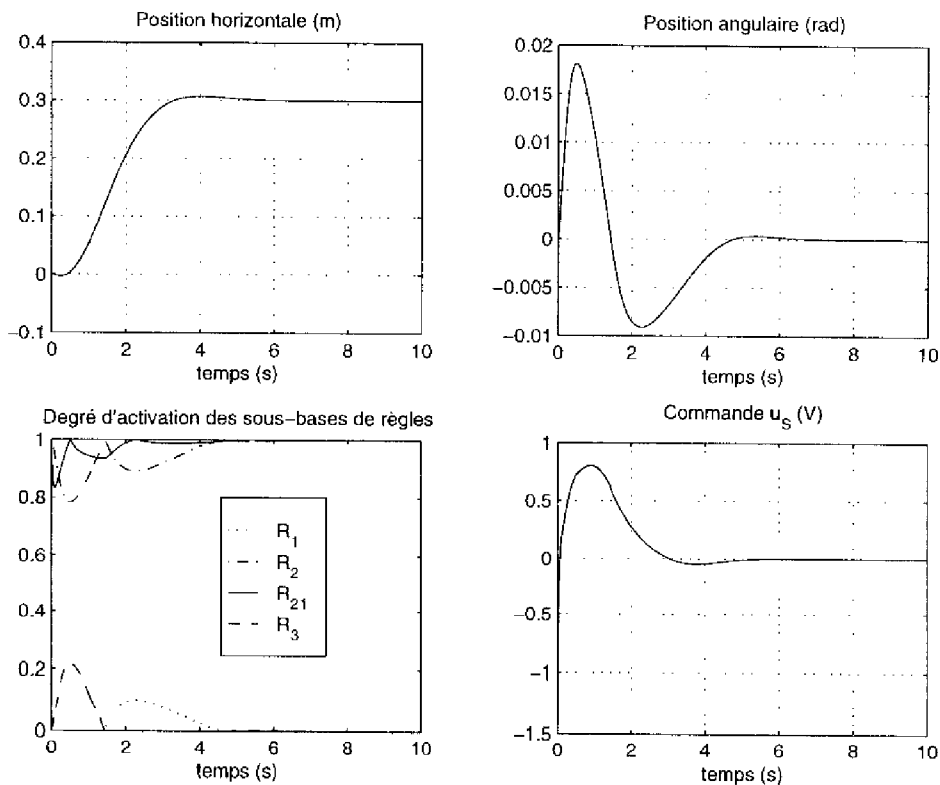


FIG. 3.15 – Réponse temporelle à une consigne r_c de 30 cm

3.3 Combinaison de techniques

Plus le nombre de variables en entrée du contrôleur flou est important, plus il est intéressant de combiner les différentes techniques présentées dans ce chapitre. On peut citer, à titre d'exemple, les combinaisons suivantes :

- fusion mathématique et hiérarchisation,
- fusion symbolique et hiérarchisation,
- fusion mathématique et fusion symbolique,
- fusion mathématique, symbolique et hiérarchisation.

La fusion mathématique (paragraphe 3.1.1) combinée au contrôleur flou hiérarchique (paragraphe 3.2.2) conduit à une approche dite d'*agrégation-hiérarchisation* [BJT95, Jam96] et est illustrée dans la figure 3.16.

Dans un premier temps, les variables sont fusionnées mathématiquement, comme dans la figure 3.1, puis sont organisées hiérarchiquement selon une structure similaire à celle de la figure 3.12. Décider quelles variables fusionner et où les positionner dans la structure hiérarchique est une étape clé difficile, principalement basée sur une connaissance du système. Rappelons tout de même que pour des problèmes de régulation, la fusion mathématique d'une erreur et de ses dérivées successives conduit généralement à des résultats intéressants.

La fusion symbolique (paragraphe 3.1.2) combinée au contrôleur flou hiérarchique (paragraphe 3.2.2) conduit à des résultats similaires en terme de réduction de la base de règles mais par contre nécessite des temps de calcul plus importants. En effet, si les ensembles flous multidimensionnels offrent davantage de degrés de liberté (ensemble de points à partir desquels le maillage de Delaunay est calculé), le calcul du degré d'appartenance d'un point à un ensemble flou multidimensionnel est un peu plus complexe.

Quelques comparaisons

Le nombre de règles (tableau 3.6) pour les méthodes de réduction par *fusion mathématique ou symbolique*, par *hiérarchisation* et par *fusion-hiérarchisation* est représenté graphiquement dans la figure 3.17.

Nombre de règles pour les méthodes ci-dessous	Nombre de variables $n > 1$	
	pair	impair
Fusion mathématique ou symbolique	$m^{\binom{n}{2}}$	$m^{\binom{n+1}{2}}$
Structure hiérarchique	$(n-1) * m^2$	
Fusion-hiérarchisation	$\left(\frac{n}{2} - 1\right) * m^2$	$\left(\frac{n+1}{2} - 1\right) * m^2$

TAB. 3.6 – Nombre de règles pour diverses méthodes de réduction

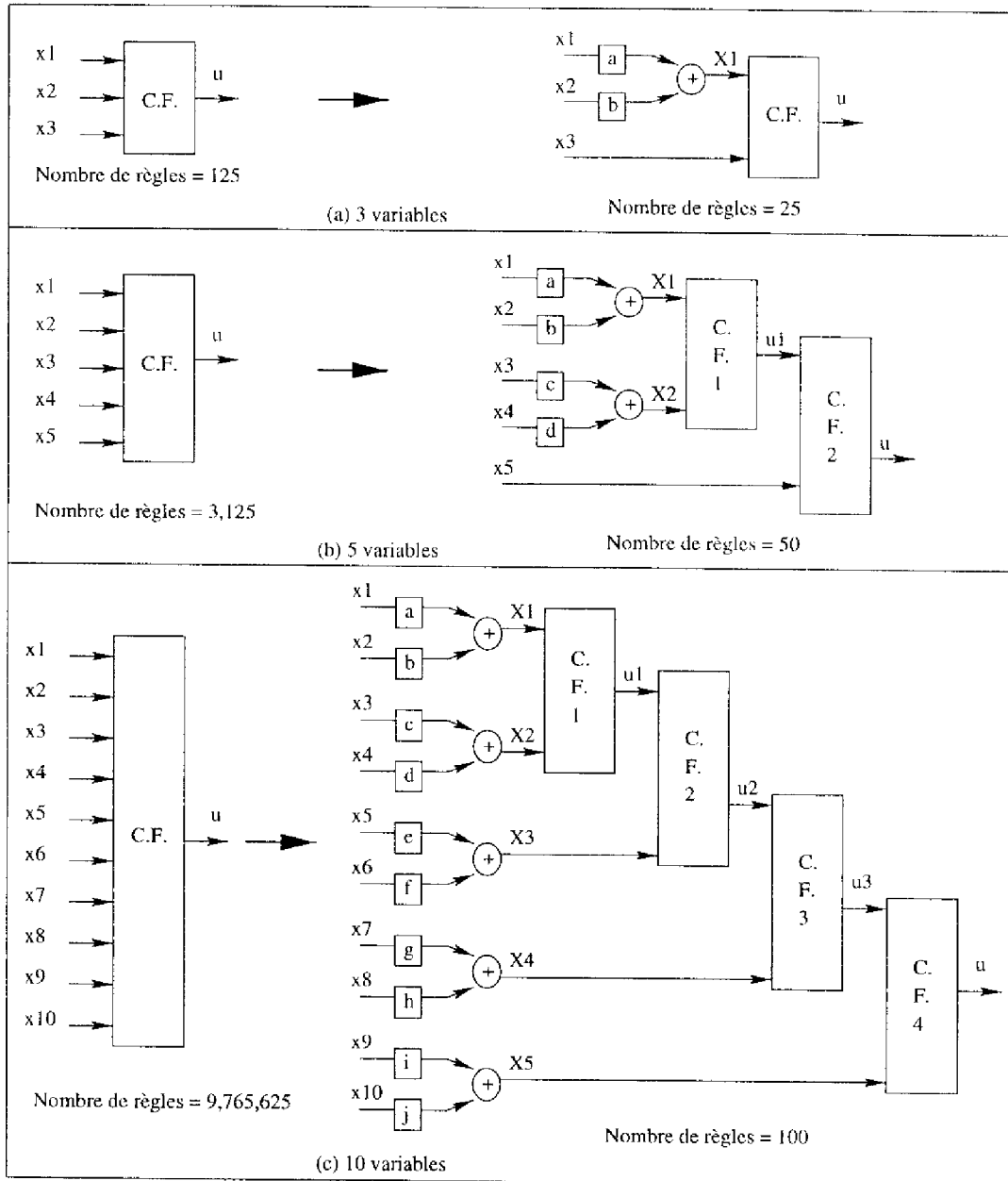


FIG. 3.16 – Méthode d'agrégation-hiérarchisation pour $n=3, 5, 10$ variables et $m = 5$ labels linguistiques par variable.

Nous appelons ici *fusion-hiérarchisation* la combinaison d'une des deux méthodes de réduction de la dimension avec la structuration hiérarchique. Précisons que les variables sont ici fusionnées par paire et qu'à chaque niveau de la hiérarchie une et une seule variable est rajoutée.

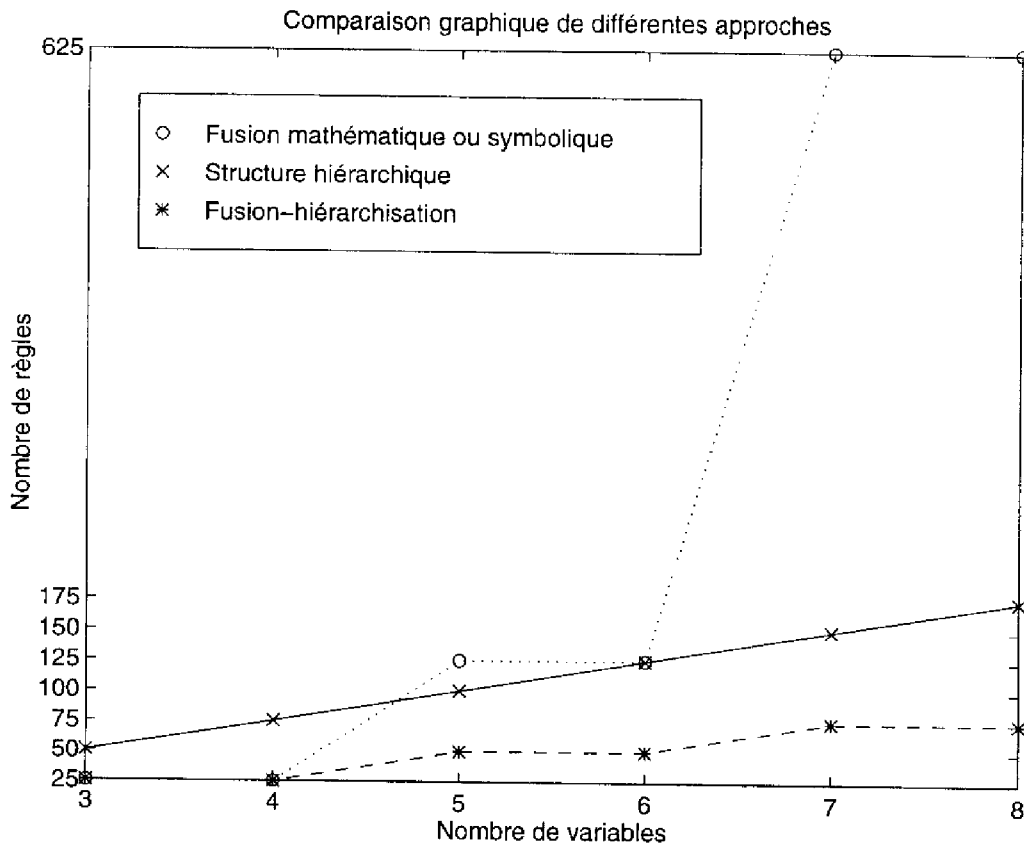


FIG. 3.17 – Comparaison de diverses méthodes de réduction de la base de règles avec $m = 5$ valeurs linguistiques par variable.

Les performances en terme de pourcentage de réduction (par rapport à une base de règles conventionnelle) des précédentes approches sont représentées graphiquement dans la figure 3.18 et sont calculées comme suit :

$$\% \text{ de réduction} = 100. \left[1 - \frac{\text{nombre de règles}}{m^n} \right] \quad (3.36)$$

avec n le nombre de variables et m le nombre de labels linguistiques par variable.

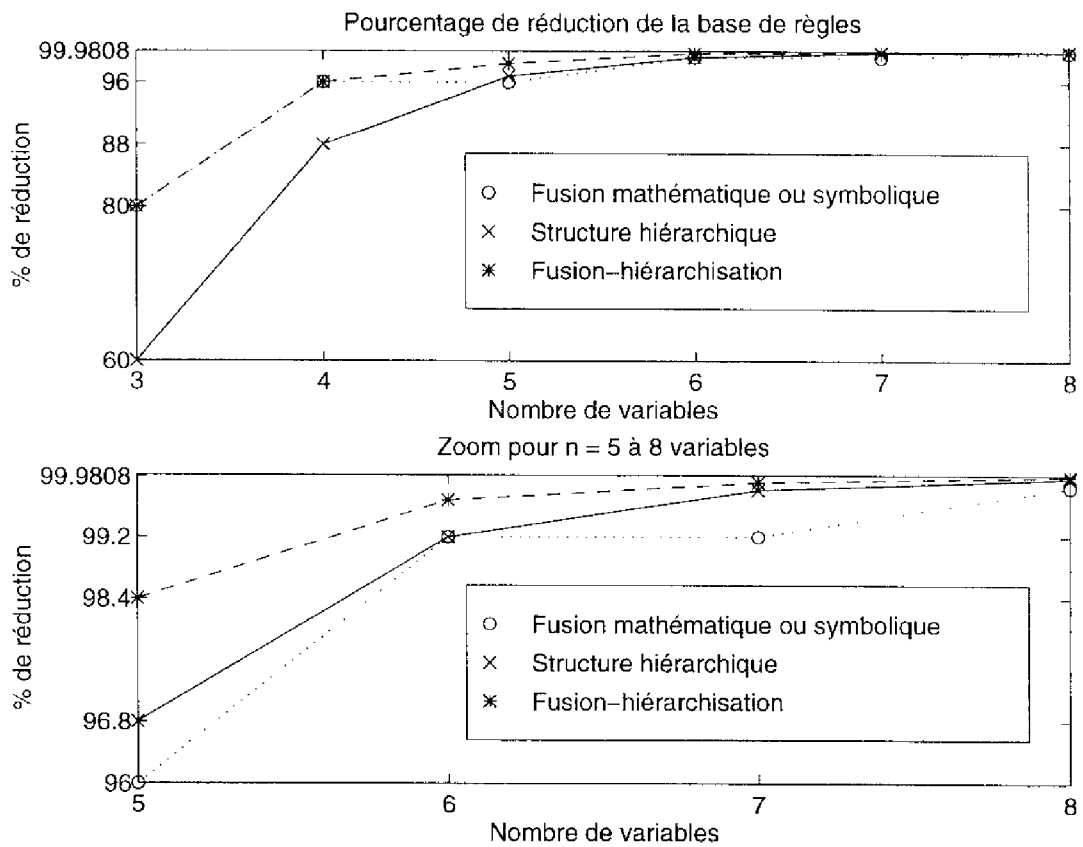


FIG. 3.18 – Pourcentage de réduction de la base de règles pour diverses méthodes de réduction ($m = 5$ labels linguistiques par variable).

Illustration sur le pendule inversé

- Fusion-hiérarchie

Une fusion mathématique de l'erreur en position et de sa variation $X_e = ce + d\Delta c$ combinée à une structure hiérarchique conduit au contrôleur flou représenté dans la figure 3.19. Le premier contrôleur flou C.F.1 calcule une première commande en fonction de X_e et de la position angulaire Φ . La base de règles correspondante est similaire à celle écrite pour le contrôleur flou basé sur une fusion mathématique seule (paragraphe 3.1.1). Quant au second contrôleur flou C.F.2, il affine la valeur de commande précédente en considérant une variable supplémentaire $\Delta\Phi$. Sa base de règles est obtenue en suivant le même raisonnement que celui qui a guidé l'écriture de la 3^{ème} base de règles du contrôleur flou basé sur la hiérarchie seule (paragraphe 3.2.2).

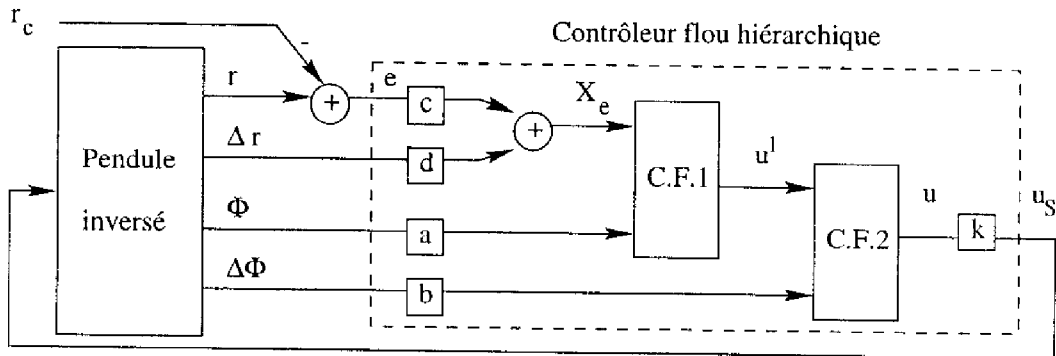


FIG. 3.19 – Contrôleur flou basé sur l'approche agrégation-hiérarchisation

Les bases de règles des deux contrôleurs flous C.F.1 et C.F.2 sont représentées dans la table 3.7.

$a\Phi$	N	Z	P
X_e	N	Z	P
	N	Z	P
	Z	P	
	P		

$C.F.1 \rightarrow u^1$

$b\Delta\Phi$	N	Z	P
u^1	N	Z	P
	N	Z	P
	Z	P	
	P		

$C.F.2 \rightarrow u$

TAB. 3.7 – Base de règles du contrôleur flou obtenu par agrégation-hiérarchisation

Dans le cas d'une fusion symbolique et non pas mathématique de e et Δe les règles du premier contrôleur flou C.F.1 s'écrivent :

- R_1 : Si $a\Phi$ est Négatif alors u est Négatif,
- R_2 : Si $a\Phi$ est Positif alors u est Positif,
- R_3 : Si $a\Phi$ est Zéro et $(ce, d\Delta e)$ est Négatif alors u est Négatif,
- R_4 : Si $a\Phi$ est Zéro et $(ce, d\Delta e)$ est Zéro alors u est Zéro,
- R_5 : Si $a\Phi$ est Zéro et $(ce, d\Delta e)$ est Positif alors u est Positif.

Les labels linguistiques *Négatif*, *Zéro* et *Positif* sont représentés par des ensembles flous bidimensionnels (voir figure 3.11).

La réponse temporelle, à une consigne en position de 30 cm, d'un contrôleur flou basé sur une *fusion* (mathématique ou symbolique) et une *hiérarchisation* est représentée dans la figure 3.20. La sortie des deux contrôleurs flous C.F.1 et C.F.2 y est aussi représentée.

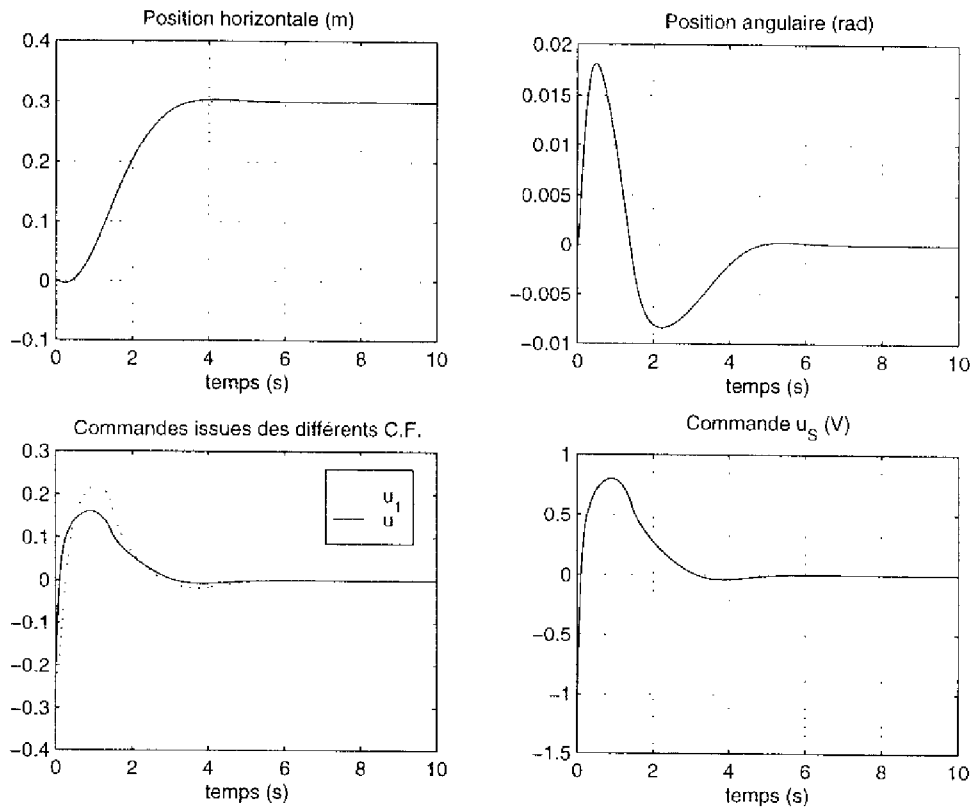


FIG. 3.20 – Réponse temporelle à une consigne r_c de 30 cm

- Fusion mathématique et symbolique

Dans le cas d'une fusion mathématique et symbolique des variables en entrée du contrôleur flou, celles-ci sont d'abord combinées mathématiquement comme dans la figure 3.7. Les règles de commande sont :

R_1 : Si (X_e, X_ϕ) est *Négatif* alors u est *Négatif*,

R_2 : Si (X_e, X_ϕ) est *Zéro* alors u est *Zéro*,

R_3 : Si (X_e, X_ϕ) est *Positif* alors u est *Positif*.

Les ensembles flous *Négatif*, *Zéro* et *Positif* sont des ensembles flous bidimensionnels construits à partir d'une triangulation de Delaunay d'un ensemble de points $\mathcal{P} = \{P_1, P_2, \dots, P_9\}$ régulièrement répartis sur l'univers de discours normalisé $[-1,1] \times [-1,1]$. Ils sont représentés dans la figure 3.21.

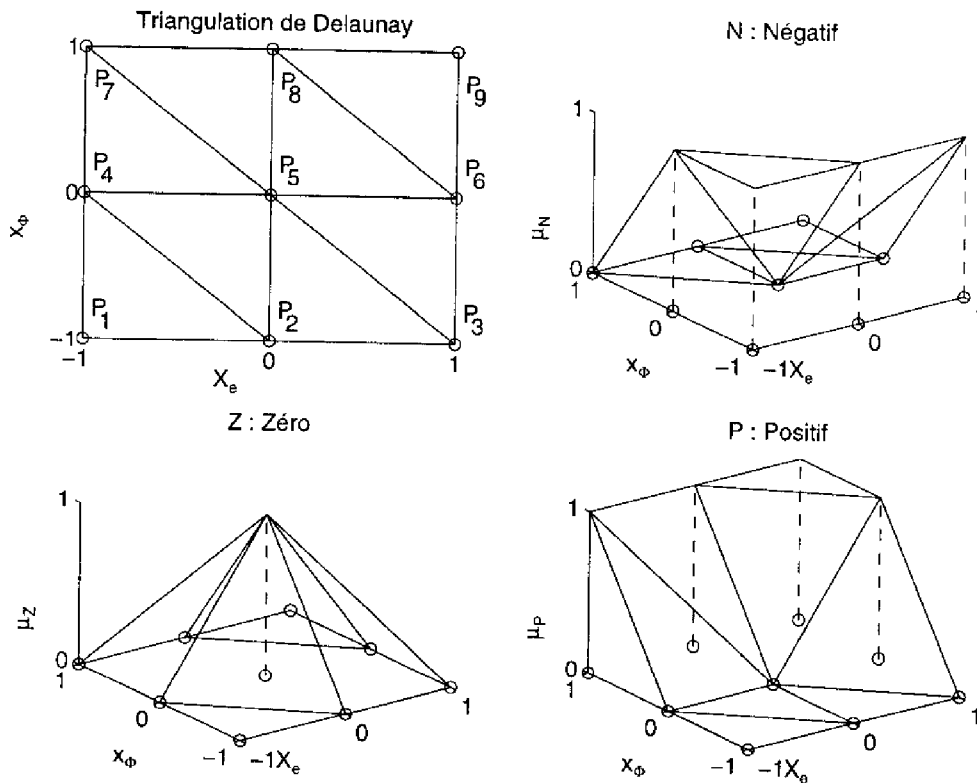


FIG. 3.21 – Les fonctions d'appartenance sont définies à partir d'une triangulation de Delaunay.

Les vecteurs modaux associés aux ensembles flous *Négatif*, *Zéro* et *Positif* sont respectivement

$$\mathcal{M}_N = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathcal{M}_Z = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{et} \quad \mathcal{M}_P = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \quad (3.37)$$

La réponse temporelle à une consigne en position de 30 cm est identique à celle obtenue avec le contrôleur flou basé sur l'approche par fusion-hiérarchisation (voir figure 3.20).

3.4 Conclusion

Nous avons présenté, dans ce chapitre, diverses techniques permettant de définir un contrôleur flou de complexité réduite. L'application de ces techniques au problème bien connu du pendule inversé atteste de leur efficacité en terme de réduction du nombre de règles.

L'utilisation de telles méthodes, réduction de la dimension du problème et structuration de la loi de commande, constitue un premier pas vers la synthèse d'un contrôleur flou de complexité réduite. Nous allons, maintenant, dans le chapitre 4, aborder la phase finale de mise au point du contrôleur flou.

Chapitre 4

Auto-réglage du contrôleur flou

Compte-tenu des nombreux paramètres à régler dans un contrôleur flou (facteurs d'échelle, fonctions d'appartenance), sa mise au point par une procédure de type essais-erreurs peut se révéler longue et fastidieuse.

L'application des méthodes de réduction présentées dans le chapitre précédent permet de définir un contrôleur flou de complexité réduite. Ainsi, le nombre de paramètres à régler se trouve diminué et des techniques d'optimisation peuvent être mises en œuvre afin de faciliter la phase de mise au point du contrôleur flou.

Ces techniques d'optimisation peuvent aller des méthodes éprouvées de la programmation mathématique à des concepts plus récents tels que les algorithmes génétiques. Nous nous sommes limités à la technique de descente du gradient, sa simplicité de mise en œuvre la rendant très attrayante.

La procédure d'auto-réglage du contrôleur flou est basée sur une structure de commande adaptative par modèle de référence [AW95]. Cette structure est présentée dans le paragraphe 4.1. Dans le paragraphe 4.2, nous détaillons le mécanisme d'ajustement des paramètres du contrôleur flou.

4.1 Apprentissage par modèle de référence

Le schéma de principe de l'auto-réglage d'un contrôleur flou par modèle de référence est représenté dans la figure 4.1.

