



# Contrôle flou : de l'interpolation numérique au codage de l'expertise

Sylvie Galichet

► **To cite this version:**

Sylvie Galichet. Contrôle flou : de l'interpolation numérique au codage de l'expertise. Informatique [cs]. Université de Savoie, 2001. <tel-00413006>

**HAL Id: tel-00413006**

**<https://tel.archives-ouvertes.fr/tel-00413006>**

Submitted on 2 Sep 2009

**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.

**Université de Savoie**

**Habilitation à diriger des recherches**  
en **Electronique, Electrotechnique et Automatique**

**Contrôle flou : de l'interpolation numérique  
au codage de l'expertise**

**Sylvie Galichet**

Travaux présentés le 21 Décembre 2001 devant la Commission d'Examen :

J.M. Dion	Rapporteur
D. Dubois	Rapporteur
B. Dubuisson	Président
L. Foulloy	
S. Gentil	
R. Longchamp	
A. Titli	Rapporteur



Je tiens tout d'abord à remercier Bernard Dubuisson, Professeur à l'Université de Technologie de Compiègne, de m'avoir fait l'honneur de présider le jury. J'espère avoir été à la hauteur des attentes que peut avoir un directeur de thèse envers une étudiante à qui il a donné goût à la recherche.

Je tiens aussi à remercier Jean-Michel Dion, Directeur de Recherche CNRS au Laboratoire d'Automatique de Grenoble, d'avoir accepté d'être rapporteur de ce travail. Ses remarques et critiques constructives sont pour moi d'une grande utilité et j'espère en tirer parti avec profit.

Tous mes remerciements vont également à Didier Dubois, Directeur de Recherche CNRS à l'IRIT, et à André Titli, Professeur à l'INSA de Toulouse, qui tous deux ont accepté la lourde tâche de rapporter sur ce manuscrit. Outre le soin qu'ils ont apporté à l'analyse de ce document, ils m'ont toujours encouragée à approfondir des travaux en commande floue, notamment via des discussions au sein des groupes de travail CSN et FALCON. Je souhaite que nos échanges conservent le caractère bénéfique qu'ils ont eu et que mon année toulousaine à l'IRIT soit l'occasion d'ancrer une coopération fructueuse et durable.

Un grand merci également à Sylviane Gentil, Professeur à l'ENSIEG, d'avoir accepté d'examiner ce travail. Ayant participé à des projets communs, j'ai eu maintes occasions d'apprécier la cordialité avec laquelle elle partage ses opinions mais aussi ses compétences.

Que Roland Longchamp, Professeur à l'EPFL de Lausanne, soit également remercié pour sa participation au jury. Ses remarques et interrogations soulèvent de nombreux sujets de réflexion qui restent encore à approfondir.

Enfin, je tiens tout particulièrement à remercier Laurent Foulloy, Professeur à l'Université de Savoie, pour m'avoir incitée à rejoindre son équipe de recherche. Par son enthousiasme, sa compétence mais aussi la confiance qu'il m'a accordée, il m'a donné l'envie d'apprendre et de progresser. Durant ces dix dernières années, il a toujours su m'encourager, se montrer disponible et chaleureux et supporter mes sautes d'humeur. En résumé, je lui suis extrêmement reconnaissante d'avoir été le directeur de recherche que je souhaite à tout le monde.

Merci également à toutes les personnes qui de près ou de loin ont contribué à la réussite de ce travail, avec une mention particulière pour les pionniers de "Commande experte", Eric et Gilles. Je n'oublie pas les étudiants qui par leurs travaux ont contribué à l'avancement de cette recherche, notamment les doctorants : Yassar, Michel et Reda, mais aussi les DEA et stagiaires.

Finalement, je ne saurais terminer cette page sans que soient remerciés tous les collègues du Laboratoire ou plus généralement de l'Ecole, pour leur soutien moral, administratif ou technique. Il m'est ici impossible de citer tout le monde, alors mentionnons simplement les anti-tabacs : Emmanuel et Philippe, le secrétariat LAMII : Joelle, et tous les autres ....