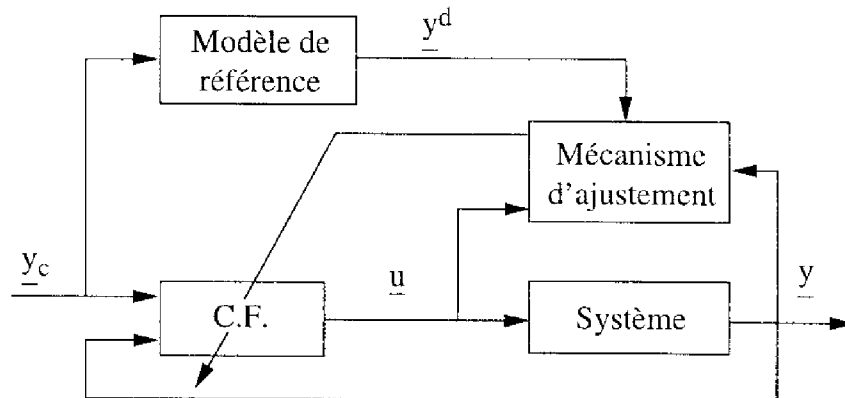


FIG. 4.1 – Schéma de principe de l'auto-réglage d'un contrôleur flou

On distingue deux boucles dans ce schéma : une boucle interne de régulation et une boucle externe permettant de modifier les paramètres du contrôleur flou. Cette boucle externe comporte un modèle de référence et un mécanisme d'ajustement. Le modèle de référence permet de spécifier le comportement désiré du processus. L'objectif du mécanisme d'ajustement est de régler les paramètres du contrôleur flou à travers la minimisation de l'erreur entre la sortie du modèle de référence et la sortie du processus.

Nous allons maintenant expliciter chacun de ces modules dans le cadre de notre étude.

Le système à commander

Le système est supposé sans retard, multivariable, avec p entrées et q sorties :

$$\underline{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_p \end{bmatrix} \quad \underline{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_q \end{bmatrix} \quad (4.1)$$

Le contrôleur flou

Dans un souci de minimisation du nombre de règles, des techniques de réduction présentées dans le chapitre 3 sont appliquées.

À titre d'exemple, une technique de réduction de la dimension (mathématique ou symbolique), combinée ou non à une hiérarchisation des variables fusionnées, conduit au contrôleur flou représenté dans la figure 4.2.

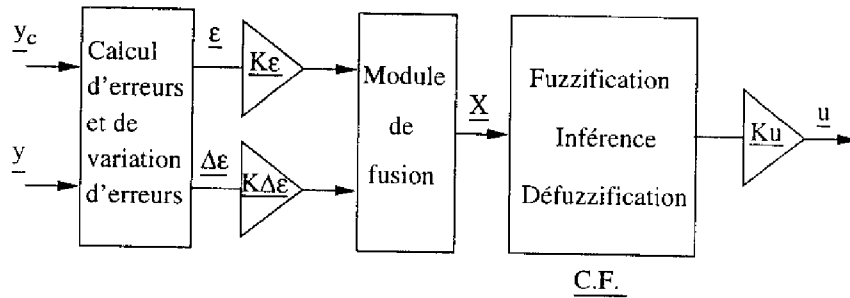


FIG. 4.2 - Structure de base du contrôleur flou

$\underline{\epsilon} = (\underline{y}_c - \underline{y})$ et $\underline{\Delta\epsilon}$ sont respectivement les vecteurs d'erreur et de variation d'erreur.

$$\underline{K\underline{\epsilon}} = \begin{bmatrix} K\epsilon_1 \\ \vdots \\ K\epsilon_q \end{bmatrix} \text{ et } \underline{K\underline{\Delta\epsilon}} = \begin{bmatrix} K\Delta\epsilon_1 \\ \vdots \\ K\Delta\epsilon_q \end{bmatrix} \text{ sont les vecteurs de gain en entrée.}$$

Le vecteur de gain en sortie est noté $\underline{Ku} = \begin{bmatrix} Ku_1 \\ \vdots \\ Ku_q \end{bmatrix}$.

$\underline{X} = \begin{bmatrix} X_1 \\ \vdots \\ X_q \end{bmatrix}$ est le vecteur contenant les variables issues du module de fusion.

Ainsi, dans le cas d'une fusion mathématique

$$X_i = K\epsilon_i \cdot \epsilon_i + K\Delta\epsilon_i \cdot \Delta\epsilon_i, \quad i = 1, \dots, q \quad (4.2)$$

et dans le cas d'une fusion symbolique

$$X_i = (K\epsilon_i \cdot \epsilon_i, K\Delta\epsilon_i \cdot \Delta\epsilon_i), \quad i = 1, \dots, q. \quad (4.3)$$

Le contrôleur flou (C.F.) est composé d'un ou de plusieurs contrôleurs flous (plusieurs dans le cas d'une structure de commande hiérarchique). Ces contrôleurs sont des systèmes flous de type Sugeno d'ordre zéro et ont au maximum deux entrées. Ces entrées peuvent être des variables issues du module de fusion, ou encore la variable de sortie d'un autre contrôleur flou. Ces contrôleurs flous étant représentés par une fonction mathématique simple et dérivable, une technique d'optimisation telle que la descente du gradient est envisageable (voir l'annexe A).

Le modèle de référence

Bien que le procédé puisse présenter de forts couplages entre ses entrées et ses sorties, les performances désirées pour chacune des sorties régulées peuvent être représentées par des modèles indépendants. Autrement dit, pour chacune des variables

régulées, un modèle de référence est déduit. Il peut s'agir, par exemple, d'un modèle mathématique précis ou d'un modèle flou. La seule contrainte pour ce modèle est qu'il traduise correctement les performances recherchées.

Le mécanisme d'ajustement

L'auto-réglage du contrôleur flou consiste à minimiser un critère quadratique

$$J = \sum_{t=t_{\text{début}}}^{t=t_{\text{fin}}} \left(\frac{1}{2} \underline{e}^T(t) \cdot Q \cdot \underline{e}(t) \right) = \sum_{t=t_{\text{début}}}^{t=t_{\text{fin}}} J(t). \quad (4.4)$$

avec $\underline{e}(t) = y(t) - \underline{y}^d(t)$ la différence entre la sortie réelle et la sortie désirée et Q une matrice diagonale de dimension (q,q) , définie non négative.

Les paramètres du contrôleur flou sont optimisés par la méthode du gradient, à chaque itération k , selon la formule

$$K(k+1) = K(k) - \eta \left(\sum_{t=t_{\text{début}}}^{t=t_{\text{fin}}} \frac{\partial J(t)}{\partial K} \right)_k \quad (4.5)$$

avec K le paramètre à régler et η le facteur de descente. L'algorithme s'achève lorsque la variation du critère n'est plus significative. Les calculs du gradient relatifs à un contrôleur flou de Sugeno d'ordre zéro sont présentés dans l'annexe A.

La mise au point finale du contrôleur flou est réalisée en deux étapes (voir la figure 4.3) :

1. Réglage des facteurs d'échelle,
2. Optimisation des fonctions d'appartenance.

Dans le paragraphe suivant, nous détaillons les différentes étapes dans l'ajustement des paramètres du contrôleur flou.

4.2 La procédure d'auto-réglage du contrôleur flou

4.2.1 Initialisation du contrôleur flou

Avant de procéder à l'auto-réglage du contrôleur flou, celui-ci est initialisé comme suit :

- Les gains en entrée sont équivalents à des facteurs d'échelle. Ils sont donc initialisés à partir d'une connaissance a priori de la valeur maximale de leurs entrées. L'erreur maximale d'un signal peut être déterminée à partir de sa valeur initiale et de ses valeurs de consigne. La valeur maximale de la variation de l'erreur d'un signal peut être obtenue par une simulation du système en boucle ouverte, ou encore, à partir du modèle de référence.

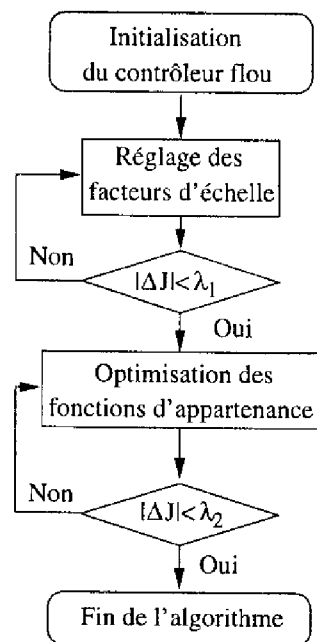


FIG. 4.3 – Organigramme de la procédure d'auto-réglage du contrôleur flou

- Les gains en sortie sont initialisés en fonction des limitations physiques des actionneurs et de façon à garantir la stabilité du système en boucle fermée.
- Les fonctions d'appartenance sont régulièrement espacées sur leur univers de discours normalisé (de dimension 1 ou 2) et forment une partition floue.
- La base de règles est initialisée à partir d'une connaissance experte sur le procédé ou encore à partir d'une analyse qualitative de celui-ci.

La procédure d'auto-réglage du contrôleur flou étant une procédure itérative où l'on engendre une suite de points convergeant vers un optimum **local** du critère, on comprendra aisément l'importance de cette étape préliminaire. Ainsi, si le contrôleur flou est correctement initialisé, le résultat obtenu est généralement satisfaisant. Dans le cas contraire, une solution consiste à redémarrer l'algorithme avec une initialisation différente.

4.2.2 Réglage des facteurs d'échelle

Les facteurs d'échelle ont une influence majeure sur les performances dynamiques du système bouclé; par conséquent, ce sont les premiers paramètres du contrôleur flou que nous adaptons. Cette phase de réglage permet d'approcher grossièrement mais surtout rapidement le modèle de référence.

4.2.3 Optimisation des fonctions d'appartenance

Cette deuxième étape, à travers l'optimisation des fonctions d'appartenance, permet un réglage plus fin du contrôleur flou. Les ensembles flous en entrée sont des triangles ou des ensembles flous bidimensionnels. Les ensembles flous en sortie sont des singletons.

Fonctions d'appartenance triangulaires

Beaucoup d'articles relatifs aux systèmes neuro-flous traitent de l'apprentissage d'un système flou à partir d'une technique d'optimisation basée sur la méthode du gradient. Les systèmes flous sont alors perçus comme de simple réseaux de neurones.

Dans [NHN91], le système flou considéré est un système flou de type Sugeno d'ordre zéro, l'opérateur de conjonction est le *produit* et les ensembles flous en entrée sont caractérisés par des fonctions d'appartenance triangulaires symétriques. F. Guely et P. Siarry [GS94] ont étendu cette étude à des systèmes flous de type Sugeno d'ordre un (les conclusions des règles sont des fonctions linéaires des variables d'entrée), ayant le *min* ou le *produit* comme opérateur de conjonction et des ensembles flous en entrée caractérisés par des fonctions d'appartenance triangulaires asymétriques.

Dans ces deux articles [NIIN91, GS94], les ensembles flous en entrée sont paramétrés par rapport à la règle dans laquelle ils apparaissent. Ainsi, un label linguistique (d'une même variable) intervenant dans différentes règles est caractérisé par des ensembles flous initialement (c'est à dire avant la procédure d'optimisation) égaux mais ces ensembles étant adaptés localement dans chacune des règles comme s'ils étaient différents, cela entraîne une perte au niveau de l'interprétation sémantique des règles. En effet, selon la règle considérée, une valeur linguistique d'une même variable peut être représentée par des ensembles flous très différents. De plus, la propriété de complétude de la base de règles n'est pas assurée car il est possible que les ensembles flous en entrée ne recouvrent pas l'univers de discours sur lequel ils sont définis.

Dans [BJPB93], les ensembles flous ne sont plus paramétrés par rapport aux règles dans lesquelles ils apparaissent mais par rapport à leur position sur leur univers de discours. Par conséquent, un label linguistique d'une même variable intervenant dans différentes règles sera caractérisé par le même ensemble flou, que ce soit avant ou après la procédure d'optimisation. Cependant, ici aussi, la complétude de la base de règles n'est pas assurée.

Afin de ne pas dégrader la bonne lisibilité des règles, nous choisissons donc de paramétrer les ensembles flous en entrée par rapport à la position qu'ils occupent sur leur univers de discours. Par conséquent, les fonctions d'appartenance associées à une même variable d'entrée sont ordonnées sur leur univers de discours

$$m_1 < \dots < m_{i-1} < m_i < m_{i+1} < \dots < m_m \quad (4.6)$$

avec m_i la valeur modale du $i^{\text{ème}}$ ensemble flou F^i . On veillera, lors de la procédure d'optimisation, à ce que cet ordonnancement soit toujours respecté de sorte que, par exemple, le label linguistique *Petit* ne soit pas représenté par un ensemble flou ayant une valeur modale supérieure à celle associée au label *Grand*.

Quant à la complétude de la base de règles, elle est garantie en optimisant uniquement les valeurs modales des ensembles flous en entrée. Le support d'un ensemble flou en entrée est alors déterminé à partir des valeurs modales des ensembles flous qui lui sont adjacents (voir la figure 4.4).

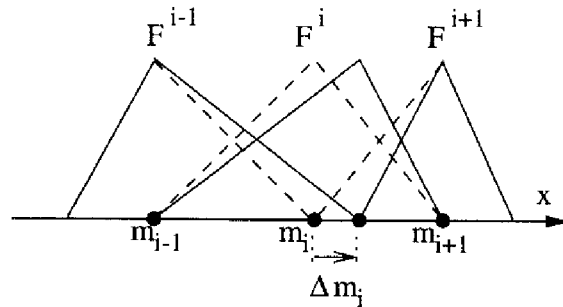


FIG. 4.4 – Les fonctions d'appartenance forment une partition floue

Si la procédure d'optimisation conduit à une dénormalisation de l'univers de discours, celui-ci est de nouveau normalisé à travers une réactualisation des facteurs d'échelle.

Fonctions d'appartenance bidimensionnelles

Les ensembles flous bidimensionnels sont construits à partir d'une triangulation de Delaunay d'un ensemble de points $\mathcal{P}_n = \{P_1, \dots, P_n\}$ initialement régulièrement répartis sur l'univers de discours normalisé $[-1, 1] \times [-1, 1]$. Des vecteurs modaux de dimension n sont alors associés à chacun des ensembles flous F^i définis sur cet univers de discours. Ces ensembles flous forment une partition floue.

L'optimisation des fonctions d'appartenance bidimensionnelles consiste à optimiser les coordonnées des points de \mathcal{P}_n . Une illustration est donnée dans la figure 4.5.

La figure 4.5 permet de mettre en évidence l'incomplétude de la base de règles suite à la modification du point P_3 (par exemple, pour le point d'entrée $p = (1, -1)$ aucune règle n'est activée).

Ce problème peut être résolu par l'introduction de points

$$\mathcal{P}_{\text{extra}} = \{P_{n+1}, \dots, P_{n+e}\} \quad (4.7)$$

formant un cadre englobant la précédente triangulation. Si les points de ce cadre ne sont pas optimisés, le problème de l'incomplétude de la base de règles est alors résolu.

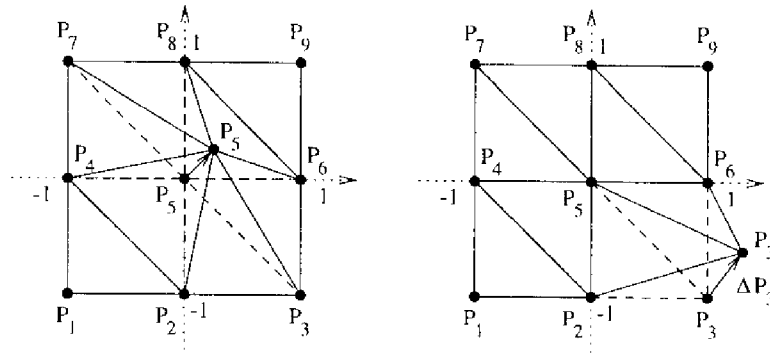


FIG. 4.5 – L'optimisation d'un point frontière $\{P_1, \dots, P_4, P_6, \dots, P_9\}$ peut conduire à une incomplétude de la base de règles.

Dans le cadre de notre étude, les points de $\mathcal{P}_{\text{extra}}$ sont obtenus par extrapolation des ensembles flous à partir de l'ensemble de points précédent \mathcal{P}_n . Le nombre de point extrapolés dépend du nombre m d'ensembles flous :

$$e = 2.(m + 1) \quad (4.8)$$

Une nouvelle triangulation de Delaunay de l'ensemble de points $\mathcal{P} = \mathcal{P}_n + \mathcal{P}_{\text{extra}}$ est alors calculée. Ce calcul n'est réalisé qu'une seule fois et est obtenu sous la forme d'une matrice de triangulation $\mathcal{T}_{\mathcal{P}}$. Chaque colonne de $\mathcal{T}_{\mathcal{P}}$ représente un triangle et contient les indices des points formant ce triangle.

Une illustration est donnée dans la figure 4.6 avec

$$\mathcal{P}_n = \begin{bmatrix} P_1 \\ \vdots \\ P_9 \end{bmatrix}, \quad \mathcal{T}_{\mathcal{P}_n} = \begin{bmatrix} 1 & 2 & 2 & 3 & 4 & 5 & 5 & 6 \\ 2 & 4 & 3 & 5 & 5 & 7 & 6 & 8 \\ 4 & 5 & 5 & 6 & 7 & 8 & 8 & 9 \end{bmatrix} \quad (4.9)$$

et trois ensembles flous N, Z, et P. Les vecteurs modaux initiaux associés à ces ensembles flous sont respectivement

$$\begin{aligned} \mathcal{M}_N &= [1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0]^T \\ \mathcal{M}_Z &= [0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0]^T \\ \mathcal{M}_P &= [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1]^T \end{aligned} \quad (4.10)$$

L'extrapolation des ensembles flous N, Z et P permet de construire

$$\mathcal{P}_{\text{extra}} = \begin{bmatrix} P_{10} \\ \vdots \\ P_{17} \end{bmatrix} \quad (4.11)$$

et les vecteurs modaux associés à N, Z et P sont modifiés en conséquence :

$$\begin{aligned} \mathcal{M}_N &= [1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0]^T \\ \mathcal{M}_Z &= [0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0]^T \\ \mathcal{M}_P &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1]^T \end{aligned} \quad (4.12)$$

La nouvelle matrice de triangulation de l'ensemble de points $\mathcal{P} = \mathcal{P}_n + \mathcal{P}_{\text{extra}}$ est

$$\mathcal{T}_{\mathcal{P}} = \left[\mathcal{T}_{\mathcal{P}_n}, \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 3 & 3 & 3 & 4 & 6 & 7 & 7 & 7 & 8 & 8 & 9 \\ 2 & 13 & 4 & 10 & 3 & 10 & 11 & 6 & 7 & 9 & 13 & 14 & 8 & 15 & 9 & 12 \\ 16 & 16 & 13 & 16 & 10 & 11 & 12 & 12 & 13 & 12 & 14 & 15 & 15 & 17 & 17 & 17 \end{bmatrix} \right] \quad (4.13)$$

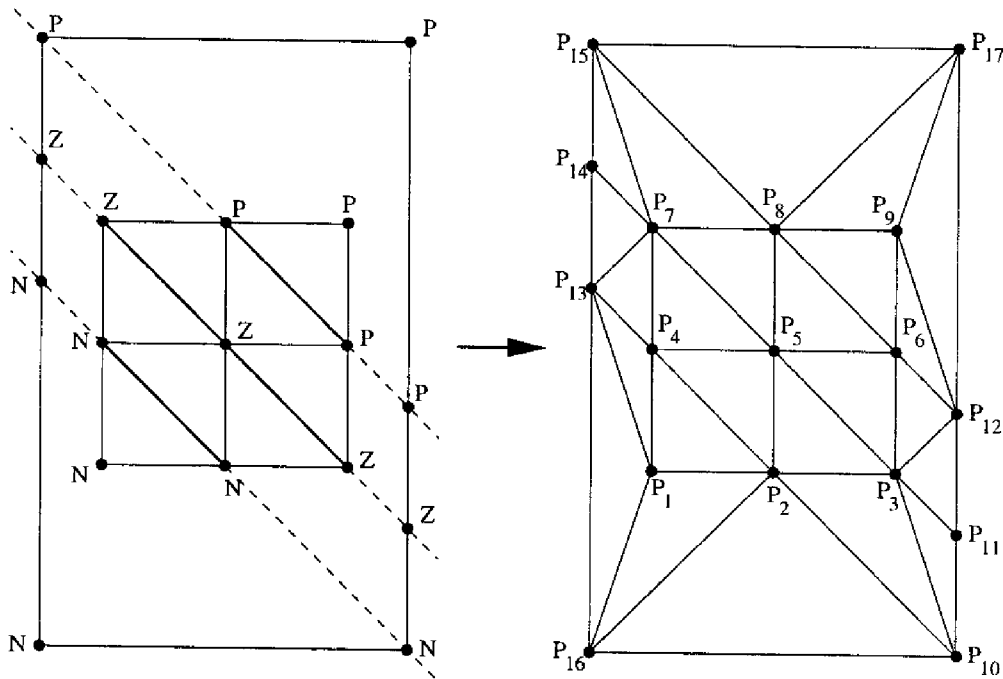


FIG. 4.6 – Extrapolation des ensembles flous bidimensionnels et construction d'une nouvelle triangulation de Delaunay.

Pour chaque point P_i de \mathcal{P} il ne correspond qu'un et un seul label linguistique. Ces labels linguistiques sont caractérisés par des ensembles flous de dimension 2: $\{F^1, \dots, F^i, \dots, F^m\}$. Le vecteur modal associé à chaque ensemble flou F^i est un

vecteur de dimension $(n + e)$:

$$\mathcal{M}_{F^i} = \begin{bmatrix} \mu_{F_i}(P_1) \\ \vdots \\ \mu_{F_i}(P_n) \\ \mu_{F_i}(P_{n+1}) \\ \vdots \\ \mu_{F_i}(P_{n+e}) \end{bmatrix} \quad (4.14)$$

Ces ensembles flous forment une partition floue

$$\sum_{i=1}^m \mathcal{M}_{F^i} = 1. \quad (4.15)$$

Notons que la matrice de triangulation de Delaunay \mathcal{T}_P ainsi que les vecteurs modaux sont fixes. Seules les coordonnées des points de \mathcal{P}_n sont optimisées. L'ensemble de points $\mathcal{P}_{\text{extra}}$ est alors automatiquement réactualisé par extrapolation des ensembles flous.

Une illustration est donnée dans la figure 4.7.

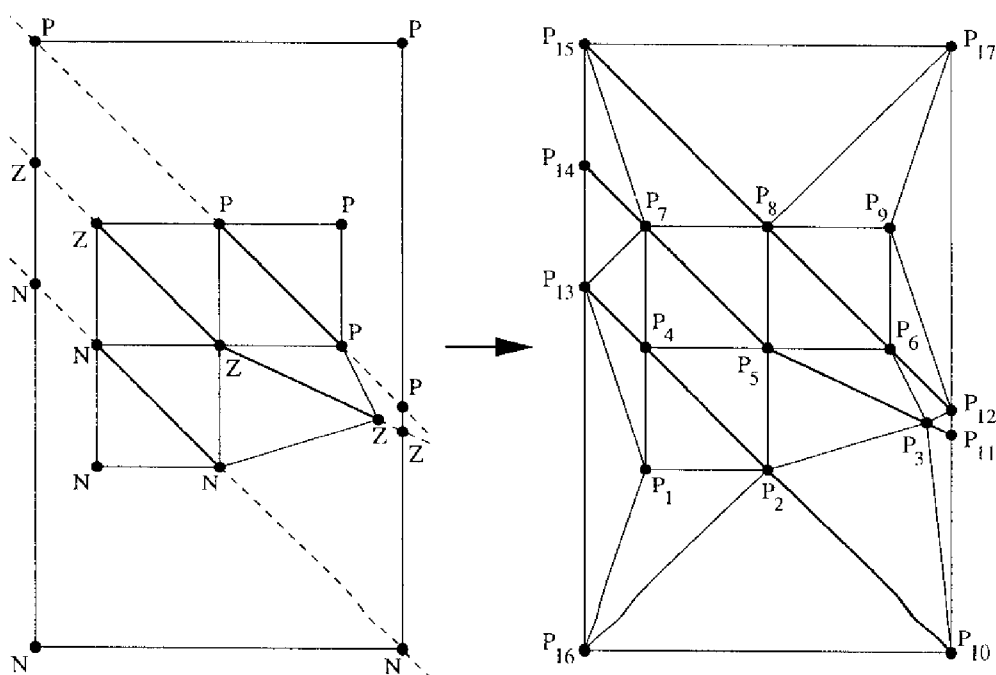


FIG. 4.7 -- Extrapolation des ensembles flous bidimensionnels et triangulation de Delaunay associée.

4.3 Conclusion

Dans ce chapitre, nous nous sommes penchés sur la phase finale de mise au point d'un contrôleur flou, à travers la poursuite d'une trajectoire de référence.

Les systèmes flous considérés sont des systèmes flous de type Takagi-Sugeno d'ordre zéro, l'opérateur de conjonction est le produit et les ensembles flous en entrée sont, soit des triangles, soit des ensembles flous bidimensionnels construits à partir d'une triangulation de Delaunay.

La méthode choisie pour régler les différents paramètres du contrôleur flou est la méthode du gradient.

La mise au point du contrôleur flou est réalisée en deux étapes :

1. réglage des gains,
2. optimisation des fonctions d'appartenance.

Le problème de l'optimisation d'ensembles flous triangulaires a été étendu à des ensembles flous bidimensionnels construits à partir d'une triangulation de Delaunay.

Dans le chapitre suivant, nous rappelons les différentes étapes de la méthodologie proposée.



Chapitre 5

Conclusion de la deuxième partie

Dans cette partie, nous nous sommes attachés, au problème de l'explosion combinatoire du nombre de règles d'un contrôleur flou.

Cette partie traite donc des techniques de réduction de la complexité d'une base de règles.

Nous avons, dans un premier temps, cherché à diminuer le nombre de règles d'une base donnée. Cependant, dans la pratique, on ne dispose généralement pas de cette base de règles et la synthèse d'un contrôleur flou est à réaliser. Plutôt que de synthétiser un contrôleur flou puis de réduire son nombre de règles, une démarche plus logique consiste à concevoir directement un contrôleur flou de complexité réduite. Le nombre de paramètres du contrôleur flou est alors considérablement diminué et sa synthèse s'en trouve d'autant plus facilitée. Néanmoins, malgré une réduction importante du nombre de paramètres, la mise au point finale du contrôleur flou reste une étape délicate et peut en effrayer plus d'un. C'est pourquoi, nous nous sommes intéressés à la technique d'optimisation, très simple, du gradient.

La méthodologie proposée est donc la suivante :

1. réduction de la dimension du problème de commande,
2. structuration de la loi de commande,
3. mise au point du contrôleur flou à l'aide d'une technique du gradient.

Il nous reste maintenant à valider cette méthodologie sur quelques applications.



Troisième partie

Applications



Chapitre 1

Application à un mélangeur

Nous illustrons la précédente méthodologie sur un exemple académique : un mélangeur multivariable à deux entrées et deux sorties.

Différentes structures de commande floue sont ici définies. Elles sont issues des méthodes de réduction suivantes :

- fusion mathématique et décentralisation [LT96b],
- fusion mathématique et hiérarchisation [LT97a, LTM],
- fusion symbolique et hiérarchisation [LT97c].

Les stratégies de commande floue sont comparées à une commande optimale et leur robustesse est testée expérimentalement.

1.1 Présentation du système

Le procédé est extrait de [JTV89]. En vue de pouvoir disposer d'un volume constant d'un mélange de concentration désirée, un mélangeur (représenté dans la figure 1.1) est constitué d'une cuve cylindrique de section $S = 1 \text{ m}^2$ alimentée par deux fluides de concentrations respectives C_1 et C_2 . Les débits d'entrée de ces fluides sont notés Q_1 et Q_2 , le débit de sortie Q_S , la concentration du fluide à l'intérieur du mélangeur C et le niveau du fluide à l'intérieur du mélangeur H .