# Table des matières

---

## Partie A : Synthèse

I	Curriculum Vitae .....	3
II	Activités d'enseignement .....	4
III	Activités de recherche .....	5
IV	Activités d'encadrement .....	8
V	Activités d'intérêt collectif .....	9
VI	Publications .....	11

---

## Partie B : Travaux de recherche

<b>Introduction générale .....</b>	<b>19</b>
------------------------------------	-----------

<b>Chapitre I : Interpolation numérique .....</b>	<b>21</b>
---	-----------

I	Introduction .....	21
II	Contrôleurs flous numériques .....	22
II.1	Contrôleurs flous de Mamdani .....	23
II.2	Contrôleurs flous de Sugeno .....	23
III	Equivalences modales .....	24
III.1	Propriété aux valeurs modales .....	24
III.2	Principe des équivalences modales .....	25
IV	Application du principe des équivalences modales à la synthèse d'un contrôleur flou de type PI.....	26
IV.1	Partitionnement des univers de discours .....	26
IV.2	Synthèse d'un contrôleur flou de type Mamdani .....	27
IV.2.1	Construction de la base de règles.....	27

IV.2.2	<i>Réglage du contrôleur synthétisé</i> .....	28
IV.2.3	<i>Introduction de gains de normalisation</i> .....	29
IV.3	Synthèse d'un contrôleur flou de type Sugeno .....	31
IV.4	Effet de l'interpolation entre les points modaux dans un contrôleur de Mamdani .....	31
IV.5	Obtention d'un équivalent linéaire exact .....	35
V	Extension à la synthèse de contrôleurs flous linéaires par retour d'état .....	37
VI	Application à la synthèse de contrôleurs flous non linéaires .....	39
VI.1	Régulation de niveau en raffinerie .....	39
VI.2	Transformation linéarisante .....	41
VII	Apprentissage d'une base de règles à partir d'exemples .....	44
VII.1	Préambule .....	44
VII.2	Technique d'apprentissage .....	45
VII.3	Application : prédiction de la tension artérielle .....	47
VIII	Propriétés structurelles des systèmes de Sugeno à conclusion constante .....	48
VIII.1	Décomposition en sous-systèmes élémentaires .....	48
VIII.2	Propriétés d'un sous-système élémentaire .....	49
VIII.3	Inversion d'un système flou .....	51
IX	Synthèse d'une loi de commande par inversion de modèle flou : approche par linéarisation entrée/sortie.....	52
IX.1	Principe .....	53
IX.2	"Robustification" de la structure de commande .....	53
IX.3	Identification du modèle flou .....	56
X	Conclusion .....	57

## **Chapitre II : Codage de l'expertise .....59**

I	Introduction .....	59
II	Manipulation d'informations symboliques .....	61
II.1	Description et signification .....	61
II.2	Typologie des systèmes flous .....	62
II.3	Implantation SF4 .....	64
III	Traitement d'entrées imprécises .....	66
IV	Lien avec l'approche numérique TSK .....	70
IV.1	TSK <--> SF4 avec règles pondérées .....	70
IV.2	Extension du mécanisme de calcul TSK aux règles à conclusions linguistiques floues .....	73
IV.3	Approximation linguistique ou réduction d'une base de règles .....	75
V	Expertise en commande floue .....	78
V.1	Expertise et contrôleurs symboliques .....	79
V.1.1	<i>Règles générales</i> .....	82

V.1.2	<i>Gestion de la commutation</i> .....	82
V.1.3	<i>Règles anticipatives</i> .....	83
V.1.4	<i>Ajustement du contrôleur flou</i> .....	83
V.2	Expertise et contrôleurs numériques .....	85
V.2.1	<i>Régulation de niveau dans un procédé de fabrication     semi-continue de barres de laiton</i> .....	85
V.2.2	<i>Régulation de niveau en raffinerie</i> .....	87
V.3	Aspect méthodologique .....	88
VI	Distribution de l'expertise dans la structure de commande .....	90
VI.1	Introduction .....	90
VI.2	Expertise et capteurs flous .....	91
VI.2.1	<i>Agrégation par interpolation</i> .....	91
VI.2.2	<i>Agrégation par jeu de règles</i> .....	95
VI.2.2.1	Capteur flou de débouchage .....	95
VI.2.2.2	Capteur flou d'oscillations .....	96
VI.3	Expertise et hiérarchisation .....	97
VI.3.1	<i>Supervision de lois de commande</i> .....	97
VI.3.2	<i>Hiérarchisation d'une connaissance experte</i> .....	99
VII	Conclusion .....	102
	<b>Conclusion générale .....</b>	<b>105</b>
	<b>Références .....</b>	<b>107</b>



## **Partie A : Synthèse**



# I Curriculum Vitae

Nom :	GALICHET
Prénoms :	Sylvie, Claire, Annie
Date et lieu de naissance :	18-1-1963 à Villers-Semeuse (08)
Nationalité :	Française
Situation de famille :	Célibataire
Adresse personnelle :	7, rue du val vert 74600 Seynod
Téléphone :	04 50 51 21 28

## I.1 Statut

Fonction : Maître de conférences, 61<sup>ème</sup> section  
Etablissement : ESIA (Ecole Supérieure d'Ingénieurs d'Annecy) - Université de Savoie  
BP806 74016 Annecy Cedex  
Date d'installation : 1-10-91  
Date de titularisation : 1-10-93  
Situation actuelle : Délégation CNRS - IRIT Toulouse  
Titulaire de la prime d'encadrement doctoral depuis 95

## I.2 Principaux diplômes

Juil. 86 Diplôme d'ingénieur en informatique de l'UTC (Université de Technologie de Compiègne)  
Sept. 86 DEA, spécialité Contrôle des systèmes, UTC, obtenu avec mention  
Déc. 89 Thèse de Doctorat, spécialité Contrôle des systèmes, UTC, mention très honorable

## I.3 Expériences professionnelles

88 / 90 Stagiaire doctorant, Service Etudes Avancées, Thomson Vidéo Equipement, Cergy-Pontoise  
90 / 91 Ingénieur de recherche CNRS, affecté au GdR Traitement du Signal et Images, en poste à l'EN-SERB, Bordeaux  
91 / 01 Maître de conférences à l'Université de Savoie, ESIA Annecy

## I.4 Activités de recherche

86 / 90 Laboratoire Heudiasyc (HEUristique et DIAgnostic des SYstèmes Complexes), URA CNRS 817, UTC, Compiègne.  
DEA et thèse sous la direction de B. DUBUISSON, professeur à l'UTC.  
Thèmes : Reconnaissance des formes statistique, robotique industrielle, traitement d'images couleur.  
90 / 91 GdR Traitement du Signal et Images, laboratoire LARFRA (Laboratoire de Reconnaissance des Formes et Robotique Agricole), ENSERB, Bordeaux.  
Thème : Développement logiciel en analyse et traitement d'images.  
91 / 01 Laboratoire LAMII (Laboratoire d'Automatique et de Micro-Informatique Industrielle), ESIA.  
Thèmes : Modélisation et contrôle flou.

## I.5 Activités d'enseignement

88 / 90 Vacations à l'UTC (Compiègne).  
Responsabilité de cours, TD, et TP de programmation structurée en 1ère année de formation d'ingénieurs.  
90 / 91 Vacations à l'ENSERB (Bordeaux).  
Création d'un cours de codage et compression de l'image en 3ème année de formation d'ingénieurs.  
91 / 01 Service statutaire à l'ESIA (Annecy).  
Responsabilité de cours, TD et TP en école d'ingénieurs, MST, DESS et DEA.

## I.6 Principales responsabilités collectives

Membre du CA de l'Université de Savoie (mandat 92-96).  
Membre du CA de l'ESIA depuis 94.  
Membre de la CSE 61ème section de l'Université de Savoie depuis 95.  
Responsable de l'opération "Commande Experte" au LAMII (mandats 92-94 et 94-96).

## II Activités d'enseignement

### II.1 Panorama

A ma nomination à l'université de Savoie en Octobre 91, j'intègre l'UFR FAST (Faculté Annécienne des Sciences et Techniques) qui regroupe deux Maîtrises de Sciences et Techniques (PAIM : Productique Appliquée aux Industries Mécaniques et MIA : Micro-Informatique et Automatique). En Septembre 93, la FAST est transformée en Ecole Supérieure d'Ingénieurs d'Annecy (ESIA). Issues des formations existantes, deux filières d'ingénieurs sont ainsi créées. Parallèlement, une nouvelle Maîtrise de Sciences et Techniques (PAI : Physique Appliquée et Instrumentation) voit le jour. Cette dernière a aujourd'hui rejoint l'ESIA qui actuellement, compte donc trois filières d'ingénieurs.