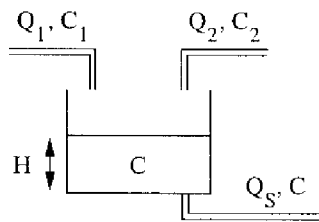


FIG. 1.1 - Schéma du mélangeur

Les variables régulées sont la hauteur et la concentration à l'intérieur du mélangeur et les actions de commande sont les deux débits d'entrée.

Afin de faciliter la modélisation du processus, retenons les hypothèses suivantes :

- le mélange des fluides d'entrée est instantané,
- le fluide de sortie s'écoulant par gravité, le débit de sortie est donné par la relation $Q_S = k \cdot \sqrt{H}$ où k est une constante ($k = 20$),

les concentrations C_1 et C_2 des deux fluides d'entrée sont constantes ($C_1 = 1$ mole/l et $C_2 = 2$ moles/l).

Le procédé est alors décrit par deux équations non linéaires :

$$\frac{dH}{dt} = \frac{Q_1}{S} + \frac{Q_2}{S} - \frac{k}{S} \sqrt{H} \quad (1.1)$$

$$\frac{dC}{dt} = \frac{1}{H} \left(\frac{Q_1}{S} \cdot (C_1 - C) + \frac{Q_2}{S} \cdot (C_2 - C) \right) \quad (1.2)$$

Le modèle linéarisé du mélangeur est obtenu en supposant que les débits d'entrée varient faiblement autour de leur valeur nominale, soient :

$$Q_1 = Q_{10} + q_1 \quad Q_2 = Q_{20} + q_2 \quad H = H_0 + h \quad C = C_0 + c$$

q_1 , q_2 , h et c , sont respectivement, les variations relatives de Q_1 , Q_2 , H et C autour de leur valeur nominale respective Q_{10} , Q_{20} , H_0 et C_0 .

Le modèle dans l'espace d'état est alors défini par

$$\begin{aligned}\dot{\underline{x}} &= \begin{bmatrix} -1 & 0 \\ \frac{2.T}{S} & -1 \\ 0 & T \end{bmatrix} \cdot \underline{x} + \begin{bmatrix} \frac{1}{S} & \frac{1}{S} \\ \frac{C_1 - C_0}{S.H_0} & \frac{C_2 - C_0}{S.H_0} \end{bmatrix} \cdot \underline{e} \\ \underline{y} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \underline{x}\end{aligned}\quad (1.3)$$

avec

$$\underline{\dot{x}} = \begin{bmatrix} h \\ c \end{bmatrix} \quad \underline{e} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \quad T = \frac{S.H_0}{Q_{10} + Q_{20}}$$

La matrice de transfert est définie par

$$\begin{bmatrix} h(p) \\ c(p) \end{bmatrix} = \begin{bmatrix} \frac{K_{11}}{1 + T_{11}.p} & \frac{K_{12}}{1 + T_{12}.p} \\ \frac{K_{21}}{1 + T_{21}.p} & \frac{K_{22}}{1 + T_{22}.p} \end{bmatrix} \cdot \begin{bmatrix} q_1(p) \\ q_2(p) \end{bmatrix}\quad (1.4)$$

avec

$$\begin{aligned}K_{11} = K_{12} = \frac{2.T}{S} \quad K_{21} = \frac{(C_1 - C_0).T}{S.H_0} \quad K_{22} = \frac{(C_2 - C_0).T}{S.H_0} \\ T_{11} = T_{12} = 2.T = 2.T_{21} = 2.T_{22}\end{aligned}$$

1.2 Présentation du cahier des charges

Les objectifs de commande sont la compensation des couplages, un gain statique unitaire et une dynamique en boucle fermée deux fois plus rapide que celle en boucle ouverte.

Dans [JTV89] une commande optimale est calculée autour du point de fonctionnement noté PDF1 défini par

$$H_0 = 1 \text{ m} \quad C_0 = 1.25 \text{ moles/l} \quad Q_{10} = 15 \text{ l/s} \quad Q_{20} = 5 \text{ l/s} \quad (1.5)$$

Les stratégies de commande floue sont donc mises au point autour de ce même point de fonctionnement puis comparées à la commande optimale proposée par Jaume et al.

Nous définissons donc deux trajectoires de référence indépendantes, représentées dans la figure 1.2, l'une pour la hauteur, l'autre pour la concentration :

$$h_d(p) = \frac{K_h}{1 + T_h.p} \cdot h_c(p), \quad c_d(p) = \frac{K_c}{1 + T_c.p} \cdot c_c(p) \quad (1.6)$$

avec $h_c(p)$ et $c_c(p)$ les niveaux de consigne en hauteur et en concentration. $K_h = K_c = 1$ les gains statiques, $T_h = 50s$ et $T_c = 25s$ les constantes de temps pour imposer la dynamique en boucle fermée.

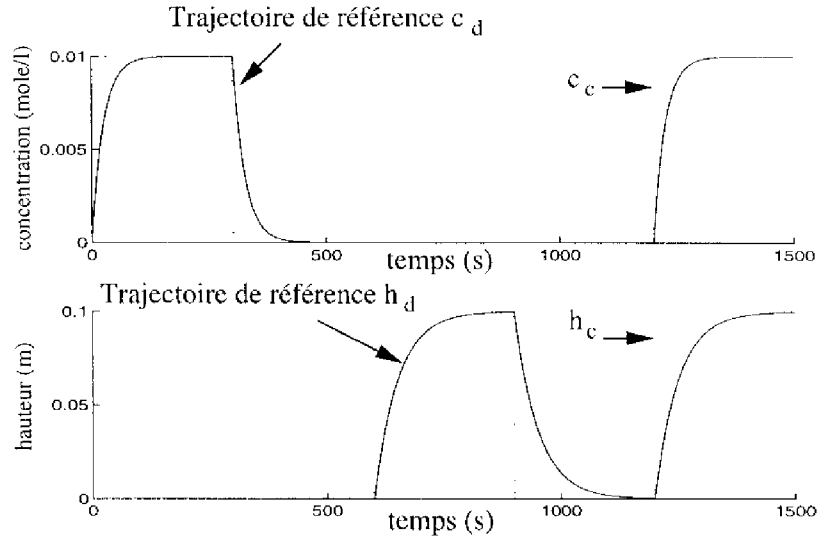


FIG. 1.2 – Trajectoires de référence pour la hauteur et la concentration.

La mise au point des contrôleurs flous est réalisée en minimisant le critère quadratique :

$$C = \sum_{t=t_{\text{début}}}^{t=t_{\text{fin}}} \left(\frac{1}{2} q_{11} [h(t) - h_d(t)]^2 + \frac{1}{2} q_{22} [c(t) - c_d(t)]^2 \right) \quad (1.7)$$

Les critères de comparaison entre la commande optimale de [JTV89] et les commandes floues portent :

- sur la dynamique des sorties et les termes de couplage :

critère en fonction de l'erreur en hauteur $\epsilon_h = h_c - h$

$$J(\epsilon_h) = \int_0^t q_{11} \cdot \epsilon_h^2 \cdot dt \quad (1.8)$$

- critère en fonction de l'erreur en concentration $\epsilon_c = c_c - c$

$$J(\epsilon_c) = \int_0^t q_{22} \cdot \epsilon_c^2 \cdot dt \quad (1.9)$$

- sur les coûts de commande

$$J(q_1, q_2) = \int_0^t (r_{11} \cdot q_1^2 + r_{22} \cdot q_2^2) \cdot dt \quad (1.10)$$

- sur le critère de performance global J

$$J = \int_0^t (q_{11} \cdot c_h^2 + q_{22} \cdot c_c^2 + r_{11} \cdot q_1^2 + r_{22} \cdot q_2^2) \cdot dt \quad (1.11)$$

La loi de commande optimale de [JTV89] assure la minimisation du critère J représenté dans l'équation (1.11). Les valeurs numériques des coefficients intervenant dans ce critère sont choisies de sorte que chacun des termes le composant ait une contribution semblable :

$$q_{11} = 100 \quad q_{22} = 70 \quad r_{11} = 0.45 \quad r_{22} = 4$$

La robustesse des lois de commande floue, optimisées autour du premier point de fonctionnement (1.5), sera évaluée expérimentalement autour d'un second point de fonctionnement noté PDF2 et défini par

$$H_0 = 2 \text{ m} \quad C_0 = 1.25 \text{ moles/l} \quad Q_{10} = 21.21 \text{ l/s} \quad Q_{20} = 7.07 \text{ l/s} \quad (1.12)$$

La matrice de transfert autour de ce point est :

$$\begin{bmatrix} h(p) \\ c(p) \end{bmatrix} = \begin{bmatrix} \frac{K'_{11}}{1 + T'_{11} \cdot p} & \frac{K'_{12}}{1 + T'_{12} \cdot p} \\ \frac{K'_{21}}{1 + T'_{21} \cdot p} & \frac{K'_{22}}{1 + T'_{22} \cdot p} \end{bmatrix} \cdot \begin{bmatrix} q_1(p) \\ q_2(p) \end{bmatrix} \quad (1.13)$$

avec

$$K'_{11} = K'_{12} = 1,4 \cdot K_{12} \quad K'_{21} = 0,7 \cdot K_{21} \quad K'_{22} = 0,7 \cdot K_{22}$$

$$T'_{11} = T'_{12} = 1,4 \cdot T_{12} \quad T'_{21} = T'_{22} = 1,4 \cdot T_{22}$$

La dynamique du système en boucle ouverte est donc plus lente et les gains statiques ont changé.

Remarques

Pour simplifier la mise en œuvre de l'algorithme d'auto-réglage du contrôleur flou, le jacobien du processus régulé $\begin{pmatrix} \frac{\partial h(t)}{\partial q_1(t)} & \frac{\partial h(t)}{\partial q_2(t)} \\ \frac{\partial c(t)}{\partial q_1(t)} & \frac{\partial c(t)}{\partial q_2(t)} \end{pmatrix}$ est approximé par son signe $\begin{pmatrix} +1 & +1 \\ -1 & +1 \end{pmatrix}$, approximation couramment utilisé en commande neuronale.

Dans tout ce qui suit, nous utilisons seulement le modèle linéarisé (1.3) du mélangeur pour la mise au point des contrôleurs flous autour de PDF1 et, ce, afin d'accélérer les temps de calcul. Les équations non linéaires (1.1) et (1.2) du mélangeur sont utilisées afin d'évaluer les résultats des différentes stratégies de commande floue.

1.3 Analyse qualitative du mélangeur

Le modèle qualitatif de C. Raymond [RBT94] est ici directement utilisé et est représenté dans la figure 1.3.

Commandes		Sorties	
q_1	q_2	h	c
+	0	++	-
++	+	+++	0
-	--	---	0
0	+	++	++
-	+	0	+++
0	-	+	+
+	-	0	---

Légende

+ : augmentation de la variable
 - : diminution de la variable
 0 : variation nulle ou presque nulle de la variable

Le nombre de signes représente l'amplitude de la variation

FIG. 1.3 – Modèle qualitatif du mélangeur

Ce modèle qualitatif est obtenu à partir de plusieurs simulations du procédé en boucle ouverte, lesquelles consistent à appliquer différents couples d'entrée (q_1, q_2) de type échelon et à observer la dynamique des sorties h et c correspondantes. Deux exemples de simulation en boucle ouverte sont donnés dans la figure 1.4.

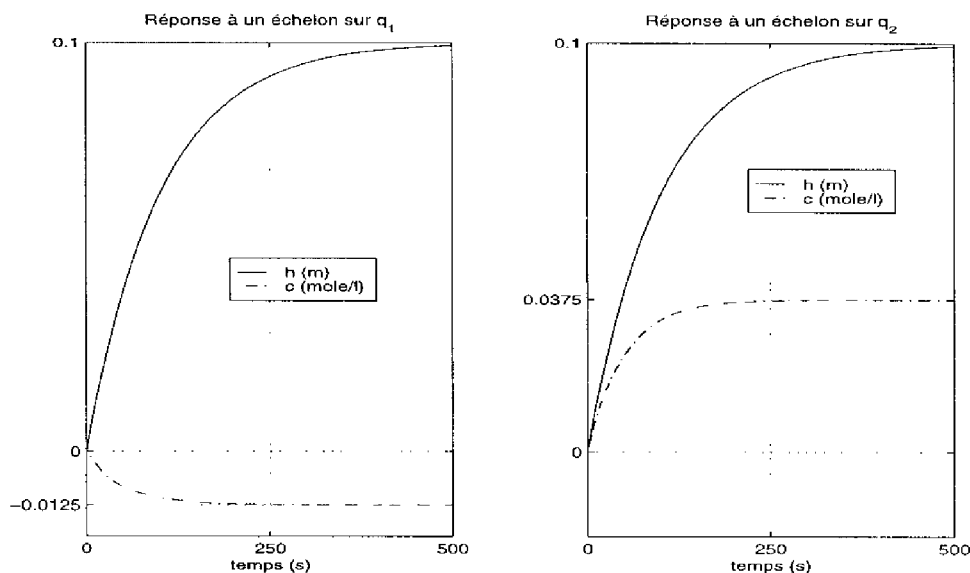


FIG. 1.4 – Essais en boucle ouverte autour de PDF1

Dans l'hypothèse d'une approche boîte noire, on peut déduire de ces essais que le

système est sans retard et que la dynamique de chaque sortie par rapport à chaque entrée est de type premier ordre. Par conséquent, un contrôleur flou de type P.I. sera suffisant pour réguler le processus.

1.4 Réduction de la dimension du problème de commande

Une fusion mathématique de l'erreur et de la variation de l'erreur, correspondant à la hauteur et à la concentration, conduit aux deux variables suivantes :

$$X_h = K_{\epsilon_h} \cdot \epsilon_h + K_{\Delta\epsilon_h} \cdot \Delta\epsilon_h \quad (1.14)$$

$$X_c = K_{\epsilon_c} \cdot \epsilon_c + K_{\Delta\epsilon_c} \cdot \Delta\epsilon_c \quad (1.15)$$

X_h et X_c représentent respectivement, l'"état général" de la hauteur et de la concentration du fluide à l'intérieur du mélangeur.

Dans le cas d'une fusion symbolique, l'"état général" de la hauteur et celui de la concentration du fluide à l'intérieur du mélangeur sont respectivement donnés par :

$$X_h = (K_{\epsilon_h} \cdot \epsilon_h, K_{\Delta\epsilon_h} \cdot \Delta\epsilon_h) \quad (1.16)$$

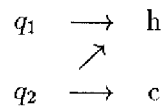
$$X_c = (K_{\epsilon_c} \cdot \epsilon_c, K_{\Delta\epsilon_c} \cdot \Delta\epsilon_c) \quad (1.17)$$

1.5 Structuration de la loi de commande

L'analyse des lignes grisées du modèle qualitatif (voir la figure 1.3) permet de mettre en évidence les couplages reliant les entrées et les sorties du procédé :

- q_1 et q_2 agissent de façon équivalente sur h ,
- l'action de q_2 sur c est plus importante que celle de q_1 sur c .

Le graphe causal des couplages forts dans le procédé est donc le suivant :



TAB. 1.1 – Graphe causal des couplages forts

L'action de q_1 sur c est ici négligée compte tenu du fort couplage entre q_2 et c .

Si l'on néglige aussi l'influence de q_2 sur h , une structure de commande décentralisée, représentée dans la figure 1.5, peut être adoptée : les sorties h et c sont alors respectivement asservies par les commandes q_1 et q_2 .

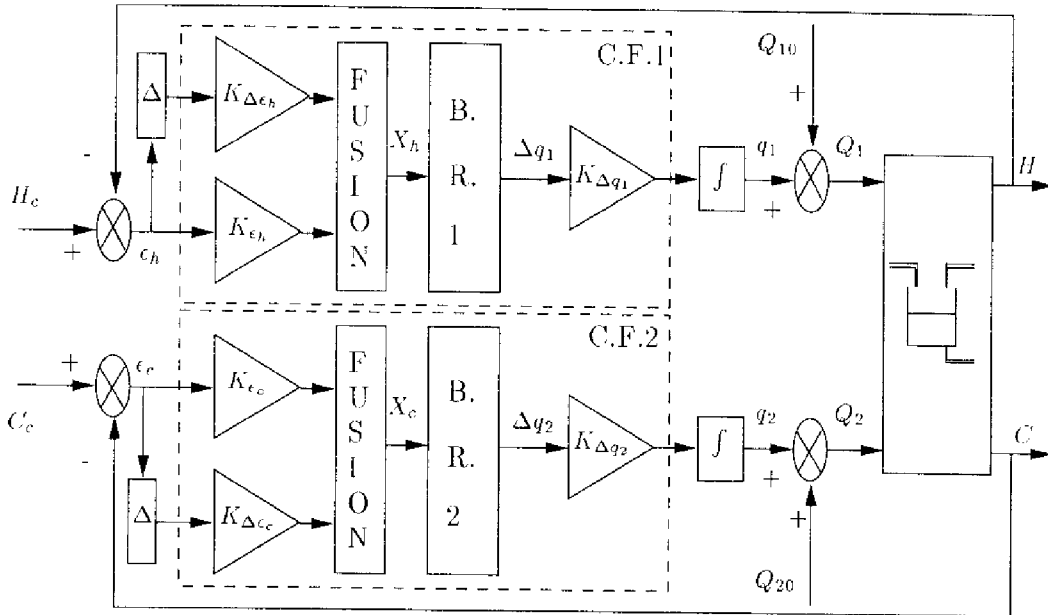


FIG. 1.5 - Structure de commande décentralisée

L'initialisation des facteurs d'échelle est la suivante :

$$K_{\epsilon_h} = 10 \quad K_{\Delta\epsilon_h} = 500 \quad K_{\Delta q_1} = 0.1 \quad K_{\epsilon_c} = 100 \quad K_{\Delta\epsilon_c} = 2550 \quad K_{\Delta q_2} = 0.1 \quad (1.18)$$

Quant aux ensembles flous, ils sont régulièrement répartis sur leur univers de discours.

Les bases de règles B.R.1 et B.R.2, associées respectivement aux contrôleurs flous C.F.1 et C.F.2 de la figure 1.5. sont initialisées comme suit :

$X_i(i = h, c)$	NM	NP	ZE	PP	PM
$\Delta q_i(i = 1, 2)$	NM	NP	ZE	PP	PM

TAB. 1.2 - B.R.i (i=1,2)

Dans la figure 1.5 nous avons négligé l'action de q_2 sur h . La prise en compte de tous les couplages forts représentés dans la table 1.1 conduit à une structure de commande hiérarchique représentée dans la figure 1.6.

On considère dans un premier temps l'état de la concentration du fluide à l'intérieur du mélangeur et une première action de commande q_2 est calculée par C.F.1. Cette commande permet de réguler c mais agit également sur h . De façon à compenser l'action de q_2 sur h , plutôt que de considérer uniquement l'état de la hauteur

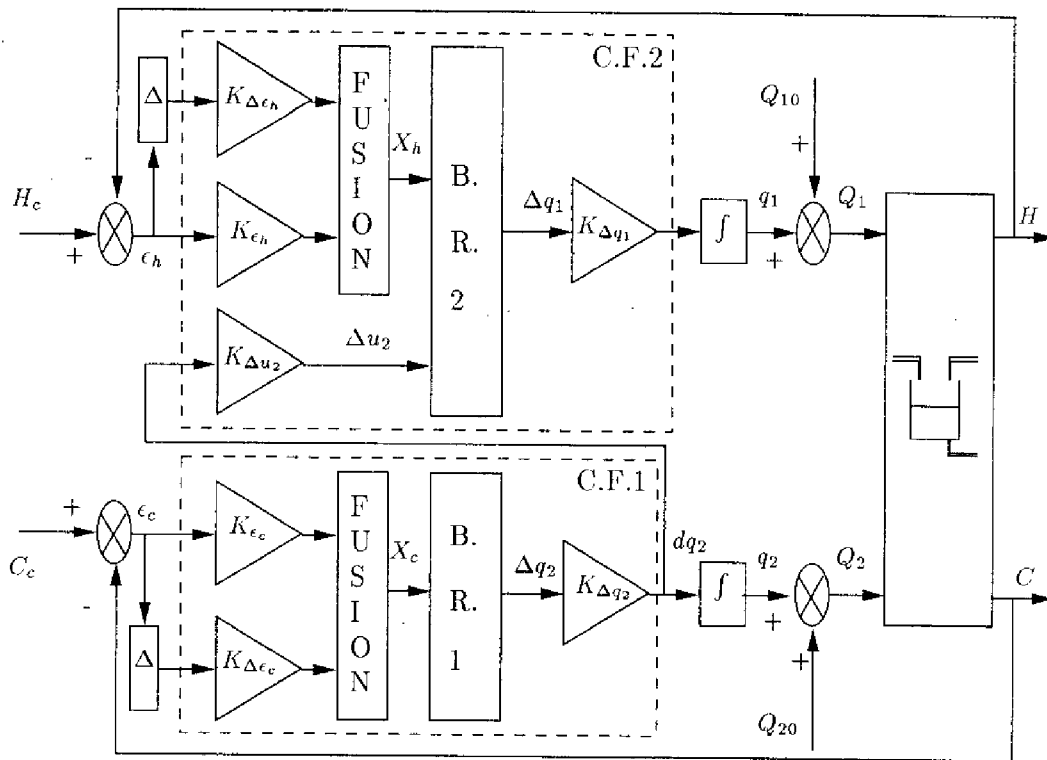


FIG. 1.6 – Structure de commande hiérarchique

pour calculer le débit q_1 , on tient aussi compte du débit q_2 calculé par le premier contrôleur flou. La sortie de C.F.1 est donc une variable d'entrée pour C.F.2.

L'initialisation des facteurs d'échelle est la même que précédemment (voir l'équation (1.18)) avec en plus $K_{\Delta u_2} = \frac{1}{K_{\Delta q_2}} = 10$ afin de normaliser la 2^{ème} entrée du contrôleur flou C.F.2. Les ensembles flous sont uniformément répartis sur leur univers de discours. Les bases de règles initiales, B.R.1 et B.R.2, associées, respectivement aux contrôleurs flous C.F.1 et C.F.2 de la structure hiérarchique, sont respectivement données dans les tables 1.3 et 1.4.

X_c	N	Z	P
Δq_2	-1	0	1

TAB. 1.3 – Base de règles initiale associée au contrôleur flou C.F.1

Δu_2	N	Z	P
X_h	-1	-1	-1
	0	0	0
	1	1	1

TAB. 1.4 – Base de règles initiale associée au contrôleur flou C.F.2

Notons qu'à l'initialisation, ces deux structures de commandes, décentralisée (figure 1.5) et hiérarchique (figure 1.6), sont identiques.

1.6 Analyse des résultats obtenus dans le cas d'une fusion mathématique

La dimension du problème de commande est ici réduite à travers une fusion mathématique formulée dans les équations (1.14) et (1.15). Les résultats obtenus avec les deux structures de commande, décentralisée et hiérarchique, sont maintenant présentés.

1.6.1 Evolution du critère quadratique C

L'évolution du critère quadratique (1.7) au cours des deux phases de mise au point des contrôleurs flous (réglage des gains et optimisation des ensembles flous) est représentée dans la figure 1.7 pour la structure de commande floue décentralisée, et dans la figure 1.8 pour la structure de commande floue hiérarchique.

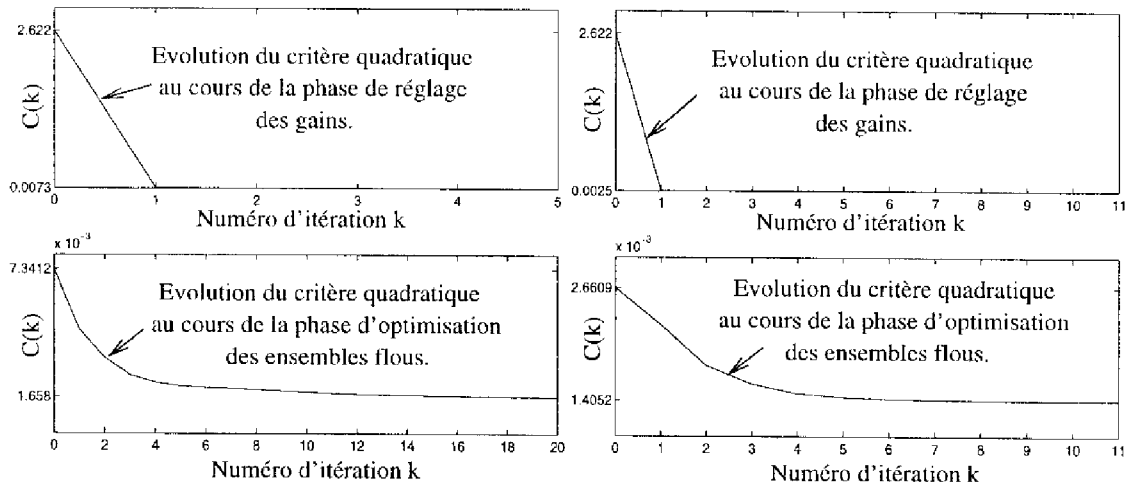


FIG. 1.7 – Structure de commande décentralisée: évolution du critère

FIG. 1.8 – Structure de commande hiérarchique: évolution du critère

Notons que la phase de réglage des gains entraîne une diminution rapide du critère quadratique. La phase d'optimisation des ensembles flous est une approche de précision et conduit à un réglage *plus en finesse* des contrôleurs flous.

1.6.2 Evolution des paramètres des contrôleurs flous

Structure de commande décentralisée

L'évolution des gains au cours de la phase finale de mise au point du contrôleur flou décentralisé est donnée dans le tableau 1.5.

Les fonctions d'appartenance associées aux contrôleurs flous C.F.1 et C.F.2, après optimisation, sont représentées dans la figure 1.9.

Gains	C.F.1			C.F.2		
	K_{ϵ_h}	$K_{\Delta\epsilon_h}$	$K_{\Delta q_1}$	K_{ϵ_c}	$K_{\Delta\epsilon_c}$	$K_{\Delta q_2}$
après réglage	10.04	499.98	0.92	100.05	2550	0.27
après optimisation	10.13	504.39	1.25	104.3	2658.2	0.36

TAB. 1.5 - Evolution des facteurs d'échelle

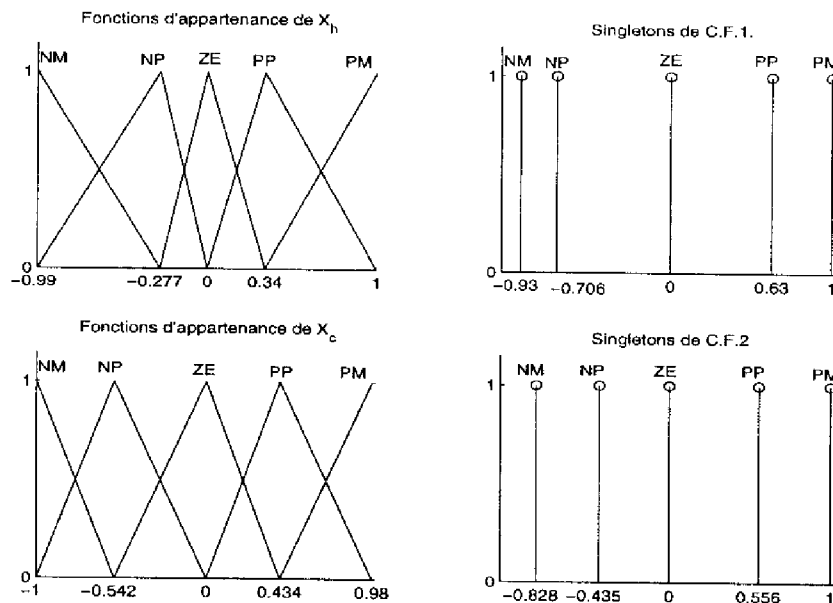


FIG. 1.9 - Fonctions d'appartenance après optimisation

Structure de commande hiérarchique

L'évolution des gains au cours de la phase finale de mise au point du contrôleur flou hiérarchique est donnée dans le tableau 1.6. Précisons que le gain $K_{\Delta u_2}$ n'est pas réglé lors de la première phase de l'algorithme mais mis à jour par $K_{\Delta u_2} = \frac{1}{K_{\Delta q_2}}$ normalisant ainsi la seconde entrée du contrôleur flou C.F.2.