Dans ce contexte, j'ai été amenée à m'investir dans les différentes filières au gré de leur création, de leur évolution et du recrutement des enseignants. J'ai ainsi créé ou assuré la responsabilité de nombreux modules dont la diversité est illustrée par la liste des intitulés de cours correspondants :

<i>Systèmes d'exploitation</i>	<i>Unix</i>
<i>Analyse numérique</i>	<i>Algorithmie</i>
<i>Langage C</i>	<i>Programmation machine et assembleur</i>
<i>Techniques d'interfaçage</i>	<i>Logique floue</i>
<i>Programmation graphique sous Labview</i>	<i>Instrumentation virtuelle</i>
<i>Communication inter-applications</i>	

Chacun de ces modules représente environ 12h de cours, 12h de TD et 16h de TP. Hormis le cours à assurer, leur responsabilité implique l'animation pédagogique, l'organisation et la mise en place des TD et TP ainsi que l'encadrement d'au moins un groupe d'étudiants. Par ailleurs, j'ai participé à des enseignements de Prolog, d'algèbre linéaire, d'automatique continue ou encore de Java.

### II.2 Génie informatique

Ces dernières années, les formations dispensées à l'ESIA s'étant stabilisées, j'ai recentré mes activités autour du génie informatique. J'ai notamment introduit une nouvelle formation basée sur l'environnement de développement graphique Labview. Composée de trois modules, celle-ci a été complètement créée à l'ouverture de la filière PAI. Aujourd'hui, elle est également dispensée en filière A2I (Automatique et Informatique Industrielle). Un premier module d'initiation à la programmation sous Labview est enseigné en deuxième année. Si celui-ci est commun aux deux filières concernées, l'enseignement de troisième année prend un caractère spécifique à travers deux modules. Selon la spécialité des étudiants sont ainsi traités la programmation à distance des instruments ou le développement d'applications informatiques communicantes. Restent à souligner que ces deux derniers modules sont essentiellement basés sur des enseignements pratiques et que leur mise en place a nécessité de concevoir de nouvelles plateformes de manipulation. Le choix d'un matériel attractif aux yeux des étudiants (mini-robots, Webcam, ...) a permis de motiver les élèves mais aussi d'aborder de façon concrète des applications multidisciplinaires.

De façon plus globale et par le biais de suivis de stages en entreprise, j'ai pu évaluer et tenter d'améliorer l'adéquation entre la formation de nos étudiants et les besoins industriels en terme de génie informatique.

### **II.3 Lien avec la recherche**

Outre les enseignements évoqués ci-dessus qui concernent exclusivement le second cycle, je suis intervenue en DEA Automatique Industrielle, cohabilité entre l'université Claude Bernard de Lyon 1, l'INSA de Lyon et l'université de Savoie. C'est ainsi que j'ai enseigné un cours de commande floue, planifié tous les deux ans en alternance avec un module de capteurs flous. La répartition du public étudiant sur plusieurs sites géographiques m'a amenée à concevoir une version visio-conférence de cet enseignement.

Pour conclure sur mes activités d'enseignement, ajoutons que j'ai proposé et encadré de nombreux projets d'étudiants au laboratoire (24h en 2<sup>ème</sup> année, 120h en 3<sup>ème</sup>).

### **III Activités de recherche**

Dès mon arrivée à Annecy, j'ai rejoint le LAMII (Laboratoire d'Automatique et de Micro-Informatique Industrielle) et intégré l'opération "Commande Experte" pour développer des travaux en commande floue. Malgré des restructurations internes au laboratoire, ce thème est resté le fil conducteur de mes activités de recherche tout au long de ces dix dernières années.

Après une brève période de reconversion thématique puisque mes travaux de recherche initiaux étaient dans le domaine du traitement des images, j'ai développé deux axes principaux. Le premier concerne la synthèse de contrôleurs flous et repose sur le principe des équivalences modales. Quant au second, il découle de la typologie des contrôleurs flous et exploite une implantation symbolique des systèmes flous.

#### **III.1 Principe des équivalences modales**

Dans les années 90, la communauté internationale s'attache à analyser les performances associées aux contrôleurs flous. Dans ce contexte, le projet CSN (Commande Symbolique et Neuro-mimétique) du pôle AI du GR Automatique initie une étude comparative de divers contrôleurs flous sur un "benchmark" issu de la commande adaptative. La participation active de l'équipe du LAMII à ce groupe de travail nous a amenés à rédiger un article de synthèse sur le sujet [R3] mais aussi à réfléchir sur les mécanismes de réglage d'un contrôleur flou. Dans l'optique d'une future exploitation de résultats connus en commande linéaire, nous nous sommes alors posé la question suivante : "Comment synthétiser de manière automatique un contrôleur flou équivalent à un contrôleur linéaire ?". Le principe des équivalences modales, développé au sein de l'équipe, permet de solutionner ce problème. Ainsi, étant donné un contrôleur connu de type P (proportionnel), PI (proportionnel intégral) ou PID (proportionnel intégral dérivé), il devient possible d'automatiquement générer la base de règles et les réglages d'un contrôleur flou équivalent. Si la construction du système flou permet de garantir l'équivalence pour des entrées modales, seul un choix approprié des opérateurs d'inférence permet d'obtenir une interpolation linéaire entre les points modaux.

Les premiers travaux publiés dans [R2] sont à l'origine de nombreuses extensions visant à appliquer le principe des équivalences modales pour la synthèse de contrôleurs plus élaborés. Toujours dans un cadre linéaire, est ainsi abordée la construction de contrôleurs flous par retour d'état [O4]. La tentative s'avère fructueuse pour des procédés mono-entrée, mono-sortie. Par contre, le traitement des procédés multivariables reste délicat lorsqu'une implantation de type Mamdani est envisagée pour le contrôleur flou. Un système flou spécifique doit, en effet, être associé à chaque sortie dans la mesure où aucun réglage des fonctions d'appartenance associées aux entrées ne permet de traiter simultanément les différentes sorties. Cette difficulté disparaît lorsque des contrôleurs de type Sugeno sont exploités, ces derniers n'impo-

sant aucune contrainte sur la spécification des conséquences des règles.

Dès 93, nous avons abordé l'aspect non linéaire en particulier à travers une application industrielle de régulation de niveau dans une capacité tampon en raffinerie [C8]. Cette étude, menée en collaboration avec ELF, exploite le principe des équivalences modales pour initialiser un contrôleur flou à partir d'un contrôleur non linéaire existant. Suite à ces travaux, nous nous sommes intéressés à la synthèse d'un système flou implantant une transformation linéarisante connue dans le cadre d'une commande par linéarisation entrée-sortie.

Tous ces travaux ont pour but de copier un contrôleur existant mais n'apportent aucun élément de réponse au véritable problème de synthèse des contrôleurs flous. Dans ce contexte, la démarche de l'automatique conventionnelle requiert la connaissance d'un modèle du procédé à piloter. Dans un cadre flou, apparaît alors la problématique de la modélisation floue. Celle-ci a été appréhendée dans la thèse de Y. Nakoula visant à identifier un modèle flou à partir de points d'apprentissage. La méthode proposée consiste à sélectionner un sous-ensemble de points d'apprentissage qui devront être reproduits de manière exacte par le modèle flou. Pour ce faire, et toujours par application du principe des équivalences modales, la construction du modèle est réalisée de façon à associer un point modal à chaque point concerné [O3].

L'ensemble des travaux précédents a été exploité dans la thèse de R. Boukezzoula pour répondre au problème de suivi de trajectoire pour une classe de systèmes non linéaires. Deux architectures de commande ont été proposées, la première basée sur une approche par linéarisation entrée-sortie, la seconde implantant une structure de commande à modèle interne [R6]. Dans les deux cas, le contrôleur est obtenu par inversion du modèle flou du procédé. Une composante de commande additive est intégrée dans la structure globale de façon à la rendre robuste vis-à-vis des incertitudes. Cette dernière est synthétisée par mode de glissement ou par optimisation Hinfini.

### **III.2 Implantation symbolique d'un système flou**

Pour concevoir et implanter un contrôleur, il est intéressant de pouvoir exploiter l'ensemble des connaissances disponibles sur le procédé : modèle mathématique, données d'apprentissage mais aussi expertise. Si les travaux issus du principe des équivalences modales permettent d'intégrer aisément des informations numériques, la prise en compte de la connaissance experte reste délicate. Dans ce contexte, l'équipe s'intéresse aux mécanismes de conversion numérique/symbolique et symbolique/numérique intervenant dans les systèmes flous. C'est ainsi que la typologie des contrôleurs flous [O1], développée au laboratoire, distingue quatre familles de systèmes selon la nature des informations véhiculées en interne. Celle-ci permet de classifier facilement les contrôleurs existants mais aussi d'implanter un même contrôleur de différentes manières.