Gains	C.F.1			C.F.2			
	K_{ϵ_c}	$K_{\Delta\epsilon_c}$	$K_{\Delta q_2}$	K_{ϵ_h}	$K_{\Delta\epsilon_h}$	$K_{\Delta u_2}$	$K_{\Delta q_1}$
après réglage	99.94	2550	0.51	9.95	500.01	1.96	1.67
après optimisation	99.70	2544	0.535	9.97	501.26	1.95	1.698

TAB. 1.6 - Evolution des facteurs d'échelle

Les bases de règles associées aux contrôleurs flous C.F.1 et C.F.2, après optimisation, sont respectivement données dans les tables 1.7 et 1.8.

X_c	N	Z	P
Δq_2	-0.94	0	1

TAB. 1.7 - Base de règles associée au contrôleur flou C.F.1

Δu_2		N	Z	P
X_h	N	-0.92	-0.91	-0.96
	Z	0.22	0	-0.27
	P	0.95	1	0.74

TAB. 1.8 - Base de règles associée au contrôleur flou C.F.2

Notons que les deux règles suivantes, règles qui initialement avaient une conclusion nulle, sont cohérentes :

$$\begin{cases} \text{SI } X_h \text{ est Z et } \Delta u_2 \text{ est N ALORS } \Delta q_1 \text{ est } 0.22. \\ \text{SI } X_h \text{ est Z et } \Delta u_2 \text{ est P ALORS } \Delta q_1 \text{ est } -0.27. \end{cases}$$

En effet, lorsque X_h est Zéro, ces deux règles compensent l'action de Δq_2 sur h .

Les fonctions d'appartenance en entrée, associées aux contrôleurs flous C.F.1 et C.F.2, après optimisation, sont représentées dans la figure 1.10.

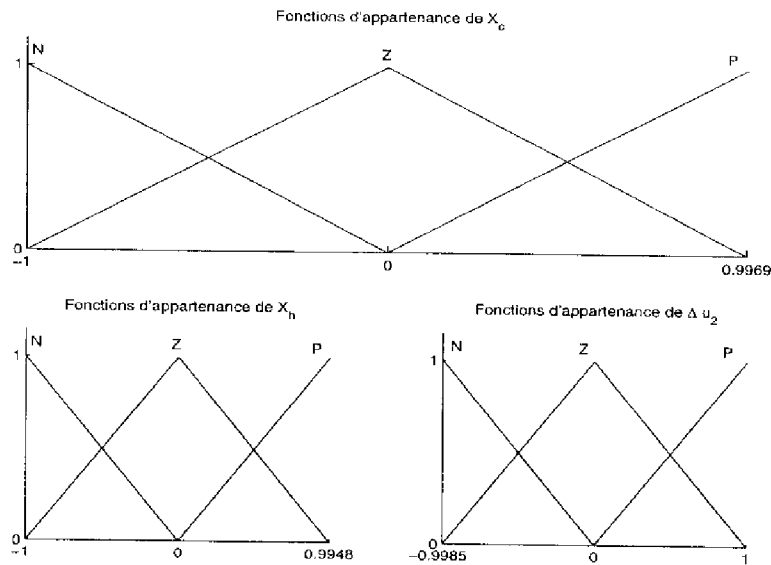


FIG. 1.10 - Fonctions d'appartenance après optimisation

1.6.3 Résultats obtenus par rapport à la trajectoire de référence

Le suivi de la trajectoire en concentration, représenté dans la figure 1.11, est satisfaisant et les interactions sont très légèrement mieux compensées par la structure hiérarchique. Quant au suivi de la trajectoire en hauteur, représenté dans la figure

1.12, il est très satisfaisant et aucune différence n'est visible entre les deux structures de commande floue et la trajectoire de référence.

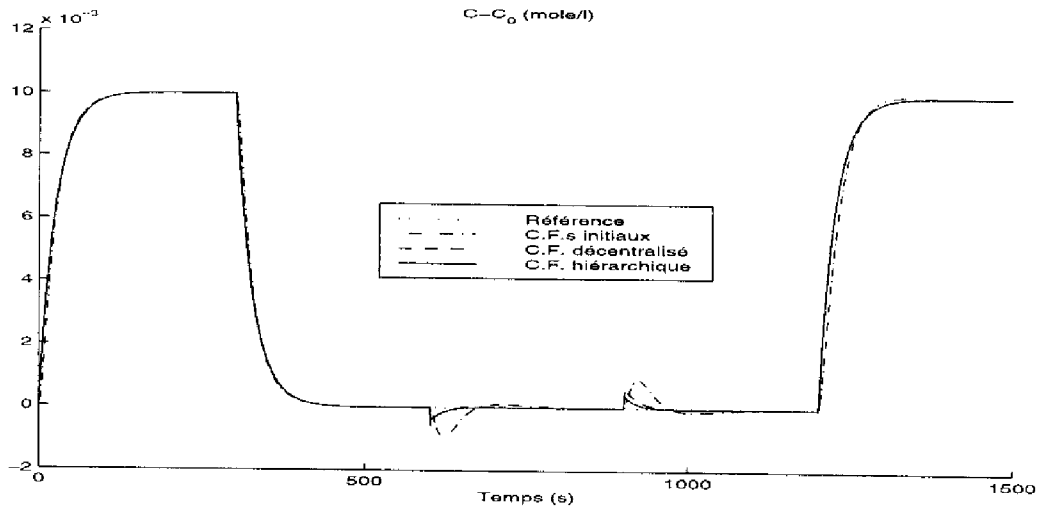


FIG. 1.11 - Suivi de la trajectoire de référence en concentration.

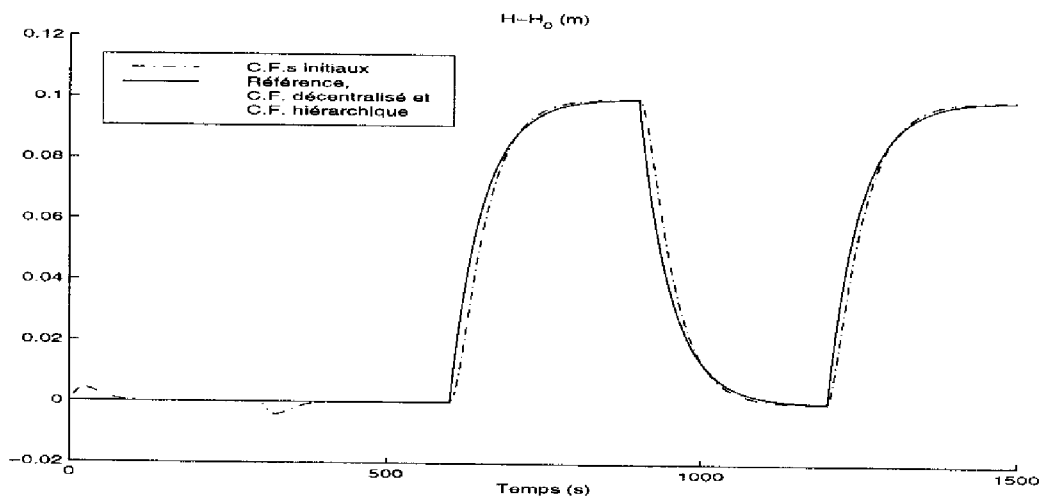


FIG. 1.12 - Suivi de la trajectoire de référence en hauteur.

Nous allons maintenant comparer ces deux contrôleurs flous, décentralisé et hiérarchique, à une loi de commande optimale. Les comparaisons portent sur une consigne de hauteur seule (ce qui correspond à la période allant de 600 à 900 secondes), puis, sur une consigne de concentration seule (période allant de 0 à 300 secondes). Ce choix est fait, d'une part, afin d'observer avec davantage de précision la compensation des interactions, d'autre part, afin de ne pas surcharger inutilement le chapitre, les trajectoires de référence étant parfaitement suivies par ailleurs.

1.6.4 Comparaison avec une commande optimale

Consigne de hauteur

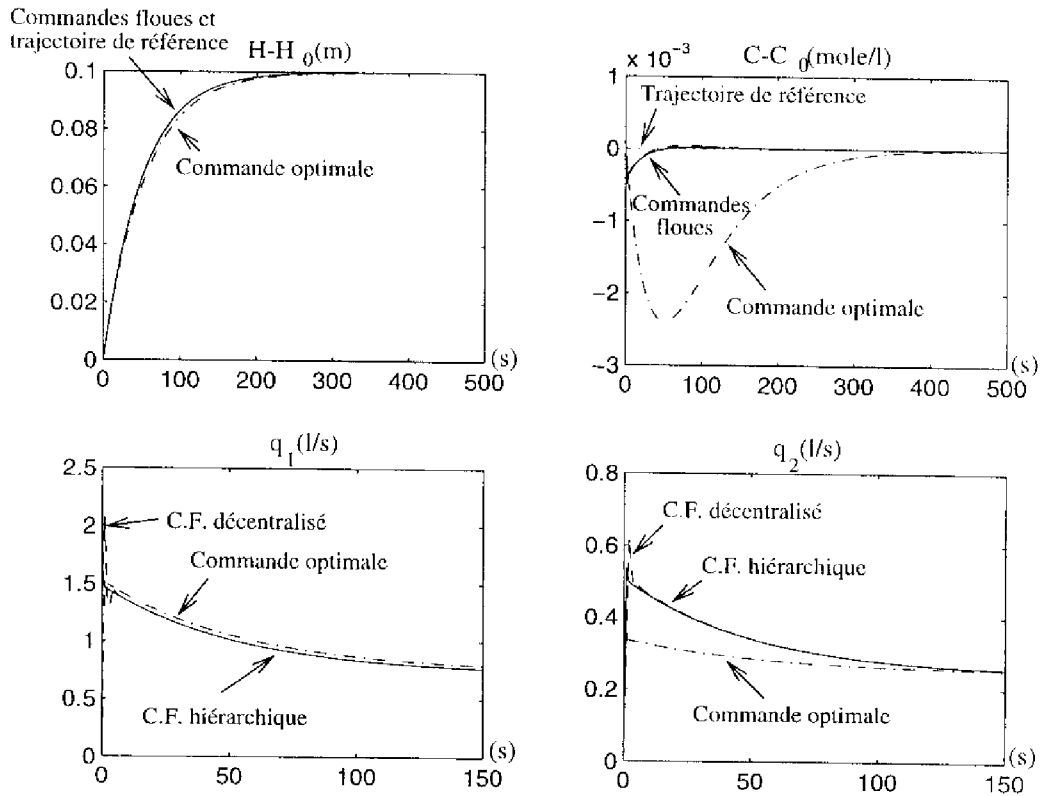


FIG. 1.13 - Consigne de hauteur: $h_c = 0.1$ m et $c_c = 0$ mole/l

Simulation autour de PDF1	Contrôleur flou		Commande optimale
	décentralisé	hiérarchique	
$J(\epsilon_h)$	25.23	25.21	26.87
$J(\epsilon_c) \cdot 10^{-4}$	1.31	1.12	380
$J(q_1, q_2)$	304.31	303.54	296.51
J	329.54	328.75	323.42

TAB. 1.9 - Critères de performance - consigne de hauteur

Les résultats obtenus avec les deux structures de commande floue, décentralisée et hiérarchique, sont similaires. Le comportement dynamique de la hauteur est conforme au cahier des charges et les interactions sur la concentration sont bien compensées (voir figure 1.13 et table 1.9). Par contre, les deux structures de commande floue sollicitent un peu plus les actionneurs (+2.63% de consommation pour la structure décentralisée et +2.37% pour la structure hiérarchique).

Consigne de concentration

Comme prévu, la structure de commande hiérarchique compense davantage les interactions sur la hauteur (voir figure 1.14 et table 1.10). Le comportement dynamique de la concentration est tout à fait satisfaisant pour les deux structures de commande floue: la trajectoire de référence est bien suivie. Le coût de commande et le critère global restent cependant supérieurs à ceux de la commande optimale (+12.02% de consommation pour la structure décentralisée et +11.53% pour la structure hiérarchique).

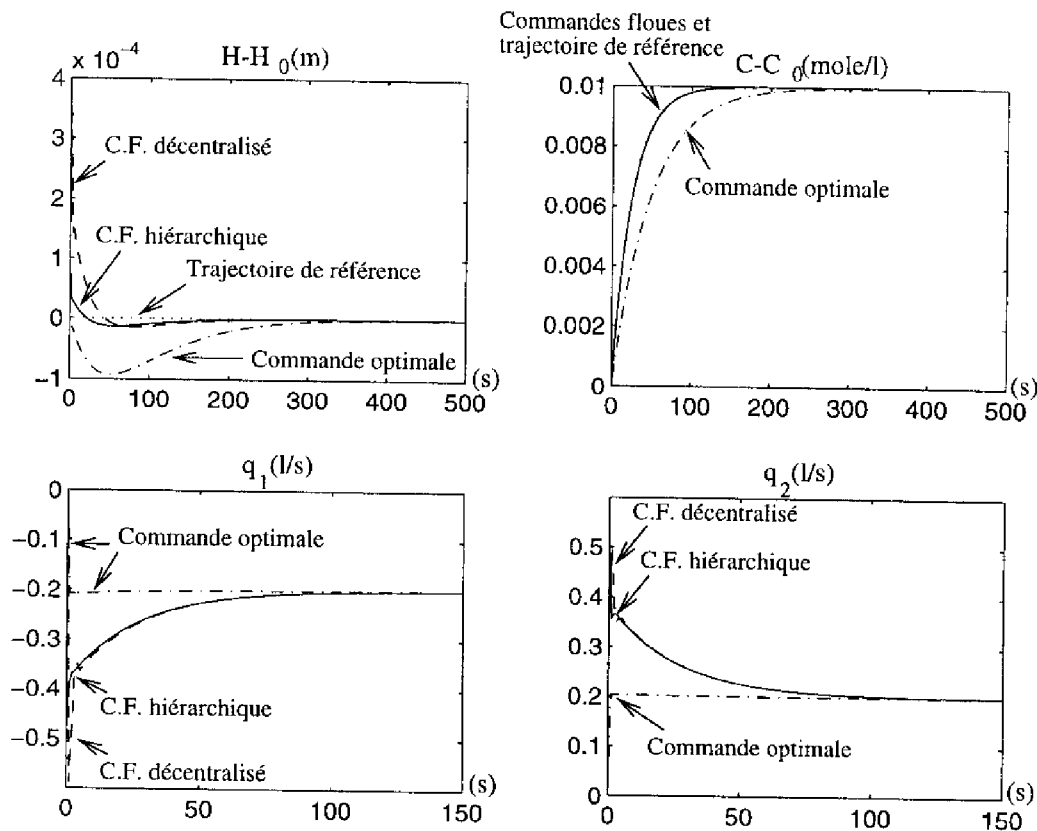


FIG. 1.14 - Consigne de concentration: $h_c = 0$ m et $c_c = 0.01$ moles/l

Pour conclure, nous pouvons dire que la structure de commande hiérarchique, contenant seulement 12 règles de commande, améliore les performances sans pour autant accroître le coût énergétique; bien que ce dernier soit cependant supérieur à celui de la commande optimale, il faut noter que la commande floue ne sollicite pas les actionneurs de façon brutale et que les temps de réponse des stratégies floues sont plus rapides.

Simulation autour de PDF1	Contrôleur flou		Commande optimale
	décentralisé	hiérarchique	
Nombre de règles	10	12	—
$J(c_h).10^{-5}$	4.40	0.23	8.28
$J(c_c).10^{-2}$	8.83	8.77	17
$J(q_1, q_2)$	99.98	99.54	89.25
J	100.07	99.63	89.42

TAB. 1.10 – Critères de performance - consigne de concentration

1.6.5 Analyse expérimentale de la robustesse

Les stratégies de commande floue ont été réglées autour du point de fonctionnement PDF1 (1.5). Nous évaluons maintenant leur robustesse en testant ces contrôleurs flous autour d'un régime nominal différent (PDF2) défini par (1.12).

Par rapport aux trajectoires de référence de la figure 1.2, les réponses obtenues, pour les deux structures de commande floue, sont similaires à celles représentées dans les figures 1.11 et 1.12. Par conséquent, les deux stratégies de commande floue sont robustes par rapport aux variations des paramètres du modèle. Toutefois, la comparaison des critères quadratiques, $J(c_h)$ et $J(c_c)$, obtenus autour de PDF2, avec ceux obtenus autour de PDF1, pour une consigne de hauteur seule (table 1.11) puis de concentration seule (table 1.12), permet une meilleure évaluation de la robustesse des deux contrôleurs flous.

	Contrôleur flou			
	décentralisé		hiérarchique	
	PDF		PDF	
	1	2	1	2
$J(c_h)$	25.23	25.18	25.21	23.14
$J(c_c).10^{-4}$	1.31	1.08	1.12	0.82

TAB. 1.11 – Critères de performance - consigne de hauteur

	Contrôleur flou			
	décentralisé		hiérarchique	
	PDF		PDF	
	1	2	1	2
$J(c_h).10^{-5}$	4.40	22.24	0.23	1.68
$J(c_c).10^{-2}$	8.83	9.24	8.77	9.08

TAB. 1.12 – Critères de performance - consigne de concentration

Pour une consigne de hauteur (table 1.11), la trajectoire de référence h_d est parfaitement suivie et les interactions sur la concentration sont bien compensées. Pour une consigne de concentration (table 1.12), la concentration suit bien sa trajectoire de référence mais les interactions sur la hauteur sont moins bien contrôlées. Toutefois, la compensation des couplages sur la hauteur, avec la structure de commande hiérarchique, reste tout à fait satisfaisante.

Pour conclure, nous pouvons dire que le contrôleur flou hiérarchique est plus robuste que le contrôleur flou décentralisé, ce qui est tout à fait logique.

1.7 Analyse des résultats obtenus dans le cas d'une fusion symbolique

Nous avons vu, dans le paragraphe précédent, que les résultats obtenus avec la structure de commande floue hiérarchique sont meilleurs que ceux obtenus avec la structure de commande décentralisée. De plus, le coût en terme de commande est inférieur. Par conséquent, seule la structure de commande hiérarchique est ici adoptée. La dimension du problème de commande est ici réduite à travers une fusion symbolique formulée dans les équations (1.16) et (1.17).

Dans ce paragraphe, uniquement les résultats pour une consigne de hauteur seule, puis une consigne de concentration seule, sont reproduits, les trajectoires de références étant parfaitement suivies lorsque les deux variables h et c doivent rallier une consigne.

1.7.1 Evolution du critère quadratique C

L'évolution du critère quadratique (1.7) au cours des deux phases de mise au point du contrôleur flou hiérarchique, réglage des gains et optimisation des ensembles flous, est représentée dans la figure 1.15.

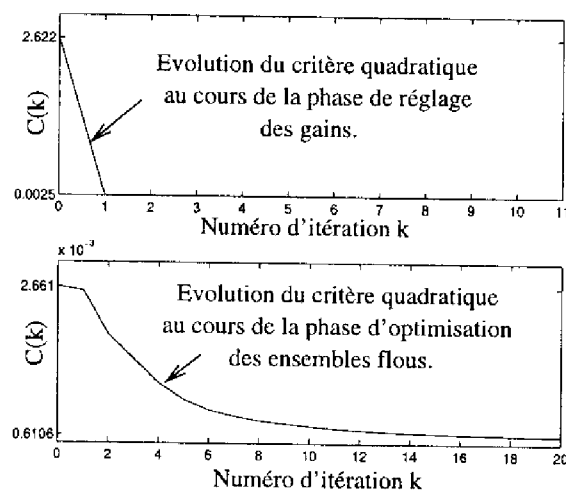


FIG. 1.15 – Structure de commande hiérarchique et fusion symbolique: évolution du critère

Par rapport à la structure hiérarchique précédente, basée sur une fusion mathématique, la minimisation du critère quadratique est améliorée lors de la phase d'optimisation des ensembles flous (comparer les figures 1.8 et 1.15). Ceci s'explique de part le nombre de degrés de liberté supplémentaires offert par les fonctions d'appartenance bidimensionnelles. Toutefois, cette amélioration a un coût. En effet, ce

type d'ensembles flous construits à partir d'une triangulation de Delaunay est plus gourmand en temps calcul.

1.7.2 Evolution des paramètres du contrôleur flou hiérarchique

L'évolution des gains au cours de la phase finale de mise au point du contrôleur flou hiérarchique est donnée dans le tableau 1.13. Le gain $K_{\Delta u_2}$ n'est pas réglé lors de la première phase de l'algorithme mais mis à jour par $K_{\Delta u_2} = \frac{1}{K_{\Delta q_2}}$.

Gains	C.F.1			C.F.2			
	K_{c_c}	$K_{\Delta c_c}$	$K_{\Delta q_2}$	K_{ϵ_h}	$K_{\Delta \epsilon_h}$	$K_{\Delta u_2}$	$K_{\Delta q_1}$
après réglage	99.94	2550	0.51	9.95	500.01	1.96	1.67
après optimisation	99.906	2538.6	0.546	9.93	499.03	1.95	1.7

TAB. 1.13 - Evolution des facteurs d'échelle

Seules les fonctions d'appartenance associées aux entrées du contrôleur flou C.F.2, après optimisation, sont ici représentées (voir la figure 1.16), les ensembles flous bidimensionnels associés à X_c étant similaires à ceux définis pour X_h .

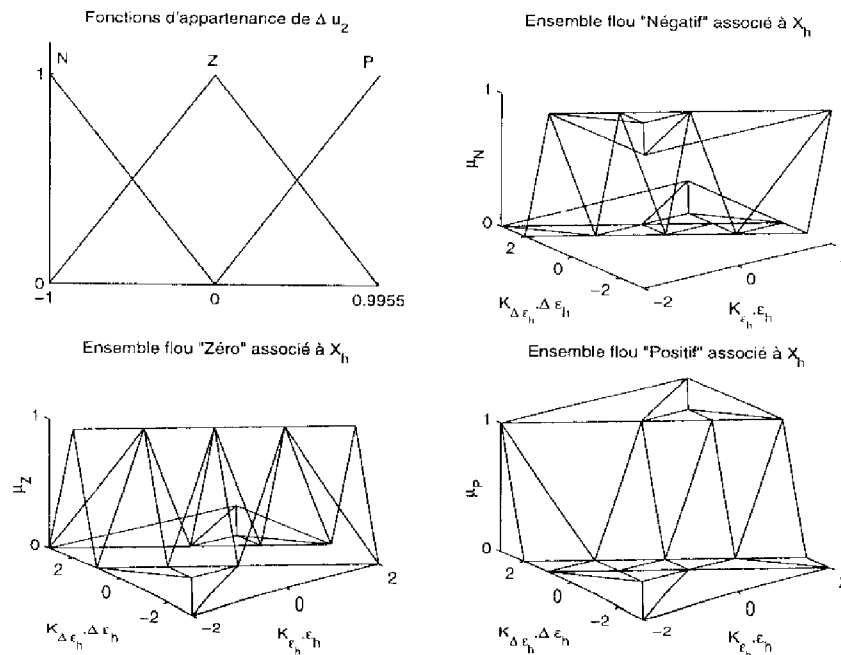


FIG. 1.16 - Fonctions d'appartenance de C.F.2 après optimisation

Les bases de règles associées aux contrôleurs flous C.F.1 et C.F.2, après optimisation, sont respectivement données dans les tables 1.14 et 1.15.

X_c	N	Z	P
Δq_2	-0.94	0	1

TAB. 1.14 - Base de règles associée au contrôleur flou C.F.1

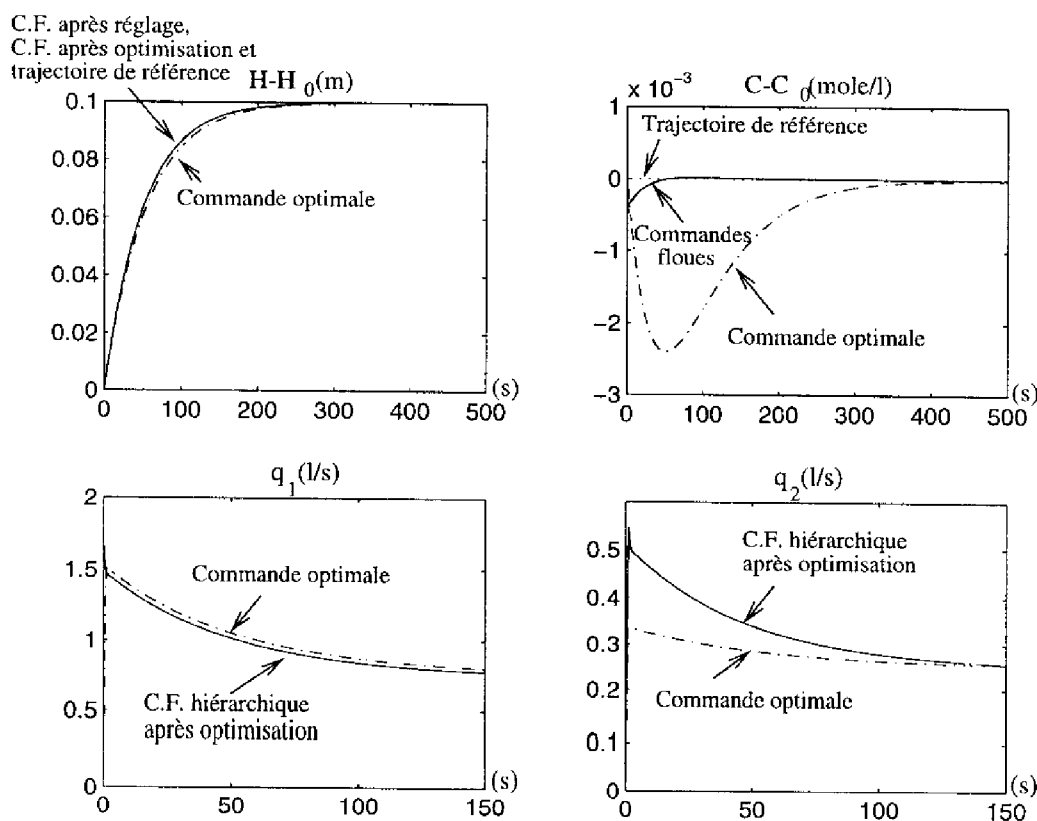
Δu_2	N	Z	P	
X_h	N	-0.91	-0.916	-0.95
	Z	0.21	0	-0.27
	P	0.94	1	0.74

TAB. 1.15 - Base de règles associée au contrôleur flou C.F.2

1.7.3 Comparaison avec une commande optimale

Consigne de hauteur

La hauteur suit bien sa trajectoire de référence et les interactions sur la concentration sont bien compensées (voir figure 1.17 et table 1.16). La commande floue sollicite un peu plus les actionneurs (+2.49% de consommation). Notons que la différence des performances avant et après la phase d'optimisation des ensembles flous est minime.