Ces travaux ont notamment permis d'exhiber la possibilité d'une implantation purement symbolique des systèmes flous. Dans ce cas, ces derniers, dénotés SF4, exploitent des règles linguistiques, éventuellement pondérées. A ce titre, ils sont tout particulièrement adaptés à la représentation des connaissances d'expert. Ils possèdent également l'avantage de dissocier clairement les phases de fuzzification, d'inférence et de défuzzification, chacune de ces fonctionnalités pouvant donc être réalisée par un composant spécifique. Un capteur flou est ainsi facilement intégrable dans une structure de commande floue [R1].

Dans le cadre de la thèse de M. Dussud, contrat CIFRE avec la SERT (Société d'Etudes et de Réalisations Techniques), ce type de systèmes flous a été utilisé pour la régulation de niveau d'acier en lingotière dans une machine de coulée continue [R4]. L'approche proposée est basée sur une structure à deux niveaux, un niveau bas permettant d'intégrer des correcteurs de nature quelconque et un niveau haut correspondant à une couche de supervision. Cette dernière a pour

objectif de déterminer les contrôleurs à utiliser en fonction du contexte identifié. La situation courante est qualifiée linguistiquement par le biais de capteurs flous de perturbations (détection d'oscillations et de débouchages de busette).

Si les systèmes SF4 facilitent la manipulation d'informations linguistiques, ils sont également exploitables dans un contexte numérique. En effet, tout système flou de Sugeno à conclusions constantes peut être implanté de façon exacte sous la forme d'un SF4. Les travaux issus du principe des équivalences modales sont alors directement transposables dans un monde linguistique. Ainsi, la technique d'apprentissage évoquée précédemment a pu être mise en oeuvre dans un cadre symbolique, notamment à des fins de simplification ou de réduction d'une base de règles. C'est également la stratégie utilisée pour la prédiction des variations de la pression artérielle à partir de la fréquence cardiaque et de l'accélération corporelle [R5] (collaboration avec le LAG, le CHU de Grenoble et l'entreprise ELA Recherche spécialisée dans l'appareillage médical, projet Emergence de la région Rhône-Alpes).

L'approche symbolique de l'inférence a également été validée dans le cadre du raisonnement approché [O5]. Il a notamment été montré que dans le cas d'une représentation conjonctive des règles, l'inférence peut être menée tant au niveau numérique qu'au niveau symbolique. Dans le cas de règles implicatives, le problème a été restreint aux règles à certitude utilisant l'implication de Dienes, hypothèse sous laquelle il est possible de se ramener au cas conjonctif par une mise sous forme normale disjonctive.

### **III.3 Projets en cours et perspectives**

La majorité des travaux développés dans l'équipe depuis une dizaine d'années se situent dans une stratégie de commande bas niveau. Si cette stratégie a permis d'établir un "pont" entre automatique conventionnelle et commande floue, ils semblent aujourd'hui aboutis dans la mesure où les systèmes flous utilisés peuvent s'exprimer analytiquement et ainsi être introduits dans une structure de commande classique. On peut alors qualifier l'approche floue de prédictive, robuste, optimale, ..., selon la nature de la structure de commande exploitée.

Cette évolution de la commande floue au cours de la décennie passée n'est pas spécifique au LAMII, ni même à la communauté française, puisque la majorité des publications du domaine s'inscrit dans cette démarche. On peut toutefois noter que cette tendance s'oppose aux motivations d'origine de la commande floue, à savoir exploitation d'une expertise et prise en compte des imprécisions. Outre ce constat, il apparaît qu'une réelle avancée en terme d'automatisation des systèmes industriels ne peut être obtenue qu'au prix d'une globalisation de l'approche. Autrement dit, il est souhaitable de repenser la problématique en terme d'optimisation du fonctionnement global d'une application et non plus simplement en terme d'optimisation de chacun de ses constituants. Dans cette optique, on est alors amené à envisager une meilleure exploitation des connaissances de nature symbolique de façon à aborder une problématique de commande beaucoup plus générale, c'est-à-dire de haut niveau, regroupant les aspects liés à la fusion d'informations, le diagnostic, la décision et la supervision.

Des travaux préliminaires ont déjà été entrepris dans ce sens, notamment dans la thèse de M. Dussud. Il convient maintenant de les généraliser et de les intégrer dans une méthodologie globale. Une telle démarche nécessite d'oublier dans un premier temps l'aspect commande pour se focaliser sur la théorie des systèmes flous (sémantique des règles, raisonnement par contraintes, représentation des incertitudes, ...). C'est dans ce contexte que j'ai demandé et obtenu une délégation auprès du CNRS afin de rejoindre l'équipe de D. Dubois et H. Prade à l'IRIT pour l'année universitaire 2001-2002. Ce séjour devrait me permettre d'approfondir mes connaissances et de faire émerger de nouvelles problématiques en terme d'automatisation des systèmes.

## IV Activités d'encadrement

### IV.1 Thèses

- [Nakoula97] Nakoula Y., "Apprentissage des modèles linguistiques flous, par jeu de règles pondérées", Thèse de doctorat, Spécialité Electronique - Electrotechnique - Automatique, Université de Savoie, 21 Juillet 97.  
Jury : S. Gentil (P), H. Bersini (R), J. Ragot (R), H. Prade, S. Galichet, L.Foulloy.
- [Dussud99] Dussud M., "Régulation de niveau en lingotière de coulée continue : approche classique et approche floue", Thèse de doctorat, Spécialité Electronique - Electrotechnique - Automatique, Université de Savoie, 5 Novembre 99.  
Jury : P.M. Frank (P), A. Chérury (R), T.M. Guerra (R), L.Foulloy, S. Galichet, P. Simonnin.
- [Boukezzoula00] Boukezzoula R., "Commande floue d'une classe de systèmes non linéaires : application au problème de suivi de trajectoire", Thèse de doctorat, Spécialité Electronique - Electrotechnique - Automatique, Université de Savoie, 31 Mars 2000.  
Jury : P. Borne (P), A. Ollero (R), C. Mélin (R), A. Titli, L.Foulloy, S. Galichet.

### IV.2 DEA

- [AlSaleh94] Al Saleh H., "Analyse de la sémantique attribuée aux systèmes flous", Rapport DEA Automatique Industrielle, LAMII/CESALP, Université de Savoie, Annecy, France, Juillet 94.
- [Darbas94] Darbas D., "Commande floue appliquée à la régulation de niveau d'acier", Rapport DEA Automatique Industrielle, LAMII/CESALP, Université de Savoie, Annecy, France, Juillet 94.
- [Pernot95] Pernot L., "Utilisation de techniques floues pour la commande de procédés non linéaires", Rapport DEA Automatique Industrielle, LAMII/CESALP, Université de Savoie, Annecy, France, Juillet 95.
- [Boukezzoula96] Boukezzoula R., "Intégration de nouvelles fonctionnalités dans le langage PLICAS", Rapport de DEA Automatique Industrielle, LAMII/CESALP, Université de Savoie, Annecy, France, Juillet 96.
- [Passaquay96] Passaquay D., "Modélisation et commande de procédés à base de logique floue", Rapport de DEA Automatique et Informatique Industrielle, INSA Toulouse, stage effectué au LAMII/CESALP, Université de Savoie, Annecy, France, Juillet 96.
- [Prete97] Prete D., "Contrôleurs flous auto-adaptatifs - Analyse de différentes stratégies", Rapport de DEA Automatique Industrielle, Université de Savoie, Annecy, France, Juillet 97.
- [Farys99] Farys E., "Synthèse de contrôleurs flous", Rapport de DEA Automatique Industrielle, Université de Savoie, Annecy, France, Juillet 99.



## V Activités d'intérêt collectif

### V.1 Responsabilités administratives

Membre du CA de l'Université de Savoie (mandat 92-96).

Membre du CA de l'ESIA depuis 94.

Membre de la CSE 61<sup>ème</sup> section de l'Université de Savoie depuis 95.

Elue responsable de l'opération "Commande Experte", mandats 92-94 et 94-96.

- gestion financière des contrats et subventions
- élaboration et développement des divers axes de recherche
- membre de droit du Conseil de laboratoire

Elue représentant au Conseil de laboratoire au sein de l'équipe "Techniques Avancées de Commande", mandats renouvelés de 97 à 00.

Responsable du tronc commun 2<sup>ème</sup> année à l'ESIA, 97-99.

- gestion des emplois du temps
- organisation de l'enseignement
- accueil des étudiants

Responsable de la discipline informatique générale à l'ESIA, 93-95.