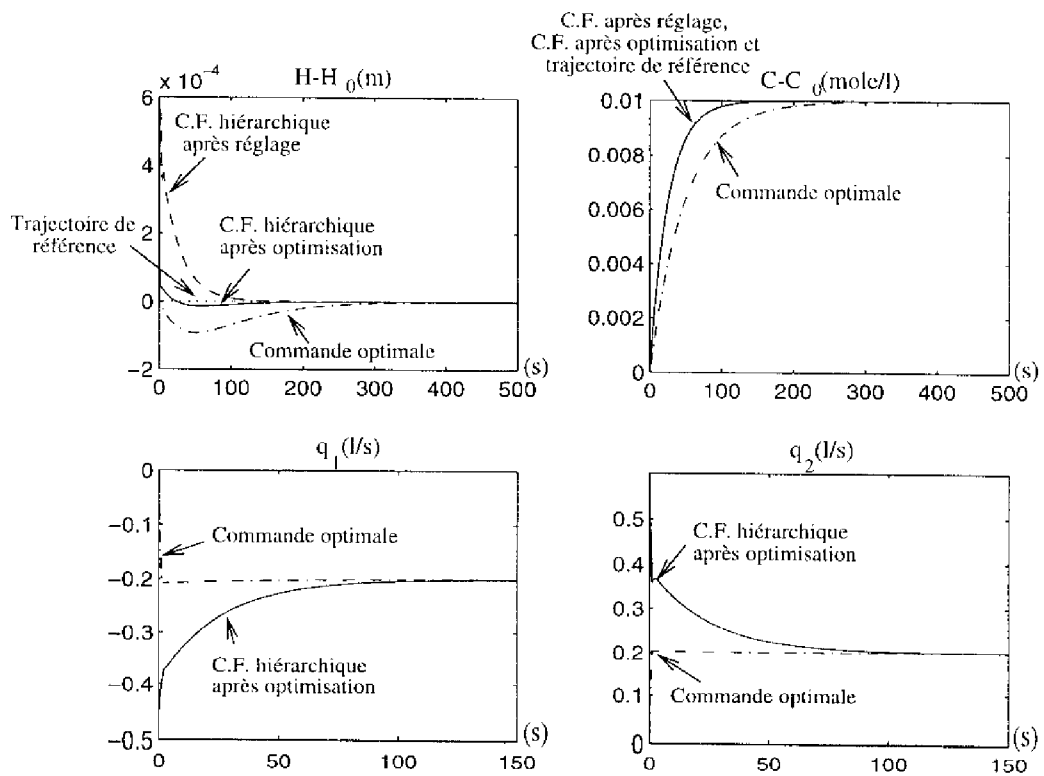
FIG. 1.17 - Consigne de hauteur: $h_c = 0.1$ m et $c_c = 0$ mole/l

Simulation autour de PDF1	Contrôleur flou réglé	Contrôleur flou optimisé	Commande optimale
$J(c_k)$	25.06	25.12	26.87
$J(c_c) \cdot 10^{-4}$	1.23	1.06	380
$J(q_1, q_2)$	304.05	303.88	296.51
J	329.11	329	323.42

TAB. 1.16 – Critères de performance - consigne de hauteur

Consigne de concentration

Le comportement dynamique de la concentration est déjà satisfaisant après le réglage des gains (voir figure 1.18). La phase d'optimisation des ensembles flous contribue principalement à améliorer la compensation des interactions sur la hauteur (voir figure 1.18 et table 1.17). Les actionneurs sont davantage sollicités par la commande floue (+11.69% de consommation).

FIG. 1.18 – Consigne de concentration: $h_c = 0$ m et $c_c = 0.01$ moles/l

Simulation autour de PDF1	Contrôleur flou		Commande optimale
	réglé	optimisé	
$J(\epsilon_h) \cdot 10^{-5}$	28.5	0.33	8.28
$J(\epsilon_c) \cdot 10^{-2}$	8.82	8.69	17
$J(q_1, q_2)$	99.82	99.68	89.25
J	99.91	99.76	89.42

TAB. 1.17 – Critères de performance - consigne de concentration

1.7.4 Analyse expérimentale de la robustesse

Nous comparons maintenant les simulations autour de PDF1 avec celles obtenues autour de PDF2, et ce, sans toucher au réglage du contrôleur flou hiérarchique.

Lors d'une consigne de hauteur (concentration), la hauteur (concentration) suit parfaitement sa trajectoire de référence : voir table 1.18 (1.19). Les interactions sur la concentration sont légèrement mieux compensées et celles sur la hauteur sont moins bien contrôlées mais restent faibles (voir la figure 1.19 et les tables 1.18 et 1.19).

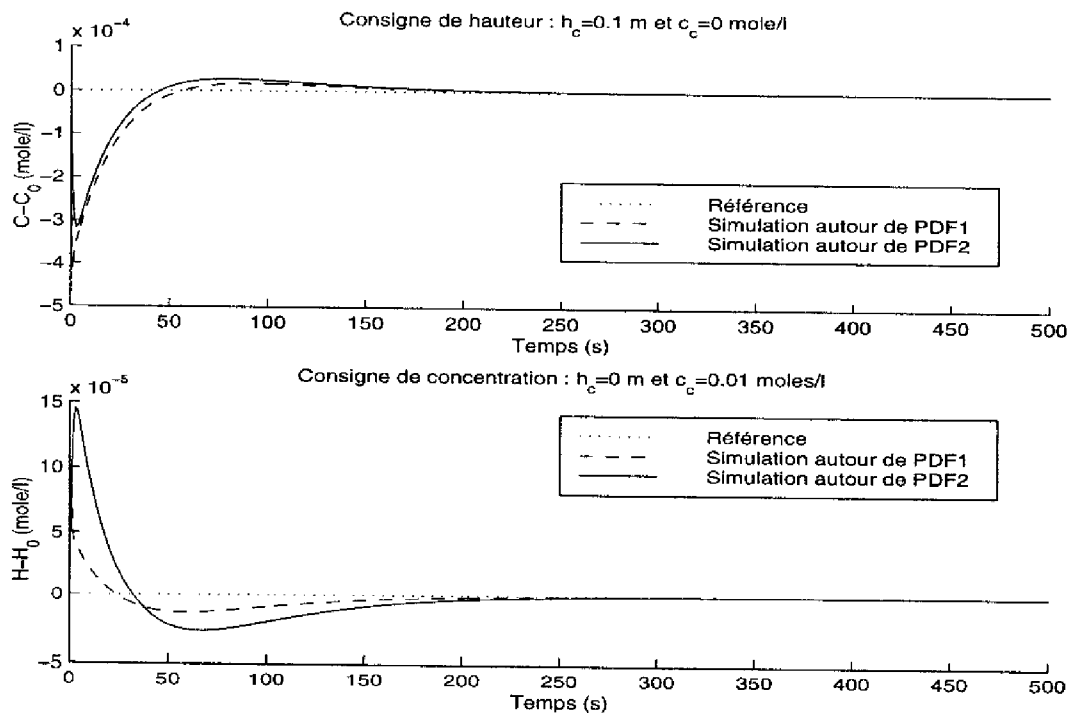


FIG. 1.19 – Compensation des interactions autour de PDF1 et PDF2

Contrôleur flou hiérarchique	Simulation	
	PDF1	PDF2
$J(\epsilon_h)$	25.12	25.05
$J(\epsilon_c).10^{-4}$	1.06	0.79

TAB. 1.18 - Critères de performance - consigne de hauteur

Contrôleur flou hiérarchique	Simulation	
	PDF1	PDF2
$J(\epsilon_h).10^{-5}$	0.33	2.1
$J(\epsilon_c).10^{-2}$	8.69	9

TAB. 1.19 - Critères de performance - consigne de concentration

1.8 Conclusion

Dans ce chapitre, nous avons appliqué la méthodologie présentée dans la partie II, sur un exemple académique: un mélangeur multivariable à deux entrées et deux sorties.

Nous avons procédé, dans un premier temps, pour chacune des variables régularisées, à une fusion mathématique de l'erreur (entre cette variable et sa consigne) et de sa variation. Une analyse qualitative du processus nous a permis de déduire différentes stratégies de commande floue, décentralisée et hiérarchique, contenant respectivement 10 et 12 règles de commandes.

Les contrôleurs flous sont initialisés à partir de la connaissance disponible sur le procédé. Cette connaissance est ainsi traduite sous forme de règles linguistiques. La mise au point de ces différents contrôleurs flous est réalisée à travers la poursuite d'une trajectoire de référence, et ce à l'aide d'une technique de descente du gradient. Ces deux lois de commande floue, décentralisée et hiérarchique, sont positivement comparées à une loi de commande optimale. La structure de commande hiérarchique améliore les performances et sa robustesse a pu être vérifiée.

Par la suite, seule la structure de commande hiérarchique est conservée et une fusion symbolique des variables est appliquée. Cette dernière repose sur l'utilisation d'ensembles flous bidimensionnels et permet une meilleure diminution du critère quadratique.

Ce type d'approche, réduction de la dimension du problème puis structuration de la loi de commande, nous paraît d'ores et déjà fort intéressant. Le nombre de règles est considérablement réduit et cela facilite non seulement l'écriture des règles mais aussi l'auto-régulation du contrôleur flou. De plus, les règles de commande floue sont simples et explicites et peuvent être validées par un opérateur possédant un minimum d'expérience sur le processus.

Dans le chapitre suivant, nous nous intéressons à un problème plus concret: un procédé biologique de traitement des eaux usées.

Chapitre 2

Application à un bioréacteur

Nous nous intéressons, dans ce chapitre, à un procédé biologique de traitement des eaux usées dont l'étude se ramène à celle d'un bioréacteur contenant une population mixte alimenté en mode continu par un mélange bi-substrat.

Seules les techniques de réduction par fusion mathématique et hiérarchisation sont ici appliquées. Elles conduisent à un contrôleur flou hiérarchique contenant seulement 14 règles de commande, lesquelles permettent de réguler de façon très satisfaisante les concentrations en biomasse (micro-organismes) et en substrat xénotique (matière polluante). Ce contrôleur flou hiérarchique est mis au point à l'aide d'une technique de descente du gradient. La robustesse de la stratégie de commande floue est testée expérimentalement.

2.1 Présentation du procédé

Ces dernières années ont vu se développer de nombreux efforts en matière de protection de l'environnement. Aussi, le recours aux procédés biologiques d'épuration des eaux usées, en tant qu'outil de dépollution des effluents industriels, s'est multiplié. Parmi les techniques utilisées, celle du lagunage aéré est la plus ancienne mais aussi la plus intéressante d'un point de vue économique. Elle consiste à déverser l'effluent dans un bassin (ou une association de bassins) où séjourne une population mixte de bactéries destinée à digérer la matière polluante. La lagune est équipée d'aérateurs, ce qui permet une meilleure oxygénation des eaux, et par conséquent, intensifie leur traitement.

Le procédé ici étudié est extrait de [You96] et concerne la dépollution biologique des effluents d'une industrie papetière par lagunage aéré. Ce type d'effluent est composé de deux substrats :

- un substrat dit *énergétique* caractérisé par une forte biodégradabilité,
- un substrat dit *xénobiotique* caractérisé par une faible biodégradabilité.

Le premier type de substrat correspond à de la cellulose. Il constitue la principale source énergétique de croissance des micro-organismes. Le second type de substrat est une synthèse issue de l'activité humaine et est principalement constitué d'un mélange de dérivés ligneux et de lignosulfonates. Il est considéré comme le principal substrat polluant à dégrader par voie microbienne.

Dans ce contexte, l'optimisation de la biodégradation de la matière polluante se ramène à l'étude d'un bioréacteur contenant une population mixte alimenté en mode continu par un mélange bi-substrat.

2.2 Modélisation mathématique du procédé

2.2.1 Equations d'évolution du procédé

Les équations d'évolution pour chacun des éléments macroscopiques de la réaction biologique, biomasse et substrats, sont issues de bilans matières sur un intervalle de temps déterminé :

$$\text{Variation} = \pm \text{Conversion} + \text{Alimentation} - \text{Soutirage}$$

où le terme de conversion correspond soit à une vitesse de production, soit à une vitesse de consommation.

Les dynamiques du procédé d'épuration d'un effluent bi-substrat, dans le cas d'un réacteur homogène à alimentation et soutirage permanent, donc de volume

constant, sont modélisées par le système d'équations différentielles suivantes:

$$\begin{cases} \dot{c}(t) = \mu_c(t)c(t) - D(t)c(t) \\ \dot{s}(t) = -\frac{\mu_s(t)}{Y_{c/s}}c(t) + D(t)s_a(t) - D(t)s(t) \\ \dot{e}(t) = -\frac{\mu_e(t)}{Y_{c/e}}c(t) + D(t)e_a(t) - D(t)e(t) \end{cases} \quad (2.1)$$

où $c(t)$, $s(t)$ et $e(t)$ représentent respectivement les concentrations en g/l de biomasse, de substrat xénobiotique et de substrat énergétique. $\mu_c(t)$ représente la vitesse spécifique de croissance de la biomasse. $\frac{\mu_s(t)}{Y_{c/s}}$ et $\frac{\mu_e(t)}{Y_{c/e}}$ représentent respectivement les taux spécifiques de dégradation du substrat xénobiotique et du substrat énergétique. $Y_{c/s}$, respectivement $Y_{c/e}$, représente le rendement de conversion du substrat xénobiotique, respectivement énergétique, en biomasse. Le taux de dilution $D(t)$ correspond au rapport entre le débit d'alimentation et le volume constant du bioréacteur. $s_a(t)$ et $e_a(t)$ sont respectivement les concentrations en substrat xénobiotique et énergétique du milieu d'alimentation.

Les taux spécifiques, $\mu_s(t)$ et $\mu_e(t)$, sont des paramètres essentiels pour la description du procédé. Ils dépendent des conditions opératoires (pH, température, oxygénation) ainsi que des concentrations en substrats $s(t)$ et $e(t)$. La complexité de modélisation de ces variables est néanmoins réduite par le fait que leur expression est établie pour des conditions opératoires optimales fixes.

2.2.2 Modélisation des cinétiques de croissance et de dégradation

Le comportement de populations mixtes dans un mélange multi-substrat dépend des affinités de chaque micro-organisme vis à vis de chaque substrat. Les cinétiques d'utilisation des substrats sont gouvernées par les concentrations et les types de substrat présents dans le milieu mais également par les interactions existant entre les différentes espèces microbiennes.

Pour ce procédé d'épuration des effluents de papeterie, la croissance de la population mixte est définie par la somme des taux de croissance provenant de la conversion des deux types de substrats, xénobiotique et énergétique, en biomasse :

$$\mu_c(t) = \mu_s(t) + \mu_e(t) \quad (2.2)$$

La prise en compte des effets croisés des substrats $s(t)$ et $e(t)$ sur l'évolution des taux spécifiques de croissance $\mu_s(t)$ et $\mu_e(t)$ peut être correctement représentée par le modèle de Monod généralisé de la forme [YKB77]:

$$\mu_s(t) = \mu_{sm} \frac{s(t)}{K_s + s(t) + a_e e(t)} \quad (2.3)$$

$$\mu_e(t) = \mu_{cm} \frac{c(t)}{K_e + c(t) + a_s s(t)} \quad (2.4)$$

où μ_{sm} et μ_{cm} correspondent aux taux spécifiques de croissance maximum. Les paramètres de Michaelis-Menten, K_s et K_e , permettent de tenir compte de l'effet de limitation de croissance par chaque substrat. La constante a_e permet de modéliser l'effet inhibiteur de la présence du substrat énergétique sur la consommation du substrat xénobiotique (et inversement pour la constante a_s). Théoriquement, ces paramètres vérifient la relation $a_e = 1/a_s$. Pour notre étude, la consommation préférentielle du substrat énergétique par les micro-organismes se traduit par une faible valeur de a_s et par conséquent, une forte valeur de a_e .

Une simulation du procédé avec un taux de dilution nul (culture discontinue) est représentée dans la figure 2.1 et permet de mettre en évidence les phénomènes de compétition et d'inhibition/activation compétitive de la culture mixte en milieu bi-substrat.

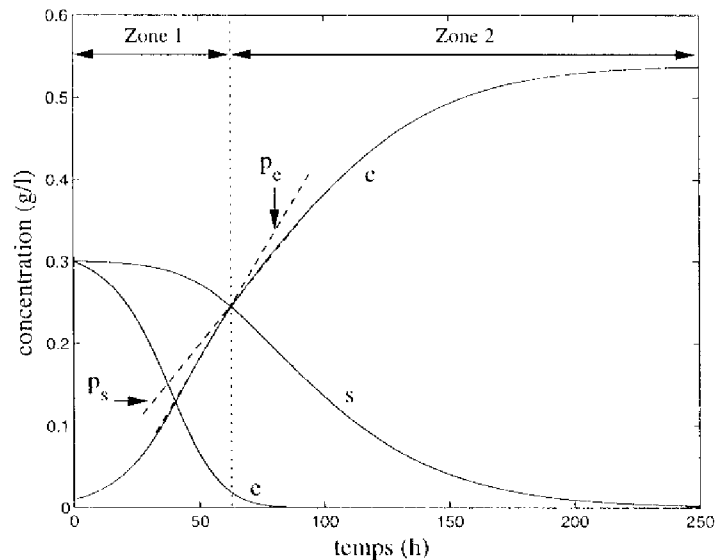


FIG. 2.1 - Interactions microbiennes de la biomasse avec les deux substrats.

Nous observons, dans la première partie de cette courbe (zone 1), que la dégradation du substrat xénobiotique $s(t)$ est presque complètement inhibée par la seule présence du substrat énergétique $c(t)$. C'est seulement lorsque ce dernier est pratiquement consommé (zone 2) qu'a lieu la dégradation du substrat xénobiotique. Notons que la croissance de la population mixte $c(t)$ s'effectue alors plus lentement (pente p_s plus faible que p_c).

Au vu de la précédente figure, on pourrait penser, à tort, qu'une absence totale de substrat énergétique permettrait une meilleure dégradation du substrat xéno-

biotique. Cependant, en dehors du fait que la présence du substrat énergétique est imposée par la nature même du procédé, sa présence est indispensable au bon fonctionnement du processus de dépollution. En effet, pour qu'un bioréacteur alimenté en mode continu soit stable, il faut qu'il y ait au moins un substrat qui puisse garantir une croissance suffisante des micro-organismes. Ce rôle est tenu par le substrat énergétique dont la conversion en biomasse permet d'assurer un taux spécifique de croissance $\mu_c(t)$ supérieur au taux de dilution du réacteur, sans quoi, celui-ci risquerait d'être lavé (terme couramment utilisé en biochimie pour signifier l'absence de biomasse dans le bioréacteur).

2.3 Présentation du cahier des charges

L'entrée du système est définie en fonction des objectifs de commande mono ou multivariable. Dans le cas monovisible, la grandeur de commande correspond au taux de dilution $D(t)$ du réacteur et la variable régulée est la concentration en substrat xénobiotique $s(t)$. Cependant, si celui-ci constitue la principale matière polluante, le rejet de micro-organismes en trop forte quantité est aussi une sorte de pollution. Par conséquent, les variables que nous cherchons à réguler sont les concentrations en biomasse et en substrat xénobiotique. Une autre grandeur de commande est donc nécessaire pour mener à bien la stratégie de conduite multivariable. Généralement, on prend la concentration en substrats du milieu d'alimentation $s_a(t)$ et $e_a(t)$. En pratique (voir figure 2.2), il suffit de dissocier l'alimentation classique en deux alimentations, l'une apportant de l'eau et l'autre le milieu contenant les deux substrats aux concentrations s_a^{max} et e_a^{max} .

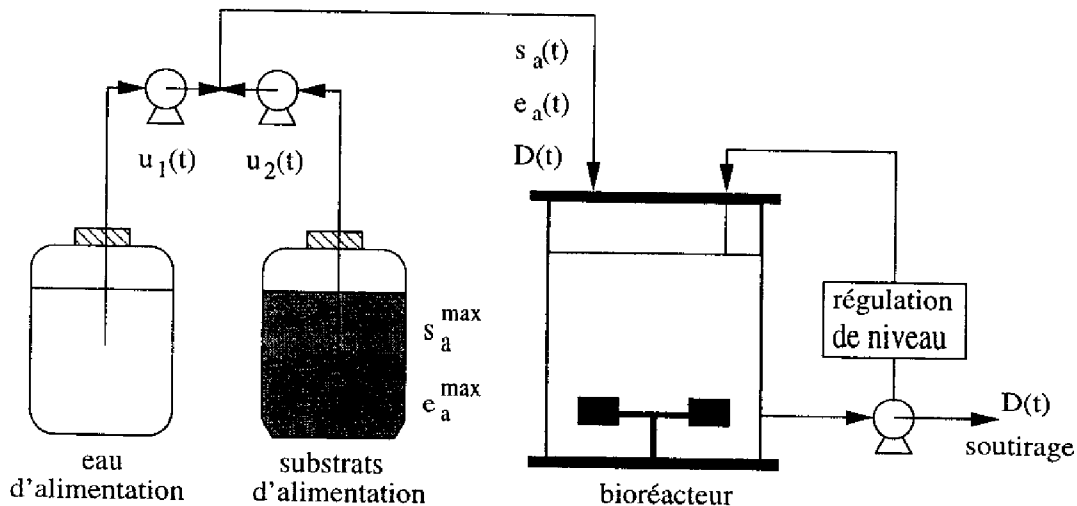


FIG. 2.2 – Alimentation du bioréacteur dans le cas multivariable

Les relations qui lient $D(t)$, $s_a(t)$ et $e_a(t)$ aux variables de commande auxiliaires

$u_1(t)$ et $u_2(t)$ sont données par :

$$\begin{cases} D(t) = u_1(t) + u_2(t) \\ s_a(t) = \frac{u_2(t)}{u_1(t) + u_2(t)} s_a^{max} \\ e_a(t) = \frac{u_2(t)}{u_1(t) + u_2(t)} e_a^{max} \end{cases} \quad (2.5)$$

où $u_1(t)$ et $u_2(t)$ correspondent respectivement aux taux de dilution en eau et en substrats.

Dans tout ce qui suit, nous utilisons le système d'équations différentielles suivant, obtenu par combinaison des équations (2.1) à (2.5), pour simuler le procédé :

$$\begin{cases} \dot{c}(t) = \mu_{sm} \frac{s(t)c(t)}{K_s + s(t) + a_e e(t)} + \mu_{em} \frac{e(t)c(t)}{K_e + e(t) + a_s s(t)} - (u_1(t) + u_2(t)) \cdot c(t) \\ \dot{s}(t) = -\frac{\mu_{sm}}{Y_{c/s}} \frac{s(t)c(t)}{K_s + s(t) + a_e e(t)} + (u_1(t) + u_2(t)) \cdot (s_a^{max} - s(t)) \\ \dot{e}(t) = -\frac{\mu_{em}}{Y_{c/e}} \frac{e(t)c(t)}{K_e + e(t) + a_s s(t)} + (u_1(t) + u_2(t)) \cdot (e_a^{max} - e(t)) \end{cases} \quad (2.6)$$

La méthode d'intégration utilisée est la méthode de Runge-Kutta d'ordre 4 et la période d'échantillonnage T_c est de 15 minutes. Les concentrations maximales des substrats s_a^{max} et e_a^{max} sont de 6 g/l. Les limitations physiques des actionneurs sont les suivantes :

$$0.0001 \leq u_i \leq 0.04 \text{ (1/h) pour } i=1,2. \quad (2.7)$$

Les conditions initiales de démarrage du procédé de dépollution sont données dans le tableau 2.1 :

Description	Paramètres	Valeurs initiales (g/l)
Biomasse	c	0.50
Substrat xénobiotique	s	0.45
Substrat énergétique	e	0.05

TAB. 2.1 - Conditions initiales du bioréacteur

En pratique, ces conditions initiales sont obtenues à partir d'une culture semi-continue (pas de soutirage) de manière à accumuler la biomasse. Lorsque celle-ci atteint une valeur suffisamment importante, on peut démarrer le processus de dépollution.

Les valeurs numériques des paramètres cinétiques sont données dans le tableau 2.2.

Description	Paramètres	Valeurs
Taux maximum de croissance (1/h)	μ_{sm}	0.1
	μ_{em}	0.2
Constantes de Michaelis-Menten (g/l)	K_s	1.5
	K_e	1
Paramètres d'inhibition/activation	a_s	0.1
	a_e	10
Rendements de conversion	$Y_{c/s}$	0.7
	$Y_{c/e}$	0.2

TAB. 2.2 – Valeurs numériques des paramètres cinétiques

Le modèle de référence représenté dans la figure 2.3 est issu de [You96] et traduit les objectifs de commande suivants : compensation des couplages, gain statique unitaire, constantes de temps de 7 heures pour la concentration en biomasse et de 5 heures pour la concentration en substrat xénobiotique.

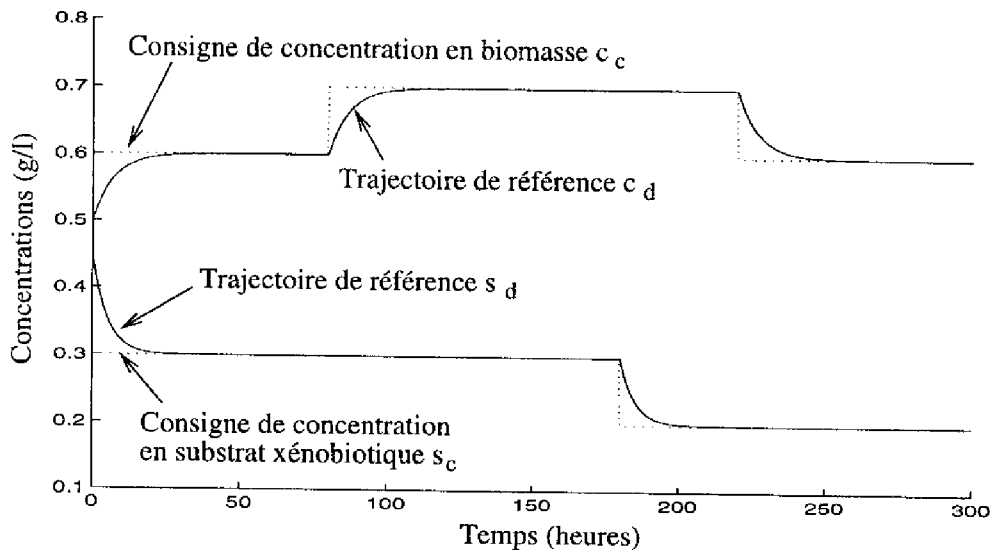


FIG. 2.3 – Trajectoire de référence pour la concentration en biomasse et en substrat xénobiotique.

2.4 Définition d'une stratégie de commande floue

Le succès des stratégies de commande floue provient en grande partie de leur potentiel de transfert de connaissances expertes sur le procédé en règles de commande floue. Aussi, l'écriture de la base de règles et la définition de la structure du contrôleur flou nécessite un minimum de connaissance sur le processus à commander.

Cette connaissance est ici acquise à partir d'une analyse qualitative du processus de dépollution.

2.4.1 Analyse qualitative du procédé

L'analyse qualitative du procédé est réalisée à partir de simulations en boucle ouverte sur le bioréacteur. Pour cela, nous appliquons différents couples d'entrée (u_1 , u_2) de type échelon et observons la dynamique des concentrations en biomasse et en substrats (voir la figure 2.4).

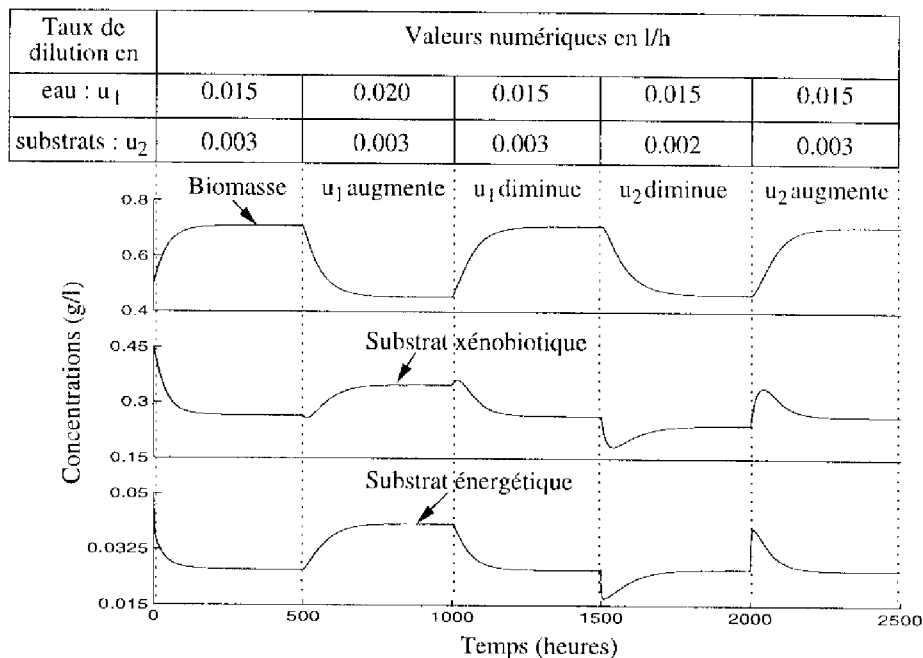


FIG. 2.4 - Simulations "tests" sur le bioréacteur

Les simulations représentées dans la figure 2.4 permettent de tirer les conclusions suivantes :

- Une augmentation (diminution) du taux de dilution en eau entraîne une diminution (augmentation) de la concentration en biomasse et une augmentation (diminution) de la concentration en substrats.
- Une diminution (augmentation) du taux de dilution en substrats entraîne une décroissance (croissance) des micro-organismes.
- L'effet premier de la diminution du taux de dilution en substrats sur les concentrations en substrats est une rapide diminution, suivie aussitôt par une augmentation plus progressive. Néanmoins, l'influence finale, une fois le régime permanent établi, est une diminution de leur concentration.

- Une augmentation du taux de dilution en substrats, entraîne, dans un premier temps, une augmentation très rapide des substrats, puis, dans un deuxième temps, ceux-ci diminuent jusqu'à atteindre un niveau de concentration stable supérieur à leur niveau avant l'application de cet échelon.

2.4.2 Fusion mathématique et hiérarchisation

Les variables en entrée du contrôleur flou sont :

$$\left\{ \begin{array}{l} \text{l'erreur de concentration en biomasse } \epsilon_c = c_c - c, \\ \text{la variation de l'erreur de concentration en biomasse } \Delta\epsilon_c, \\ \text{l'erreur de concentration en substrat xénobiotique } \epsilon_s = s_c - s, \\ \text{la variation de l'erreur de concentration en substrat xénobiotique } \Delta\epsilon_s. \end{array} \right.$$

Une fusion mathématique de l'erreur et de la variation de l'erreur, pour les concentrations en biomasse et en substrat xénobiotique, conduit aux deux variables suivantes :

$$\left\{ \begin{array}{l} X_c = K_{\epsilon_c} \cdot \epsilon_c + K_{\Delta\epsilon_c} \cdot \Delta\epsilon_c \\ X_s = K_{\epsilon_s} \cdot \epsilon_s + K_{\Delta\epsilon_s} \cdot \Delta\epsilon_s \end{array} \right. \quad (2.8)$$

X_c et X_s représentent respectivement, l'état des concentrations en biomasse et en substrat xénobiotique.

La stratégie de commande floue que nous adoptons s'appuie sur la configuration monovariante du bioréacteur, cas où la concentration en substrat xénobiotique $s(t)$ est régulée par le taux de dilution en effluent. Comme nous voulons aussi réguler la concentration en biomasse, une grandeur de commande supplémentaire est ajoutée : le taux de dilution en eau. Nous choisissons donc, dans un premier temps, de réguler les concentrations $c(t)$ et $s(t)$ par $u_1(t)$ et $u_2(t)$ respectivement. L'initialisation des contrôleurs flous est donc réalisée dans ce sens et, ce, à partir des observations issues de l'analyse qualitative précédente (voir tables 2.3 et 2.4). Cependant, u_1 ayant une influence non négligeable sur s , et u_2 agissant aussi sur c , il est nécessaire, afin de mieux compenser ces couplages, de calculer les taux de dilution, en substrats et en eau, en fonction des deux variables X_c et X_s . Dans le cas d'une structure hiérarchique, deux cas, représentés dans la figure 2.5, sont possibles.

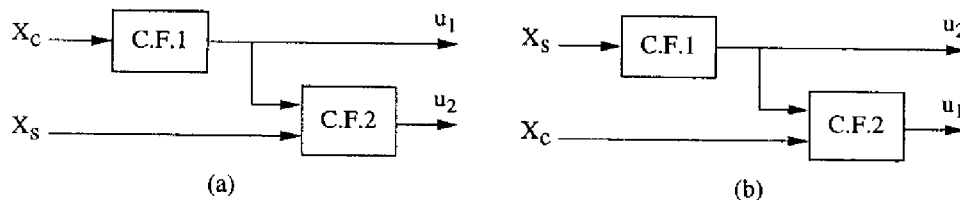


FIG. 2.5 – Deux structures hiérarchiques de commande floue sont envisageables.

La priorité étant donnée à la compensation des interactions sur le substrat xénobiotique, nous optons pour le premier cas de figure (a). La structure de commande floue correspondante est représentée plus en détail dans la figure 2.6. Le premier contrôleur flou C.F.1 considère l'état de la concentration en biomasse X_c pour calculer le taux de dilution en eau u_1 . Le second contrôleur flou, C.F.2, considère X_s et la sortie de C.F.1, pour ajuster le taux de dilution en substrats. Le fait que C.F.2 ait la sortie de C.F.1 comme entrée devrait permettre d'une part, de compenser les actions de u_1 sur le substrat xénobiotique, d'autre part de calculer u_2 en fonction de X_s mais aussi de X_c , puisque, à travers Δu_1 on a aussi des informations sur X_c .

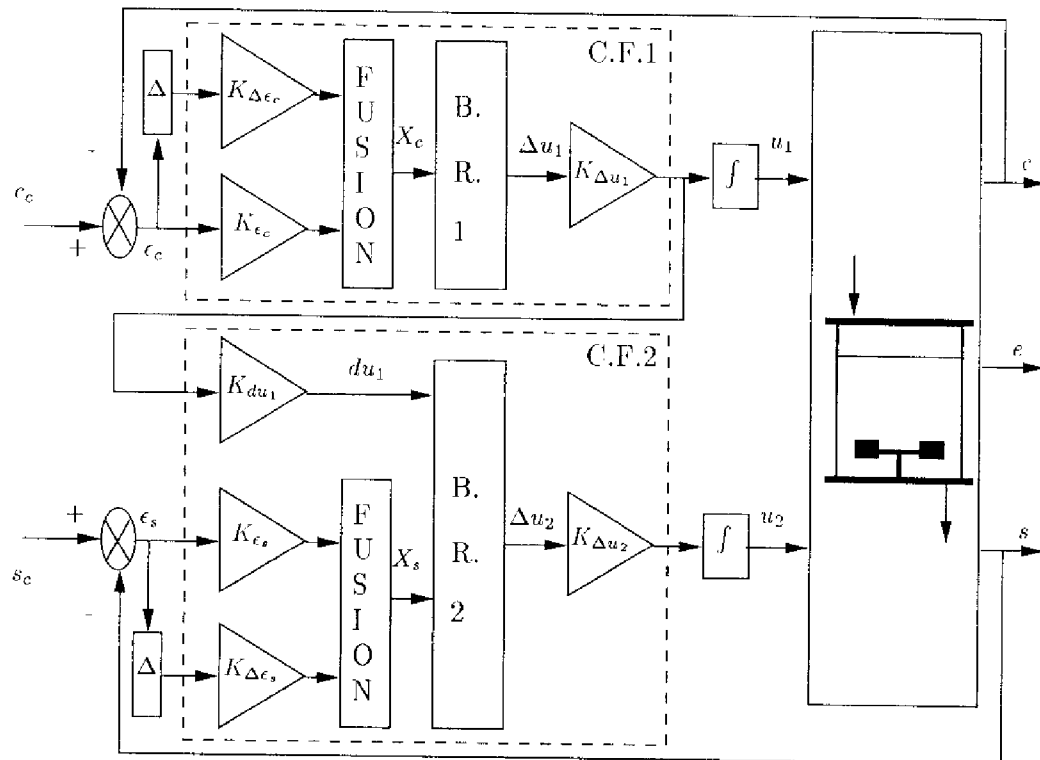


FIG. 2.6 – Structure de commande floue hiérarchique

2.4.3 Auto-réglage du contrôleur flou

La mise au point des contrôleurs flous est réalisée en minimisant le critère quadratique :

$$C = \sum_{t=t_{\text{début}}}^{t=t_{\text{fin}}} \left(\frac{1}{2} 100 [c(t) - c_d(t)]^2 + \frac{1}{2} 30 [s(t) - s_d(t)]^2 \right) \quad (2.9)$$

à l'aide d'une technique de descente du gradient.

Le jacobien du processus de dépollution $\begin{pmatrix} \frac{\partial c(t)}{\partial u_1(t)} & \frac{\partial c(t)}{\partial u_2(t)} \\ \frac{\partial s(t)}{\partial u_1(t)} & \frac{\partial s(t)}{\partial u_2(t)} \end{pmatrix}$ est grossièrement approximé par : *signe* $\begin{pmatrix} \frac{c(t) - c(t - T_e)}{u_1(t) - u_1(t - T_e)} & \frac{c(t) - c(t - T_e)}{u_2(t) - u_2(t - T_e)} \\ \frac{s(t) - s(t - T_e)}{u_1(t) - u_1(t - T_e)} & \frac{s(t) - s(t - T_e)}{u_2(t) - u_2(t - T_e)} \end{pmatrix}$.

Avant de procéder à l'auto-réglage du contrôleur flou, ce dernier est initialisé à partir de la connaissance disponible sur le procédé.

Initialisation

L'initialisation des gains en entrée des contrôleurs flous est réalisée à partir du modèle de référence (figure 2.3). En effet, la connaissance des valeurs maximales des consignes et de la variation maximale des concentrations en biomasse et en substrat xénobiotique conduit aux initialisations suivantes:

$$K_{c_s} = 6.66 \quad K_{\Delta c_s} = 34 \quad K_{c_c} = 10 \quad K_{\Delta c_c} = 71 \quad (2.10)$$

Quant au gain K_{du_1} , il est choisi égal à $\frac{1}{K_{\Delta u_1}}$ et normalise ainsi la première entrée du contrôleur flou C.F.2. Les gains en sortie sont initialisés comme suit :

$$K_{\Delta u_1} = 10.10^{-3} \quad K_{\Delta u_2} = 2,5.10^{-3} \quad (2.11)$$

Les ensembles flous sont uniformément répartis sur leur univers de discours.

Les bases de règles B.R.1 (voir table 2.3) et B.R.2 (voir table 2.4) sont initialisées de sorte que les concentrations en biomasse et en substrat xénobiotique soient respectivement asservies par u_1 , le taux de dilution en eau, et u_2 , le taux de dilution en substrats.

X_c	NM	NP	Z	PP	PM
Δu_1	1	0.5	0	-0.5	-1

TAB. 2.3 – Base de règles initiale associée au contrôleur flou C.F.1 $\rightarrow u_1$

du_1	N	Z	P
X_s	N	-1	-1
	Z	0	0
	P	1	1

TAB. 2.4 – Base de règles initiale associée au contrôleur flou C.F.2 $\rightarrow u_2$

Réglage des gains

Les facteurs d'échelle sont les premiers paramètres du contrôleur flou à être réglés. Leur évolution au cours de leur phase de réglage est donnée dans le tableau 2.5. Nous rappelons que le gain K_{du_1} est mis à jour par $K_{du_1} = \frac{1}{K_{\Delta u_1}}$ normalisant ainsi la première entrée du contrôleur flou C.F.2.

Gains	C.F.1			C.F.2			
	K_{ϵ_c}	$K_{\Delta\epsilon_c}$	$K_{\Delta u_1} \cdot 10^{-3}$	K_{du_1}	K_{ϵ_s}	$K_{\Delta\epsilon_s}$	$K_{\Delta u_2} \cdot 10^{-3}$
initiaux	10	71	10	100	6.66	34	2.5
après réglage	9.78	71.23	36.3	27.54	7.11	33.92	22.9

TAB. 2.5 - Evolution des facteurs d'échelle au cours de la première phase de l'algorithme

Optimisation des fonctions d'appartenance

Au cours de la seconde et dernière phase de l'algorithme, les ensembles flous en entrée et en sortie (singletons) sont optimisés. Leur optimisation peut conduire à une dénormalisation de leur univers de discours et par conséquent les facteurs d'échelle sont eux aussi mis à jour. La valeur des gains après optimisation est donnée dans le tableau 2.6.

Gains	C.F.1			C.F.2			
	K_{ϵ_c}	$K_{\Delta\epsilon_c}$	$K_{\Delta u_1} \cdot 10^{-3}$	K_{du_1}	K_{ϵ_s}	$K_{\Delta\epsilon_s}$	$K_{\Delta u_2} \cdot 10^{-3}$
après optimisation	11.08	80.75	36.8	27.54	6.60	31.47	25.3

TAB. 2.6 - Valeur numérique des facteurs d'échelle après optimisation des ensembles flous

Les fonctions d'appartenance en entrée des contrôleurs flous C.F.1 et C.F.2, après optimisation, sont représentées dans la figure 2.7.

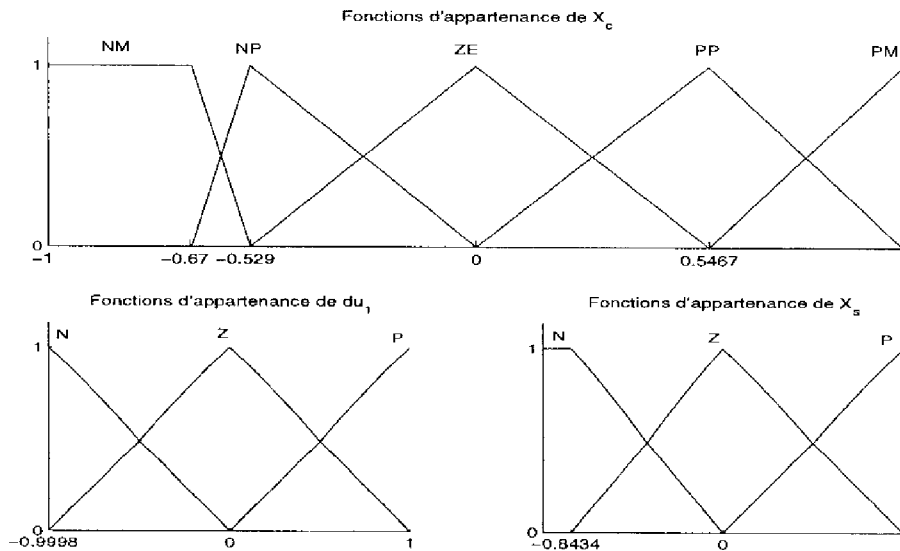


FIG. 2.7 - Fonctions d'appartenance après optimisation

Les singletons flous dans la partie conclusion des règles des deux contrôleurs C.F.1 et C.F.2, après optimisation, sont respectivement représentés dans les tables 2.7 et 2.8.

X_c	NM	NP	Z	PP	PM
Δu_1	0.98	0.61	0	-0.57	-1

TAB. 2.7 - Base de règles associée au contrôleur flou C.F.1 $\rightarrow u_1$.

du_1		N	Z	P
X_s	N	-0.905	-1	-0.909
	Z	-0.018	0	0.009
	P	0.904	0.910	0.905

TAB. 2.8 - Base de règles associée au contrôleur flou C.F.2 $\rightarrow u_2$.

2.4.4 Analyse des résultats

L'évolution du critère quadratique C (2.9), au cours de la phase de réglage des gains et au cours de la phase d'optimisation des ensembles flous, est représentée dans la figure 2.8.

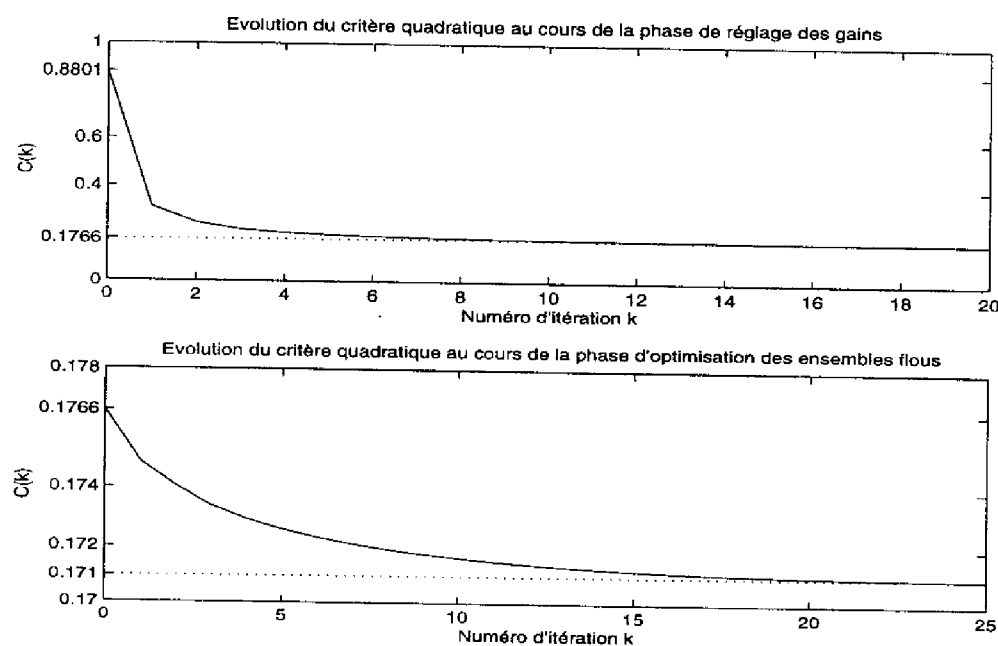


FIG. 2.8 - Evolution du critère quadratique

Nous remarquons, que la phase d'optimisation des ensembles flous conduit à une très légère diminution du critère quadratique. Le suivi des trajectoires de référence, après réglage des gains, est déjà très satisfaisant et est représenté dans les figures 2.9 et 2.10.

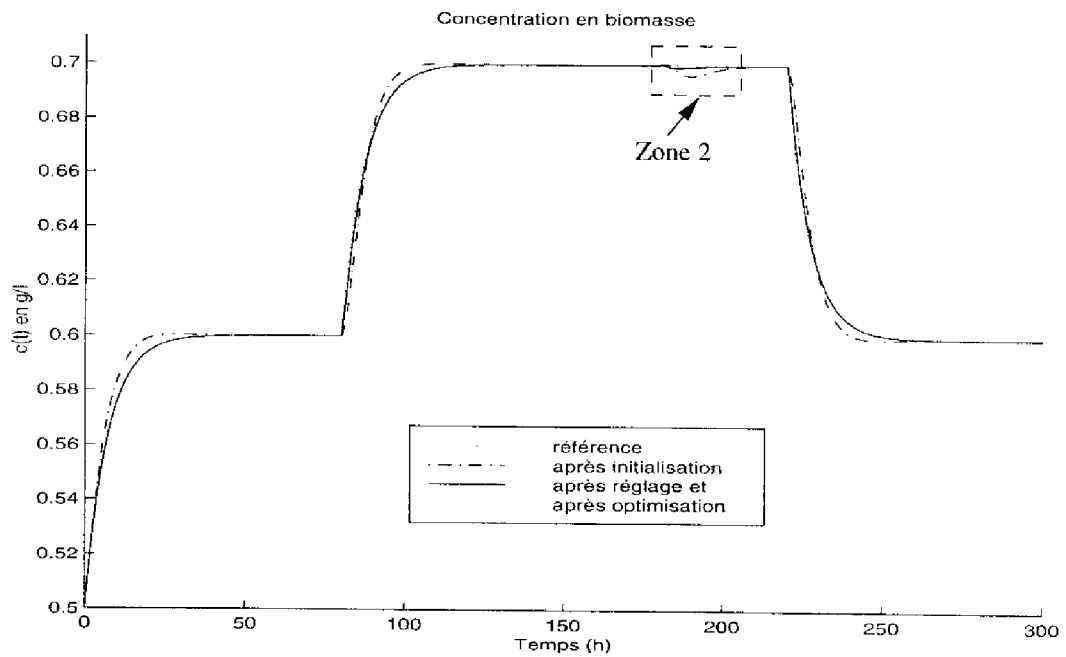


FIG. 2.9 – Suivi de la trajectoire de référence pour la concentration en biomasse

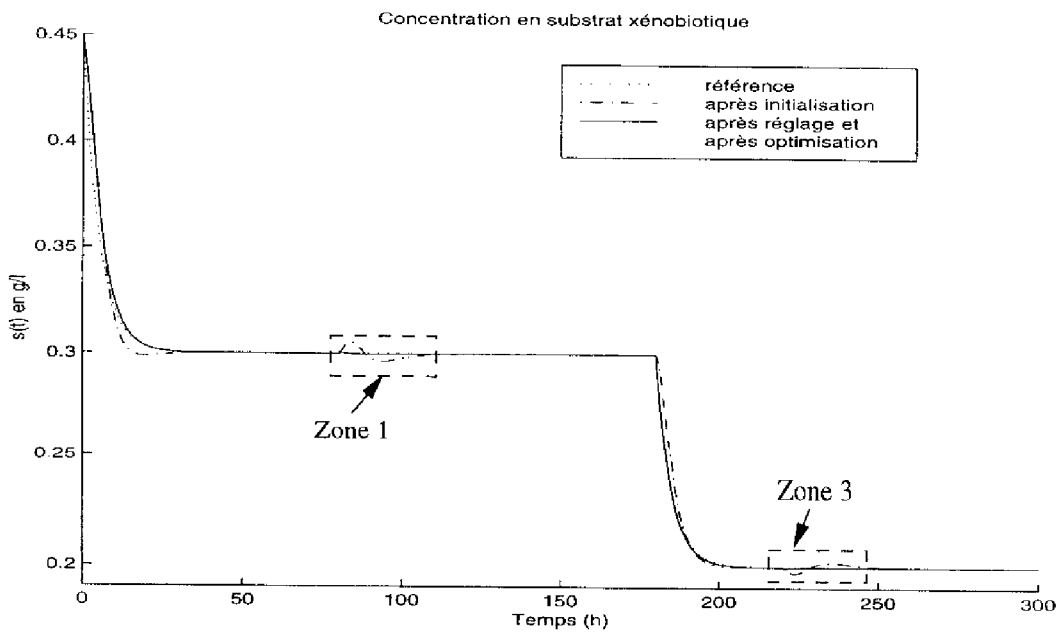


FIG. 2.10 – Suivi de la trajectoire de référence pour la concentration en substrat xénobiotique

Les trajectoires de référence, pour la concentration en biomasse (voir figure 2.9), et pour la concentration en substrat xénobiotique (voir figure 2.10), sont parfaitement suivies et un "zoom" des zones 1,2 et 3 permet de mieux observer la bonne compensation des interactions (voir figure 2.11).

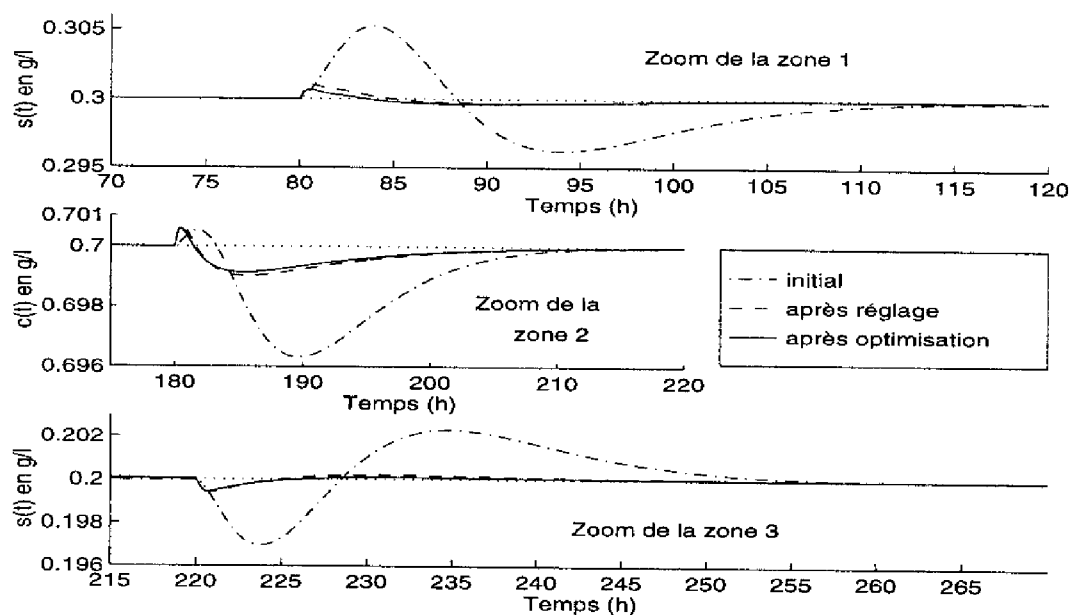


FIG. 2.11 – Les interactions sont mieux compensées après la phase d'optimisation.

Les taux de dilution, en eau et en substrats, sont représentés dans la figure 2.12.

2.4.5 Analyse expérimentale de la robustesse du contrôleur flou hiérarchique

Sans modifier le précédent contrôleur flou, la robustesse de ce dernier est testée en simulant des perturbations abruptes sur les paramètres réels du procédé. Ces variations paramétriques sont répertoriées dans le tableau 2.9.

Description	Paramètres	Valeurs initiales	Variations
Taux maximum de croissance (1/h)	μ_{sm}	0.1	+30% à t=130 h
	μ_{em}	0.2	+30% à t=70 h
Constantes de Michaelis-Menten (g/l)	K_s	1.5	+30% à t=240 h
	K_e	1	+30% à t=110 h
Paramètres d'inhibition/activation	a_s	0.1	+30% à t=50 h
	a_e	10	+30% à t=180 h

TAB. 2.9 – Variation des paramètres cinétiques réels du procédé

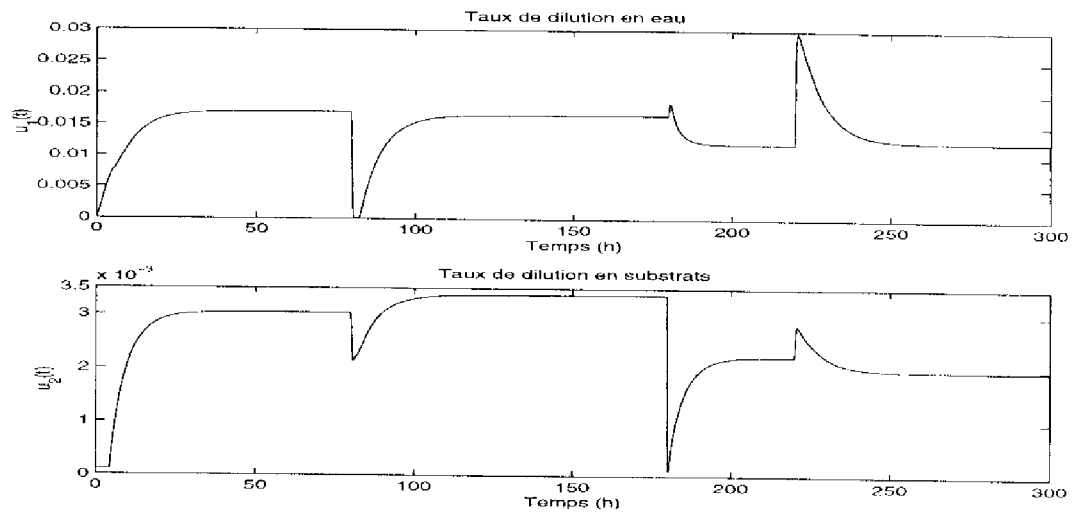


FIG. 2.12 – Evolution des taux de dilution en eau et en substrats.

Les simulations ont montrées que les perturbations sur les paramètres cinétiques réels du procédé n'altèrent pas les performances du contrôleur flou et les trajectoires de référence sont toujours parfaitement suivies. En effet, les réponses du système en boucle fermée, avec et sans perturbations sur les paramètres cinétiques, sont confondues (les courbes correspondantes ne sont donc pas représentées et l'on peut se référer aux figures 2.9 et 2.10). Exactement comme dans le cas de paramètres cinétiques fixes, les interactions sur la concentration en biomasse et la concentration en substrat xénobiotique sont très bien compensées (voir figure 2.13).

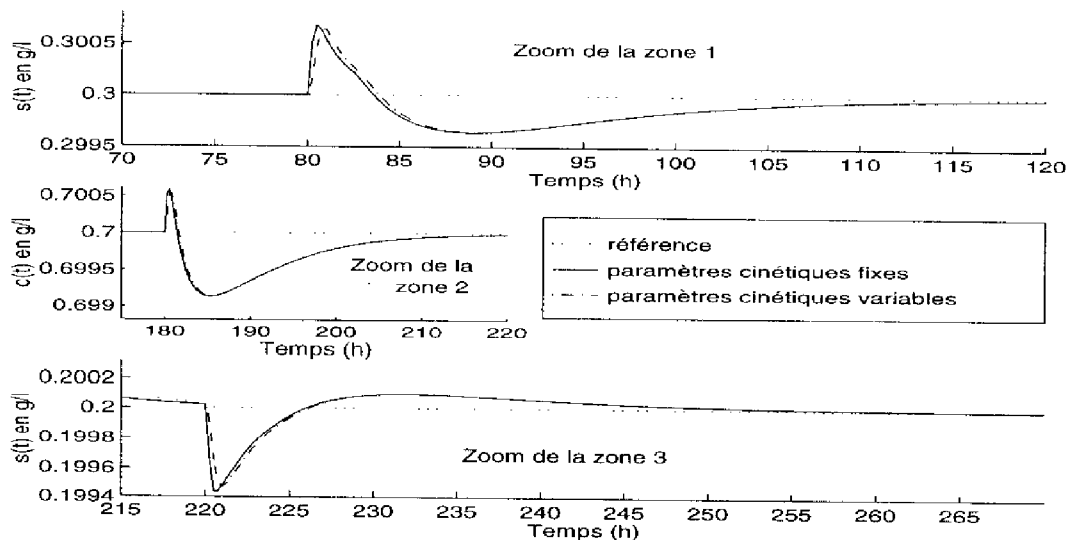


FIG. 2.13 – Les interactions sont très bien compensées.

2.5 Conclusion

Dans ce chapitre, nous avons appliqué la méthodologie résumée dans le dernier chapitre de la partie II à un problème d'actualité : le traitement biologique des eaux usées.

Après une analyse qualitative du procédé de dépollution, une stratégie de commande floue, basée sur les techniques de réduction par fusion mathématique et hiérarchisation, est décidée. Les performances du système en boucle fermée auxquelles nous prétendons sont décrites à travers un modèle de référence mathématique. Les facteurs d'échelle en entrée du contrôleur flou sont initialisés à partir des trajectoires de référence issues du précédent modèle. Quant aux règles de commande floue, elles s'écrivent aisément dès lors qu'un minimum de connaissance sur le procédé est disponible, à savoir, le sens d'évolution des sorties en fonction du sens de variation des commandes appliquées au système.

La mise au point finale du contrôleur flou (réglage des gains puis optimisation des fonctions d'appartenance) est réalisée à travers la poursuite de trajectoires de référence, et ce, à l'aide d'une technique, très simple, de descente du gradient.

Le contrôleur flou hiérarchique ici adopté contient seulement 14 règles de commande et le cahier des charges est parfaitement respecté. La robustesse de la stratégie de commande floue vis à vis des variations paramétriques du modèle du procédé a été vérifiée.

L'application de la méthodologie suivante :

1. réduction de la dimension du problème de commande,
2. structuration de la loi de commande,
3. mise au point du contrôleur flou à l'aide d'une technique du gradient,

à un procédé biologique d'épuration vient confirmer son intérêt : le nombre de règles est en effet considérablement réduit. Il serait maintenant fort intéressant d'appliquer cette méthodologie à des procédés de plus grande dimension.

Conclusion

Nous nous sommes attachée, dans ce mémoire, au problème de l'explosion combinatoire du nombre de règles d'un contrôleur flou. L'augmentation exponentielle du nombre de règles en fonction du nombre de variables d'entrée est en effet très pénalisante pour l'application des techniques floues. Cette explosion ne facilite pas la synthèse d'un contrôleur flou et, une fois ce dernier défini, des problèmes liés à sa mise en œuvre pratique peuvent apparaître : les temps de calcul peuvent être prohibitifs et l'espace mémoire requis important. Cette thèse s'inscrit donc dans la continuité des travaux existants et propose des solutions visant à simplifier la synthèse d'un contrôleur flou.

Dans la **première partie** de ce manuscrit nous avons introduit les quelques notions de base de la logique floue et de la commande floue, notions que nous avons jugées indispensables à la bonne compréhension de cette thèse.

Dans la **seconde partie** de ce mémoire nous nous sommes intéressée à la réduction de la complexité des contrôleurs flous et à leur mise au point automatique.

Dans le *premier chapitre*, nous avons dressé le bilan des travaux réalisés en commande floue multivariable et introduit la problématique et les objectifs de cette thèse, à savoir, la réduction de la base de règles et un outil d'aide à la conception de contrôleurs flous multivariables à travers des techniques d'apprentissage.

Nous avons vu, dans le *second chapitre*, que lorsque la base de règles existe déjà, on dispose de méthodes assez intéressantes pour la réduire. Nous avons classé ces différentes approches selon deux grands groupes. Le premier groupe consiste à éliminer ou à sélectionner certaines règles. Le second groupe d'approches consiste à diminuer le nombre de labels linguistiques. Nous ne nous sommes pas étendus outre mesure sur ces techniques, étant donné qu'on ne dispose généralement pas de cette base de règles. De plus, il semble plus logique, face au problème que nous nous sommes fixé, de synthétiser directement un contrôleur flou de complexité réduite.

Aussi, dans le *troisième chapitre*, nous avons présenté diverses méthodes permettant de définir un contrôleur flou contenant un nombre limité de règles. Ces techniques ont toutes été illustrées sur le problème bien connu du pendule inversé.

La fusion mathématique ou symbolique des variables d'entrée du contrôleur flou permet de réduire la dimension du problème de commande. La fusion mathématique consiste à combiner des variables mathématiquement avant de les fournir en entrée

du contrôleur flou. La fusion symbolique des variables d'entrée repose sur l'utilisation d'ensembles flous multidimensionnels. Ces ensembles flous sont construits à partir d'un maillage de Delaunay. L'application de ces méthodes diminue le nombre de variables traitées en entrée du contrôleur flou et, par voie de conséquence, le nombre de règles. Cependant, ces variables ne peuvent pas être combinées n'importe comment et leur association doit garder un sens physique. Pour des problèmes de régulation d'ordre n , et pour chacune des variables régulées, nous proposons, par analogie avec la commande en mode glissant, de combiner l'erreur (différence entre une variable et sa consigne) et ses $(n-1)$ dérivées successives.

Une autre voie intéressante pour réduire la complexité d'un contrôleur flou consiste à structurer la loi de commande: décentralisation, hiérarchisation. Cette structuration permet de décomposer le problème de commande global en sous-problèmes plus simples à résoudre, ce qui facilite considérablement l'écriture de la base de règles. Parmi les techniques de hiérarchisation citées, contrôleur flou hiérarchique de Raju et base de règles multi-niveaux, nous nous sommes particulièrement intéressée à la première. Le contrôleur flou hiérarchique de Raju conduit à une augmentation linéaire et non plus exponentielle du nombre de règles en fonction du nombre de variables en entrée. Ce nombre de règles atteint sa valeur minimale si, dans chacun des niveaux successifs, une et une seule variable supplémentaire est considérée. C'est pourquoi les contrôleurs flous utilisés dans ce mémoire ont au maximum deux entrées. Bien qu'aucune règle générale ne permette de hiérarchiser le traitement des variables, pour des problèmes de commande, il nous semble assez naturel de placer les variables les plus "importantes" en bas de la hiérarchie, le dernier contrôleur flou formant la chaîne la plus directe. Par exemple, pour le problème du pendule inversé, les variables sont considérées selon les dynamiques croissantes.

Une combinaison des techniques ci-dessus citées, réduction de la dimension et structuration de la loi de commande, est également possible.

Dans le *quatrième chapitre*, nous avons présenté la procédure d'auto-réglage du contrôleur flou. En effet, si l'application des précédentes méthodes de réduction permet une réduction considérable du nombre de paramètres à régler, la mise au point des contrôleurs flous par réglages successifs peut être assez fastidieuse. Il nous est donc apparu fort intéressant de combiner les méthodes de réduction de la base de règles (réduction de la dimension, structuration de la loi de commande) à des techniques d'apprentissage classiques. Pour sa simplicité de mise en œuvre, nous sommes limitée à la technique de descente du gradient.

L'auto-réglage du contrôleur flou est basé sur une structure de commande floue par modèle de référence. Les contrôleurs flous considérés sont des systèmes flous de type Takagi-Sugeno d'ordre zéro. L'opérateur de conjonction est le produit et les ensembles flous en entrée sont, soit des triangles, soit des ensembles flous bi-dimensionnels construits à partir d'une triangulation de Delaunay. Ces ensembles forment une partition floue, ce qui, d'une part, permet à chaque label de conserver une signification sémantique, d'autre part, assure la complétude de la base de

règles. Dans un premier temps, les contrôleurs flous sont initialisés en tirant parti de la connaissance disponible sur le procédé à commander. Cette étape est importante puisque la méthode du gradient n'aboutit pas forcément au minimum global du critère que nous cherchons à minimiser. La mise au point du contrôleur flou est ensuite réalisée à travers la minimisation de l'erreur entre la sortie d'un modèle de référence (traduisant les objectifs de commande) et la sortie du processus. Une première étape consiste à régler les gains entrées/sorties du contrôleur, ces derniers ayant une action prédominante sur la valeur de commande. La seconde étape consiste à optimiser les ensembles flous en entrée et en sortie (singletons flous). C'est une approche de précision et elle a un effet plus local sur la commande. Le problème de l'optimisation d'ensembles flous triangulaires a été étendu à des ensembles flous bidimensionnels construits à partir d'une triangulation de Delaunay. L'extension à des ensembles flous de dimension supérieure ne nous semble pas évidente et leur utilisation présente un inconvénient majeur : leur gourmandise en temps calcul.

Dans le *cinquième chapitre*, nous avons résumé les grandes étapes de la méthodologie proposée dans ce mémoire :

1. réduction de la dimension du problème de commande,
2. structuration de la loi de commande,
3. mise au point du contrôleur flou à l'aide d'une technique du gradient.

Nous avons consacré la **troisième et dernière partie** de ce manuscrit à l'application de la précédente méthodologie à la commande de deux procédés multivariables. Dans le *premier chapitre*, cette démarche a été appliquée sur un exemple académique, un mélangeur multivariable à deux entrées et deux sorties. Dans le *second chapitre*, le procédé étudié est un processus biologique de traitement des eaux usées. Ce procédé est l'un des benchmarks étudiés dans le cadre du projet européen FAMIMO "Fuzzy Algorithms for the control of Multi-Input Multi-Output processes". Dans les deux cas, la structure du contrôleur flou est définie à partir d'une analyse qualitative du processus à commander. Les résultats obtenus sont très satisfaisants et la robustesse des stratégies de commande floue a pu être vérifiée expérimentalement.

Cette partie applicative nous conforte dans notre optimisme vis à vis de telles techniques et une prochaine étape serait leur application à des systèmes de plus grande dimension. Une étude théorique des conditions de convergence de l'algorithme d'auto-réglage s'inscrit également dans la mouvance de ces travaux. Dans ce mémoire, nous nous sommes limitée à une adaptation hors-ligne des paramètres du contrôleur flou. Cependant, lors de la commande de systèmes à dynamiques fortement non linéaires et variant dans le temps, les performances du système en boucle fermée peuvent se dégrader. Il serait donc fort intéressant d'envisager une couche de supervision floue, laquelle, à partir d'une mesure en-ligne des performances, déclencherait, si nécessaire, une nouvelle adaptation des paramètres. Cette mesure des

performances, pourrait être, par exemple, la valeur numérique du critère quadratique précédemment défini pour l'auto-réglage du contrôleur flou.

Pour conclure, nous pensons que l'utilisation de la logique floue en commande constitue une alternative intéressante aux techniques de l'automatique classique, lorsque le modèle mathématique du procédé n'est pas disponible ou lorsque celui-ci présente de trop fortes non-linéarités. La commande floue permet en effet de s'affranchir d'un modèle mathématique du processus à commander et son point fort réside surtout dans sa capacité à traduire des connaissances expertes ou qualitatives sur le procédé en règles de commandes floues. Cependant, cette approche souffrait du "fléau de la dimension". Par une structuration de la commande et une fusion (mathématique ou symbolique) des variables, alliées à une procédure d'optimisation, nous avons contribué à limiter cet inconvénient de la commande floue.

Références bibliographiques

- [Ami92] Amira. *Laboratory setup inverted pendulum LIP100*. Amira GmbH, Duisburg, Allemagne, 1992.
- [AW95] K.J. Astrom et B. Wittenmark. *Adaptive Control*. Addison-Wesley, seconde édition, 1995.
- [Bav96] I.M. Bavelaar. Fuzzy Model Inversion. Rapport LAAS 96475, Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS, Décembre 1996.
- [Ben93] E. Benoit. *Capteurs symboliques et capteurs flous : un nouveau pas vers l'intelligence*. Thèse de doctorat, Université Joseph Fourier, Grenoble, Janvier 1993.
- [BJPB93] H. Bersini, J-P.Nordvik, et A. Bonarini. A Simple Direct Adaptive Fuzzy Controller Derived from its Neural Equivalent. Dans *Second IEEE International Conference on Fuzzy Systems*, pages 345-350, San Francisco, Californie, 28 Mars - 1 Avril 1993.
- [BJT95] J. Bruinzeel, M. Jamshidi, et A. Titli. A sensory fusion-hierarchical real-time fuzzy control approach for complex systems. Rapport LAAS 95347, Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS, Toulouse, Août 1995.
- [BK95] R. Babuska et U. Kaymak. Application of Compatible Cluster Merging to Fuzzy Modelling of Multivariable Systems. Dans *EUFIT'95, Third European Congress on Intelligent Techniques and Soft Computing*, pages 565-569, Aachen, Allemagne, Août 1995.
- [BLTV96] J. Bruinzeel, V. Lacrose, A. Titli, et H.B. Verbruggen. Real-time Fuzzy Control of Complex Systems using Rule-base Reduction Methods. Dans *WAC'96, Second World Automation Congress*, volume 5, pages 27-30, Montpellier, France, 27-30 Mai 1996.

- [BO94] G. Bretthauer et H.-P. Opitz. Stability of Fuzzy Systems - A Survey. Dans *EUFIT'94, Second European Congress on Intelligent Techniques and Soft Computing*, pages 283-290, Aachen, Allemagne, Septembre 1994.
- [Boh94] R. Bohm. Analysis of Fuzzy Controlled Systems using Hyperstability Theory. Dans *Third IEEE International Conference on Fuzzy Systems*, pages 1464-1467, Orlando, Floride. 26-29 Juin 1994.
- [BSV95] R. Babuska, J. Sousa, et H.B. Verbruggen. Model-Based Design of Fuzzy Control Systems. Dans *EUFIT'95, Third European Congress on Intelligent Techniques and Soft Computing*, pages 837-841, Aachen, Allemagne, Août 1995.
- [CC91] C.-L. Chen et P.-C. Chen. Application of Fuzzy Logic Controllers in Single-Loop Tuning of Multivariable System Design. *Computer in Industry*, 17:33-41, 1991.
- [DHR93] D. Driankov, H. Hellendoorn, et M. Reinfrank. *An Introduction to Fuzzy Control*. Springer-Verlag Berlin Heidelberg, 1993.
- [Dre94a] M.E. Dreier. A Fast, Non-iterative Method to Generate Fuzzy Inference Rules from Observed Data. Rapport technique, Bell Helicopter Textron Inc., Texas, U.S.A., 1994.
- [Dre94b] M.E. Dreier. *User's manual for FUZZY Logic DEVELOPMENT Kit*. Bell Helicopter Textron Inc., Texas, U.S.A., 1994.
- [FBM93] L. Foulloy, E. Benoit, et G. Mauris. Applications of Fuzzy Sensors. Dans *European Workshop on Industrial Fuzzy Control and Applications*, Terrassa, Espagne, Avril 1993.
- [FG92] L. Foulloy et S. Galichet. Représentation des contrôleurs flous. Dans *Actes des 2^{èmes} journées nationales sur les applications des ensembles flous*, Nîmes, France, Novembre 1992.
- [FGB94] L. Foulloy, S. Galichet, et E. Benoit. Fuzzy control with fuzzy state sensors. Dans *EUFIT'94, Second European Congress on Intelligent Techniques and Soft Computing*, pages 1156-1160, Aachen, Allemagne, Septembre 1994.
- [GH89] P. L. George et F. Hermeline. Maillage de Delaunay d'un polyèdre convexe en dimension d. Extension à un polyèdre quelconque. Rapport de recherche 969, INRIA, Février 1989.

- [GS94] F. Guely et P. Siarry. A Centred Formulation of Takagi-Sugeno Rules for Improved Learning Efficiency. *Fuzzy Sets and Systems*, 62:277–285, 1994.
- [HM94] J. Hohensohn et J. M. Mendel. Two-Pass Orthogonal Least-Squares Algorithm to Train and Reduce Fuzzy Logic Systems. Dans *Proceedings of the Third IEEE International Conference on Fuzzy Systems*, pages 696–700, Orlando, Floride, 26-29 Juin 1994.
- [Jag95] R. Jager. *Fuzzy Logic in Control*. Thèse de doctorat, Université Technique de Delft, Pays-Bas, 1995.
- [Jam96] M. Jamshidi. *Large-scale systems modelling, Control and Fuzzy Logic*. Prentice-Hall, Englewood Cliffs, U.S.A., 1996.
- [JGBF95] J.F. Jossierand, S. Galichet, E. Benoit, et L. Foulloy. Fuzzy cells network for fuzzy control with a fuzzy state sensor. Dans *EUFIT'95, Third European Congress on Intelligent Techniques and Soft Computing*, pages 910–914, Aachen, Allemagne, Août 1995.
- [JL93] C. Jianqin et C. Laijiu. Study on Stability of Fuzzy Closed-loop Systems. *Fuzzy Sets and Systems*, 57:159–168, 1993.
- [JTV89] D. Jaume, S. Thelliez, et M. Verge. *Applications du formalisme d'état à la commande des systèmes continus*. Eyrolles, 1989.
- [KGN85] J.B. Kiszka, M.M. Gupta, et P.N. Nikiforuk. Energetic Stability of Fuzzy Dynamic Systems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15:783–792, 1985.
- [KH93a] L. T. Kóczy et K. Hirota. Interpolative Reasoning with Insufficient Evidence in Sparse Fuzzy Rule Bases. *Information Sciences*, 71:169–201, 1993.
- [KH93b] L. T. Kóczy et K. Hirota. Modular Rule Bases in Fuzzy Control. Dans *EUFIT'93-First European Congress on Fuzzy and Intelligent Technologies*, pages 606–610, Aachen, Allemagne, Septembre 1993.
- [LBT96] V. Lacrose, J. Bruinzeel, et A. Titli. A Preliminary Study of some Rule-base Reduction Methods. Dans *EUFIT'96, Fourth European Congress on Intelligent Techniques and Soft Computing*, volume 2, pages 1046–1050, Aachen, Allemagne, 2-5 Septembre 1996.
- [Lee90] C.C. Lee. Fuzzy Logic in Control Systems: Fuzzy Logic Controller - Part 1 & 2. *IEEE Transactions on Systems, Man and Cybernetics*, 20(2):404–435, 1990.

- [LN93a] D.A. Linkens et J. Nie. Constructing Rule-Bases for Multivariable Fuzzy Control by Self-Learning. Part 1. System Structure and Learning Algorithms. *International Journal of Systems and Science*, 24(1):111-127, 1993.
- [LN93b] D.A. Linkens et J. Nie. Constructing Rule-Bases for Multivariable Fuzzy Control by Self-Learning, Part 2. Rule-Base Formation and Blood Pressure Control Application. *International Journal of Systems and Science*, 24(1):129-157, 1993.
- [LT] V. Lacrose et A. Titli. Dans le livre *FALCON "Fuzzy Algorithms For Control Engineering"*, Chapitre 21, Fuzzy Control of Multivariable Complex Systems using Rule Base Reduction Methods. Kluwer Academic Publishers. A paraître.
- [LT96a] V. Lacrose et A. Titli. Méthodes pour réduire la complexité d'un contrôleur flou. *Groupe de Recherche en Automatique du C.N.R.S., Pôle "Automatisation Intégrée", Projet "Commande Symbolique et Neuromimétique"*, Paris, France, 15 Février 1996.
- [LT96b] V. Lacrose et A. Titli. Multivariable Fuzzy Control using Rule-base Reduction and Self-tuning. Dans *TEMPUS Symposium on Qualitative System Modelling, Qualitative Fault Diagnosis and Fuzzy Logic and Control*, pages 172-177, Budapest et Balatonfured, Hongrie, 17-20 Avril 1996.
- [LT97a] V. Lacrose et A. Titli. Fusion and Hierarchy Can Help Fuzzy Logic Controller Designers. Dans *FUZZ-IEEE'97, Sixth IEEE International Conference on Fuzzy Systems*, pages 1159-1165, Barcelone, Catalogne-Espagne, 1-5 Juillet 1997.
- [LT97b] V. Lacrose et A. Titli. Fuzzy Control of an Inverted Pendulum using Sensory Fusion and Hierarchy. Dans *NASA URC-TC'97, NASA University Research Centers, Technical Advances in Education, Aeronautics, Space, Autonomy, Earth and Environment*, volume 1, pages 459-464, Albuquerque, Nouveau Mexique, USA, 16-19 Février 1997.
- [LT97c] V. Lacrose et A. Titli. Multivariable Fuzzy Control using Bi-dimensional Fuzzy Sets - Application to a Mixing Tank. Dans *EUFIT'97, Fifth European Congress on Intelligent Techniques and Soft Computing*, Aachen, Allemagne, 8-12 Septembre 1997.
- [LTM] V. Lacrose, A. Titli, et M. Jainshidi. Fusion and Hierarchy: Towards an Easier Fuzzy Logic Controller Design. *Journal of Intelligent and Fuzzy Systems*. Soumis.

- [MA75] E.H. Mamdani et S. Assilian. An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. *International Journal of Man-Machine Studies*, 7:1-13, 1975.
- [MT94] J.P. Marin et A. Titli. Comparative Analysis of Stability Methods for Fuzzy Controllers. Dans *EUFIT'94, Second European Congress on Intelligent Techniques and Soft Computing*, pages 1183-1187, Aachen, Allemagne, Septembre 1994.
- [MT97] J.P. Marin et A. Titli. Fuzzy Stability Analysis of Fuzzy Systems. Part 2: Fuzzyfication of Lyapunov Theory and Application. Dans *EUFIT'97, Fifth European Congress on Intelligent Techniques and Soft Computing*, Aachen, Allemagne, Septembre 1997.
- [MV94] C. Melin et B. Vidolof. Passive Two Rule Based Fuzzy Logic Controller. Dans *Third IEEE International Conference on Fuzzy Systems*, pages 947-951, Orlando, Floride, 26-29 Juin 1994.
- [NHN91] H. Nomura, I. Hayashi, et Wakami N. A Self-tuning Method of Fuzzy Control by Descent Method. Dans *Proceedings of 4th IFSA Congress, Engineering*, pages 155-158, 1991.
- [OFT94] OFTA. *Logique Floue*. Rapport de l'Observatoire Français des Techniques Avancées, Arago 14, Masson, Paris, 1994.
- [Opi93] H.P. Opitz. Fuzzy Control and Stability Criteria. Dans *EUFIT'93-First European Congress on Fuzzy and Intelligent Technologies*, pages 130-136, Aachen, Allemagne, Septembre 1993.
- [Pal92] R. Palm. Sliding Mode Fuzzy Control. Dans *First IEEE International Conference on Fuzzy Systems*, pages 519-526, San Diego, USA, 8-12 Mars 1992.
- [Ray95] C. Raymond. *Commande floue monovariante et multivariante selon l'expertise et par modélisation floue*. Thèse de doctorat, Institut National des Sciences Appliquées de Toulouse, Septembre 1995.
- [RBT94] C. Raymond, S. Boverie, et A. Titli. First Evaluation of Fuzzy MIMO Control Laws. Dans *Third IEEE International Conference on Fuzzy Systems*, pages 545-548, Orlando, Floride, 26-29 Juin 1994.
- [RM84] K.S. Ray et D.D. Majumder. Application of the Circle Criteria for Stability Analysis of Linear SISO and MIMO Systems Associated with Fuzzy Logic Controller. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-14(2):345-349, 1984.

- [RZ90] G.V.S. Raju et Jun Zhou. Fuzzy Logic Process Controller. Dans *IEEE International Conference on Systems Engineering*, pages 145-147. Pittsburgh, USA, Août 1990.
- [RZK91] G.V.S. Raju, J. Zhou, et R.A. Kisner. Hierarchical Fuzzy Control. *International Journal of Control*, 54(5):1201-1216, 1991.
- [SBLD82] A. Shakouri, P.P.J. van den Bosch, H.R. van Nauta Lemke, et J.G. Dijkman. Fuzzy Control for Multivariable Systems. *IEAC Computer Modern Design*, 1982.
- [Set95] M. Setnes. Fuzzy Rule-Base Simplification Using Similarity Measures. Mastère, Université Technique de Delft, Pays-Bas, Juillet 1995.
- [SGB93] M. Sugeno, M.F. Griffin, et A. Bastian. Fuzzy Hierarchical Control of an Unmanned Helicopter. Dans *Fifth IFSA World Congress*, pages 179-182, Seoul, 1993.
- [SK86] M. Sugeno et G.T. Kang. Fuzzy Modelling and Control of Multilayer Incinerator. *Fuzzy Sets and Systems*, 18:329-346, 1986.
- [SN85] M. Sugeno et M. Nishida. Fuzzy Control of Model Car. *Fuzzy Sets and Systems*, 16:103-113, 1985.
- [Sug85] M. Sugeno. An Introductory Survey of Fuzzy Control. *Information Sciences*, (36):59-83, 1985.
- [TM87] K.L. Tang et R.J. Mulholland. Comparing Fuzzy Logic with Classical Controller Design. *IEEE Transactions on Systems, Man and Cybernetics*, SMC 17(6):1085-1087, 1987.
- [Tou78] R.M. Tong. Analysis and Control of Fuzzy Systems Using Finite-state Relations. *International Journal of Control*, 27(3):131-140, 1978.
- [Tou80] R.M. Tong. Some Properties of Fuzzy Feedback Systems. *IEEE Transactions on Systems, Man and Cybernetics*, SMC 10(6):327-330, 1980.
- [TS85] F. Takagi et M. Sugeno. Fuzzy Identification of Systems and its Applications to Modeling and Control. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC 15(1):116-132, Janvier-Février 1985.
- [TS92] K. Tanaka et M. Sugeno. Stability Analysis and Design of Fuzzy Control Systems. *Fuzzy Sets and Systems*, 45:135-156, 1992.
- [TW85] M. Togai et P.P. Wang. Analysis of a Fuzzy Dynamic System and Synthesis of its controller. *International Journal of Man-Machines Studies*, 22:355-363, 1985.

- [Vid93] M. Vidyasagar. *Nonlinear Systems Analysis*. Prentice Hall, Englewood Cliffs, New Jersey, seconde édition, 1993.
- [VK93] P. Viljamaa et H.N. Koivo. Tuning of Multivariable Fuzzy Logic Controller. Dans *2nd IEEE International Conference on Fuzzy Systems*, pages 697-701, San Francisco, Mars 1993.
- [VK94] P. Viljamaa et H.N. Koivo. Tuning of a Multivariable PI-like Fuzzy Logic Controller. Dans *Third IEEE International Conference on Fuzzy Systems*, pages 388-393, Orlando, Floride, 26-29 Juin 1994.
- [VW76] Mac Vicar-Whelan. Fuzzy Sets for Man-Machine Interaction. *International Journal of Man-Machines Studies*, (8):687-697, 1976.
- [Wan94] Li-Xin Wang. *Adaptive Fuzzy Systems and Control*. PTR Prentice Hall, 1994.
- [XL89] C.-W. Xu et Y.-Z. Lu. Decoupling in Fuzzy Systems: A Cascade Compensation Approach. *Fuzzy Sets and Systems*, 29:177-185, 1989.
- [Xu91] C.-W. Xu. Linguistic Decoupling Control of Fuzzy Multivariable Processes. *Fuzzy Sets and Systems*, 41:209-217, 1991.
- [YKB77] H. Yoon, G. Klinzing, et H.W. Blanch. Competition for Mixed Substrates by Microbial Populations. *Biotechnology and Bioengineering*, 19:1193-1210, 1977.
- [You96] C. Ben Youssef. *Filtrage, estimation et commande adaptative d'un procédé de traitement des eaux usées*. Thèse de doctorat, Institut National Polytechnique de Toulouse, France, 1996.
- [Zad65] L.A. Zadeh. Fuzzy Sets. *Information and Control*, 8:338-353, 1965.
- [Zad68] L.A. Zadeh. Fuzzy Algorithms. *Information and Control*, 12:94-112, 1968.
- [Zad73] L.A. Zadeh. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3:28-44, 1973.
- [Zam66] G. Zames. On the Input-Output Stability of Time Varying Nonlinear Feedback Systems. Part I: Conditions Derived using Concepts of Loop Gain, Conicity and Positivity. *IEEE Transactions on Automatic Control*, 11(2):228-238, 1966.



Annexe A

Calcul des gradients relatifs à un système flou de Sugeno d'ordre zéro

Nous considérons ici des systèmes flous de Sugeno d'ordre zéro. Par souci de clarté, nous commençons par étudier le cas plus simple d'un contrôleur flou à une seule entrée (paragraphe A.1). Le calcul des gradients est alors étendu à un contrôleur flou à deux entrées (paragraphe A.2). Les ensembles flous considérés en entrée sont des triangles (paragraphe A.3) ou des ensembles flous bidimensionnels (paragraphe A.4) construits à partir d'une triangulation de Delaunay.

A.1 Contrôleur flou à une entrée

Les règles floues de commande sont des règles de type Takagi-Sugeno :

$$R_i : \text{si } e \text{ est } F^i \text{ alors } u \text{ est } c_i. \quad (\text{A.1})$$

Soit $\mu_i = \mu_{F^i}(e) = f^i(e)$ le degré d'activation de la $i^{\text{ème}}$ règle. La sortie du contrôleur flou est alors obtenue par une simple moyenne pondérée par μ_i des singletons c_i :

$$u = \frac{\sum_{i=1}^m \mu_i \cdot c_i}{\sum_{i=1}^m \mu_i} \quad (\text{A.2})$$

Le nombre de règles du contrôleur flou est

$$r = m \quad (\text{A.3})$$

avec m le nombre de labels linguistiques associés à la variable d'entrée e .

Les ensembles flous en entrée sont soit des triangles, soit des ensembles flous bidimensionnels dans le cas où la variable d'entrée est le résultat de la fusion symbolique de deux autres variables : $e = (x, y)$.

La mise en œuvre d'un algorithme de type gradient pour régler les gains et les fonctions d'appartenance en entrée et en sortie du contrôleur flou nécessite le calcul des dérivées partielles suivantes :

- dérivée partielle de u par rapport aux singletons flous :

$$\frac{\partial u}{\partial c_i} = \frac{\mu_i}{\sum_{i=1}^m \mu_i} \quad (\text{A.4})$$

- dérivée partielle de u par rapport à l'entrée et par rapport aux valeurs modales des ensembles flous définis sur l'univers de discours de e :

$$\frac{\partial u}{\partial p} = \frac{\sum_{i=1}^m (c_i - u) \cdot \frac{\partial f^i}{\partial p}}{\sum_{i=1}^m \mu_i} \quad (\text{A.5})$$

Dans l'équation A.5, si $f^i(e)$ est une fonction d'appartenance triangulaire,

$$p = \{e, m^1, \dots, m^k, \dots, m^m\} \quad (\text{A.6})$$

avec m^k la valeur modale du $k^{\text{ème}}$ ensemble flou défini sur l'univers de discours de e . Voir le paragraphe A.3 pour le calcul de $\frac{\partial f^i}{\partial p}$.

Si $f^i(e)$ est une fonction d'appartenance bidimensionnelle,

$$f^i(e) = f^i(x, y). \quad (\text{A.7})$$

Les m ensembles flous en entrée sont définis à partir de la même triangulation de Delaunay d'un ensemble de points $\mathcal{P} = \mathcal{P}_n + \mathcal{P}_{\text{extra}}$ (voir le chapitre 4) avec $\mathcal{P}_n = \{P_1, \dots, P_n\}$. Dans ce cas

$$p = \{x, y, P_x^1, P_y^1, \dots, P_x^k, P_y^k, \dots, P_x^n, P_y^n\}. \quad (\text{A.8})$$

Rappelons que seuls les points de \mathcal{P}_n sont optimisés. P_x^k et P_y^k sont respectivement l'abscisse et l'ordonnée du point P^k de \mathcal{P}_n : $P^k = (P_x^k, P_y^k)$. Le calcul de $\frac{\partial f^i}{\partial p}$ est donné dans le paragraphe A.4.

A.2 Contrôleur flou à deux entrées

Dans le cas d'un contrôleur flou à deux entrées notées e_1 et e_2 , les règles floues sont de la forme :

$$R_{ij} : \text{si } e_1 \text{ est } F_1^i \text{ et } e_2 \text{ est } F_2^j \text{ alors } u \text{ est } c_{ij}. \quad (\text{A.9})$$

Les caractérisations floues des variables e_1 et e_2 sont respectivement

$$\{F_1^1, \dots, F_1^{m_1}\} \quad \text{et} \quad \{F_2^1, \dots, F_2^{m_2}\} \quad (\text{A.10})$$

avec m_1 et m_2 le nombre de labels linguistiques associés à e_1 et e_2 . Ces labels linguistiques sont représentés par des ensembles flous de dimension 1 (triangles) ou 2 (ensembles flous bidimensionnels construits à partir d'une triangulation de Delaunay). u est la variable de sortie du contrôleur flou et c_{ij} est la valeur réelle de commande pour la règle R_{ij} .

Le nombre de règles de ce contrôleur flou est égal à

$$r = m_1 \cdot m_2 \quad (\text{A.11})$$

Si l'opérateur de conjonction (t-norme) est le *produit*, le degré d'activation de la règle R_{ij} est

$$\mu_{ij} = \mu_{F_1^i}(e_1) \cdot \mu_{F_2^j}(e_2) = f_1^i(e_1) \cdot f_2^j(e_2) \quad (\text{A.12})$$

La sortie du contrôleur flou est alors obtenue par une simple moyenne pondérée des conclusions c_{ij} :

$$u = \frac{\sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \mu_{ij} \cdot c_{ij}}{\sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \mu_{ij}} \quad (\text{A.13})$$

La mise en œuvre d'un algorithme de type gradient pour régler les différents paramètres (gains et valeurs modales des fonctions d'appartenance) du précédent contrôleur flou nécessite le calcul des dérivées partielles suivantes :

- dérivée partielle de u par rapport à la 1^{ère} entrée et par rapport aux valeurs modales des ensembles flous définis sur l'univers de discours de e_1 :

$$\frac{\partial u}{\partial p_1} = \frac{\sum_{j=1}^{m_2} \left(f_2^j(e_2) \cdot \sum_{i=1}^{m_1} (c_{ij} - u) \cdot \frac{\partial f_1^i}{\partial p_1} \right)}{\sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \mu_{ij}} \quad (\text{A.14})$$

- dérivée partielle de u par rapport à la 2^{ème} entrée et par rapport aux valeurs modales des ensembles flous définis sur l'univers de discours de e_2 :

$$\frac{\partial u}{\partial p_2} = \frac{\sum_{i=1}^{m_1} \left(f_1^i(e_1) \cdot \sum_{j=1}^{m_2} (c_{ij} - u) \cdot \frac{\partial f_2^j}{\partial p_2} \right)}{\sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \mu_{ij}} \quad (\text{A.15})$$

- dérivée partielle de u par rapport au singleton de la règle R_{ij} :

$$\frac{\partial u}{\partial c_{ij}} = \frac{\mu_{ij}}{\sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \mu_{ij}} \quad (\text{A.16})$$

Les notations p_1 et p_2 utilisées respectivement dans les équations (A.14) et (A.15) dépendent de la dimension des variables en entrée :

- si e_i est de dimension 1,

$$p_i = \{e_i, m_i^1, \dots, m_i^k, \dots, m_i^{m_i}\} \quad (\text{A.17})$$

avec m_i^k la valeur modale du $k^{\text{ème}}$ ensemble flou défini sur l'univers de discours de la $i^{\text{ème}}$ variable d'entrée e_i ,

– si e_i est de dimension 2, $e_i = (x_i, y_i)$ et

$$p_i = \{x_i, y_i, P_{x_i}^1, P_{y_i}^1, \dots, P_{x_i}^k, P_{y_i}^k, \dots, P_{x_i}^n, P_{y_i}^n\} \quad (\text{A.18})$$

avec $P_{x_i}^k$ et $P_{y_i}^k$ les coordonnées du $k^{\text{ème}}$ point P^k de \mathcal{P}_n : $P^k = (P_{x_i}^k, P_{y_i}^k)$.

A.3 Fonctions d'appartenance triangulaires

Les fonctions d'appartenance triangulaires sont représentées dans la figure A.1 et sont définies par :

$$f^i(x) = \begin{cases} \frac{x-m_{i-1}}{m_i-m_{i-1}} & \text{si } m_{i-1} \leq x \leq m_i, \\ \frac{m_{i+1}-x}{m_{i+1}-m_i} & \text{si } m_i \leq x \leq m_{i+1}, \\ 0 & \text{ailleurs.} \end{cases} \quad (\text{A.19})$$

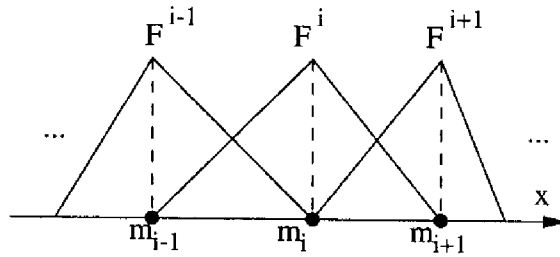


FIG. A.1 – fonctions d'appartenance triangulaires

La dérivée partielle de $f^i(x)$ par rapport à x est donnée par :

$$\frac{\partial f^i(x)}{\partial x} = \begin{cases} \frac{1}{m_i-m_{i-1}} & \text{si } m_{i-1} \leq x \leq m_i, \\ \frac{-1}{m_{i+1}-m_i} & \text{si } m_i \leq x \leq m_{i+1}, \\ 0 & \text{ailleurs.} \end{cases} \quad (\text{A.20})$$

Le calcul des dérivées partielles de $f^i(x)$ par rapport à ses paramètres conduit aux trois équations suivantes :

$$\frac{\partial f^i(x)}{\partial m_{i-1}} = \begin{cases} \frac{f^i(x)-1}{m_i-m_{i-1}} & \text{si } m_{i-1} \leq x \leq m_i, \\ 0 & \text{ailleurs.} \end{cases} \quad (\text{A.21})$$

$$\frac{\partial f^i(x)}{\partial m_i} = \begin{cases} \frac{-f^i(x)}{m_i - m_{i-1}} & \text{si } m_{i-1} \leq x \leq m_i, \\ \frac{f^i(x)}{m_{i+1} - m_i} & \text{si } m_i \leq x \leq m_{i+1}. \\ 0 & \text{ailleurs} \end{cases} \quad (\text{A.22})$$

$$\frac{\partial f^i(x)}{\partial m_{i+1}} = \begin{cases} \frac{1-f^i(x)}{m_{i+1} - m_i} & \text{si } m_i \leq x \leq m_{i+1}, \\ 0 & \text{ailleurs.} \end{cases} \quad (\text{A.23})$$

A.4 Fonctions d'appartenance bidimensionnelles

Les fonctions d'appartenance bidimensionnelles $f^i(x, y)$ sont ici définies à partir d'une triangulation de Delaunay d'un ensemble de points \mathcal{P} .

Le degré d'appartenance de l'entrée (x, y) au $i^{\text{ème}}$ ensemble flou est donné par

$$f^i(x, y) = \mu_{t_i^k}(x, y) = \mu_{t_i^k}(x, y) = a_i^k \cdot x + b_i^k \cdot y + c_i^k \quad (\text{A.24})$$

t_i^k étant le triangle de F^i auquel appartient le point (x, y) .

Soient $p_1 = (x_1, y_1)$, $p_2 = (x_2, y_2)$ et $p_3 = (x_3, y_3)$ les sommets de t_i^k . Les coefficients a_i^k , b_i^k et c_i^k sont calculés comme suit :

$$\begin{pmatrix} a_i^k \\ b_i^k \\ c_i^k \end{pmatrix} = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} \mu_{t_i^k}(p_1) \\ \mu_{t_i^k}(p_2) \\ \mu_{t_i^k}(p_3) \end{pmatrix} \quad (\text{A.25})$$

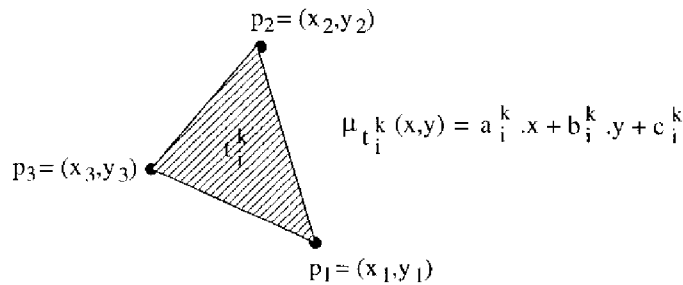


FIG. A.2 - Le degré d'appartenance de l'entrée (x, y) au $i^{\text{ème}}$ ensemble flou est calculé selon une interpolation linéaire entre les 3 sommets du triangle t_i^k auquel appartient le point (x, y) .

La dérivée partielle de $f^i(x, y)$ par rapport à la 1^{ère} composante de l'entrée est donnée par

$$\frac{\partial f^i(x, y)}{\partial x} = \frac{\partial \mu_{t_i^k}(x, y)}{\partial x} = a_i^k \quad (\text{A.26})$$

et la dérivée partielle de $f^i(x, y)$ par rapport à la 2^{ème} composante de l'entrée par

$$\frac{\partial f^i(x, y)}{\partial y} = \frac{\partial \mu_{t_i^k}(x, y)}{\partial y} = b_i^k \quad (\text{A.27})$$

Les coefficients a_i^k , b_i^k et c_i^k sont des fonctions de x_1, y_1, x_2, y_2 et x_3, y_3 uniquement (voir l'équation (A.25)). Par conséquent, les dérivées partielles de $f^i(x, y)$ par rapport aux coordonnées d'un point $P_j = (x_j, y_j)$ non sommet de t_i^k sont nulles. Seules les dérivées partielles de $f^i(x, y)$ par rapport aux coordonnées des sommets du triangle t_i^k sont calculées.

$$\text{On pose } \mu_1 = \mu_{t_i^k}(p_1), \mu_2 = \mu_{t_i^k}(p_2), \mu_3 = \mu_{t_i^k}(p_3) \text{ et } C = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{pmatrix}.$$

Le développement de l'équation (A.25) conduit aux équations littérales suivantes :

$$a_i^k = \frac{y_1(\mu_3 - \mu_2) + y_2(\mu_1 - \mu_3) + y_3(\mu_2 - \mu_1)}{\det(C)} \quad (\text{A.28})$$

$$b_i^k = \frac{x_1(\mu_2 - \mu_3) + x_2(\mu_3 - \mu_1) + x_3(\mu_1 - \mu_2)}{\det(C)} \quad (\text{A.29})$$

$$\begin{aligned} c_i^k &= \frac{x_1(y_2 \cdot \mu_3 - y_3 \cdot \mu_2) + x_2(y_3 \cdot \mu_1 - y_1 \cdot \mu_3) + x_3(y_1 \cdot \mu_2 - y_2 \cdot \mu_1)}{\det(C)} \quad (\text{A.30}) \\ &= \frac{y_1(x_3 \cdot \mu_2 - x_2 \cdot \mu_3) + y_2(x_1 \cdot \mu_3 - x_3 \cdot \mu_1) + y_3(x_2 \cdot \mu_1 - x_1 \cdot \mu_2)}{\det(C)} \end{aligned}$$

avec

$$\begin{aligned} \det(C) &= x_2 y_3 - x_3 y_2 + x_3 y_1 - x_1 y_3 + x_1 y_2 - x_2 y_1 \\ &= x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2) \quad (\text{A.31}) \\ &= y_1(x_3 - x_2) + y_2(x_1 - x_3) + y_3(x_2 - x_1) \end{aligned}$$

Les dérivées partielles de $f^i(x, y)$ par rapport aux sommets $p_j = (x_j, y_j)$, $j = 1, \dots, 3$, de t_i^k sont obtenues par :

$$\frac{\partial f^i(x, y)}{\partial x_j} = \frac{\partial \mu_{t_i^k}(x, y)}{\partial x_j} = x \frac{\partial a_i^k}{\partial x_j} + y \frac{\partial b_i^k}{\partial x_j} + \frac{\partial c_i^k}{\partial x_j} \quad (\text{A.32})$$

$$\frac{\partial f^i(x, y)}{\partial y_j} = \frac{\partial \mu_{t_i^k}(x, y)}{\partial y_j} = x \frac{\partial a_i^k}{\partial y_j} + y \frac{\partial b_i^k}{\partial y_j} + \frac{\partial c_i^k}{\partial y_j} \quad (\text{A.33})$$

avec

$$\frac{\partial a_i^k}{\partial x_1} = \frac{a_i^k(y_3 - y_2)}{\det(C)} \quad \frac{\partial a_i^k}{\partial y_1} = \frac{(\mu_3 - \mu_2) + a_i^k(x_2 - x_3)}{\det(C)} \quad (\text{A.34})$$

$$\frac{\partial a_i^k}{\partial x_2} = \frac{a_i^k(y_1 - y_3)}{\det(C)} \quad \frac{\partial a_i^k}{\partial y_2} = \frac{(\mu_1 - \mu_3) + a_i^k(x_3 - x_1)}{\det(C)} \quad (\text{A.35})$$

$$\frac{\partial a_i^k}{\partial x_3} = \frac{a_i^k(y_2 - y_1)}{\det(C)} \quad \frac{\partial a_i^k}{\partial y_3} = \frac{(\mu_2 - \mu_1) + a_i^k(x_1 - x_2)}{\det(C)} \quad (\text{A.36})$$

$$\frac{\partial b_i^k}{\partial x_1} = \frac{(\mu_2 - \mu_3) + b_i^k(y_3 - y_2)}{\det(\mathcal{C})} \quad \frac{\partial b_i^k}{\partial y_1} = \frac{b_i^k(x_2 - x_3)}{\det(\mathcal{C})} \quad (\text{A.37})$$

$$\frac{\partial b_i^k}{\partial x_2} = \frac{(\mu_3 - \mu_1) + b_i^k(y_1 - y_3)}{\det(\mathcal{C})} \quad \frac{\partial b_i^k}{\partial y_2} = \frac{b_i^k(x_3 - x_1)}{\det(\mathcal{C})} \quad (\text{A.38})$$

$$\frac{\partial b_i^k}{\partial x_3} = \frac{(\mu_1 - \mu_2) + b_i^k(y_2 - y_1)}{\det(\mathcal{C})} \quad \frac{\partial b_i^k}{\partial y_3} = \frac{b_i^k(x_1 - x_2)}{\det(\mathcal{C})} \quad (\text{A.39})$$

$$\frac{\partial c_i^k}{\partial x_1} = \frac{y_2(\mu_3 - c_i^k) + y_3(c_i^k - \mu_2)}{\det(\mathcal{C})} \quad \frac{\partial c_i^k}{\partial y_1} = \frac{x_3(\mu_2 - c_i^k) + x_2(c_i^k - \mu_3)}{\det(\mathcal{C})} \quad (\text{A.40})$$

$$\frac{\partial c_i^k}{\partial x_2} = \frac{y_3(\mu_1 - c_i^k) + y_1(c_i^k - \mu_3)}{\det(\mathcal{C})} \quad \frac{\partial c_i^k}{\partial y_2} = \frac{x_1(\mu_3 - c_i^k) + x_3(c_i^k - \mu_1)}{\det(\mathcal{C})} \quad (\text{A.41})$$

$$\frac{\partial c_i^k}{\partial x_3} = \frac{y_1(\mu_2 - c_i^k) + y_2(c_i^k - \mu_1)}{\det(\mathcal{C})} \quad \frac{\partial c_i^k}{\partial y_3} = \frac{x_2(\mu_1 - c_i^k) + x_1(c_i^k - \mu_2)}{\det(\mathcal{C})} \quad (\text{A.42})$$

Si l'on remplace les dérivées partielles (A.34) à (A.40) dans les équations (A.32) et (A.33), on aboutit aux six équations suivantes pour le calcul des dérivées partielles de $f^i(x, y) = \mu_i^k(x, y)$ par rapport aux coordonnées des sommets de t_i^k :

$$\frac{\partial \mu_i^k}{\partial x_1} = \frac{\mu_i^k(y_3 - y_2) + \mu_2(y - y_3) + \mu_3(y_2 - y)}{\det(\mathcal{C})} \quad (\text{A.43})$$

$$\frac{\partial \mu_i^k}{\partial y_1} = \frac{\mu_i^k(x_2 - x_3) + \mu_2(x_3 - x) + \mu_3(x - x_2)}{\det(\mathcal{C})} \quad (\text{A.44})$$

$$\frac{\partial \mu_i^k}{\partial x_2} = \frac{\mu_i^k(y_1 - y_3) + \mu_3(y - y_1) + \mu_1(y_3 - y)}{\det(\mathcal{C})} \quad (\text{A.45})$$

$$\frac{\partial \mu_i^k}{\partial y_2} = \frac{\mu_i^k(x_3 - x_1) + \mu_3(x_1 - x) + \mu_1(x - x_3)}{\det(\mathcal{C})} \quad (\text{A.46})$$

$$\frac{\partial \mu_i^k}{\partial x_3} = \frac{\mu_i^k(y_2 - y_1) + \mu_1(y - y_2) + \mu_2(y_1 - y)}{\det(\mathcal{C})} \quad (\text{A.47})$$

$$\frac{\partial \mu_i^k}{\partial y_3} = \frac{\mu_i^k(x_1 - x_2) + \mu_1(x_2 - x) + \mu_2(x - x_1)}{\det(\mathcal{C})} \quad (\text{A.48})$$

Réduction de la complexité des contrôleurs flous : Application à la commande multivariable.

La commande en logique floue permet de s'affranchir de l'utilisation de modèles mathématiques parfois difficiles à obtenir. Sa capacité à traduire la connaissance d'un opérateur humain en règles d'expertise énoncées dans un langage simple en fait une technique très prometteuse. Néanmoins, lorsque le nombre de variables entrant en jeu devient trop important, la base de règles explose très vite, et des problèmes liés à sa réalisation pratique en découlent.

Cette thèse s'inscrit dans la mouvance des travaux actuels sur la commande floue et s'attache au problème de l'explosion combinatoire du nombre de règles.

Dans une première partie, les principes de base de la logique floue et de la commande floue sont rappelés.

Dans une deuxième partie, des solutions visant à simplifier la synthèse d'un contrôleur flou sont présentées.

Deux cas sont considérés : la base de règles existe déjà, la base de règles n'est pas disponible et la synthèse d'un contrôleur flou de complexité réduite est à réaliser. Dans la pratique, on ne dispose généralement pas de cette base de règles, aussi, on insiste davantage sur le deuxième cas de figure. Une fois la structure du contrôleur flou défini, le nombre de paramètres à régler pouvant atteindre un nombre impressionnant, il est intéressant d'utiliser des techniques d'apprentissage afin d'automatiser la mise au point du contrôleur flou. Les paramètres de ce dernier (gains et fonctions d'appartenance, en entrée et en sortie) sont ici réglés à travers la méthode, très simple, de descente du gradient.

Dans une troisième partie, la démarche proposée dans ce mémoire est appliquée avec succès à la commande de deux processus multivariables : un bac mélangeur et un procédé biologique de traitement des eaux-usées.

Mots clefs: logique floue, commande floue multivariable, réduction de la base de règles, auto-réglage, méthode de descente du gradient.

Complexity reduction of fuzzy controllers : Application to the multivariable control

Fuzzy control enables to avoid the use of mathematical models sometimes difficult to obtain. Its capability to translate the human operator knowhow into expertise rules expressed in a simple language makes it a very promising technique. Nevertheless, when the number of variables to be considered becomes too high, the rule base rapidly explodes, and problems linked to its implementation follow.

This thesis resides within the framework of the actual research works on fuzzy control and treats the problem of the combinatorial explosion in the number of rules.

In the first part, the basic principles of fuzzy logic and fuzzy control are recalled.

In the second part, solutions, in order to make easier the synthesis of a fuzzy logic controller, are presented.

Two cases are considered : the rule base already exists, the rule base is not available and we have to develop a fuzzy logic controller of reduced complexity. In practice, we generally do not have this rule base. Therefore, we lay stress on the second point. Once the structure of the fuzzy logic controller has been decided, the number of parameters to be tuned may reach a high level. It becomes then interesting to apply some learning techniques in order to automatically tune the numerous parameters (gains and membership functions for both inputs and outputs) of the fuzzy logic controller. These parameters are optimized using the very simple gradient descent method.

In the third part, the methodology proposed in this dissertation is applied with success to the control of two multivariable processes : a mixing tank and a biological waste-water treatment process.

Key words : fuzzy logic, multivariable fuzzy control, rule-base reduction, self-tuning, gradient descent method.