### V.2 Collaborations avec le secteur socio-économique

Entreprise	Nature	Objet	Période
Renault Recherche	Contrat de recherche	Diagnostic pour l'aide à la conduite	99-00
Région Rhône-Alpes	Programme de recherche Emergence	Holter tension intelligent	98-99
SERT	Contrat de recherche lié à une thèse CIFRE	Contrôle flou pour la régulation de niveau d'acier	95-98
ELF Solaize	Action commerciale	Conseils et expertises technologiques en logique floue	93-94
ESPRIT III	Working Group FALCON 6017	Algorithmes flous pour le contrôle	92-95

### V.3 Comités de programme et d'organisation de conférences

Comités d'organisation

3rd IFAC Symposium on Intelligent Components and Instruments for Control Applications (SICICA'97), Annecy.

7th Int. Conference on Information Processing and Management of Uncertainty in knowledge-based systems (IPMU'2002), Annecy.

Comités de programme

Rencontres francophones sur la Logique Floue et ses Applications (LFA'99, LFA'2000, LFA'2001).

#### **V.4 Participations régulières à la relecture d'articles de revues**

Fuzzy Sets and Systems  
Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems  
Journal of Intelligent and Fuzzy Systems  
IEEE Trans. on Fuzzy Systems  
IEEE Trans. on Systems, Man and Cybernetics  
IEEE Trans. on Control Systems Technology  
JESA

#### **V.5 Encadrement doctoral et participation à des jurys de thèse**

Bénéficiaire d'une prime d'encadrement doctoral depuis 95.

Hormis les thèses que j'ai co-encadrées, j'ai participé en tant qu'examineur au jury des thèses suivantes :

Alsaleh H., "Utilisation classique et extensions des fonctionnalités des processeurs flous", Université de Savoie, Nov. 98.

T. Chenaina, "Etude des paramètres de contrôleurs flous", Université Paris 6, Déc. 99.

#### **V.6 Relations internationales**

Intervenant à l'école d'été en automatique de l'Université Polytechnique de Bucarest, Roumanie, Mai 2001.

Séminaires à l'Université de Timisoara et à l'Université d'Arad, Roumanie, Mai 2001.

Accueil de doctorant, maître-assistant et/ou développement de travaux communs.

Anna-Maria Miklos (Université Polytechnique de Bucarest, Roumanie)

Valentina Balas (Université d'Arad, Roumanie)

Alexandre Evsukoff (Université Santa-Ursula, Rio de Janeiro, Brésil)

#### **V.7 Participation à des groupes de travail nationaux et régionaux**

Projet CSN (Commande Symbolique et Neuromimétique) du GR Automatique.

Présentation de travaux.

Animation du groupe en collaboration avec L. Foulloy.

Club CRIN, participation au groupe de travail "Logique floue dans les transports".

Collaboration LAG-LAMII à l'initiative de l'INPG et de l'Université de Savoie dans le cadre du Bonus Qualité Recherche, 96-99.

Logique floue pour le diagnostic.

Télémanipulation de procédés via Internet.

#### **V.8 Animation scientifique**

Séminaire Alcatel Alsthom Recherche, "Equivalence floue de contrôleurs classiques", Marcoussis, Juillet 93.

Intervenant au stage "Logique floue", Cours CNRS Formation, Gif sur Yvette, 93 et 94.

Séminaire Firmenich, "Logique floue et applications", Genève, Mars 2001.

## VI Publications

### VI.1 Articles de revue

- [R1] Foulloy L., Galichet S., “Fuzzy Sensor for Fuzzy Control”, *Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 2, N° 1, 1994, pp. 55-66.
- [R2] Galichet S., Foulloy L., “Fuzzy Controllers: Synthesis and Equivalences”, *IEEE Trans. on Fuzzy Systems*, Vol. 3, N° 2, May 1995, pp. 140-148.
- [R3] Foulloy L., Galichet S., “Evaluation des performances de contrôleurs flous à partir d’un benchmark”, *RAIRO-APII-JESA*, Vol. 30, N° 5, 1996, pp. 585-605.
- [R4] Dussud M., Galichet S., Foulloy L., “Application of Fuzzy Logic Control for Continuous Casting Mold Level Control”, *IEEE Trans. on Control Systems Technology*, Vol. 6, N° 2 (Special Issue on Control in the Metal Processing Industries), March 98, pp. 246-256.
- [R5] Charbonnier S., Galichet S., Mauris G., Siché J.P., “Statistical and Fuzzy Models of Ambulatory Systolic Blood Pressure for Hypertension Diagnosis”, *IEEE Trans. on Instrumentation and Measurement*, Vol. 49, N° 5, October 2000, pp. 998-1003.
- [R6] Boukezzoula R., Galichet S., Foulloy L., “Fuzzy Nonlinear Adaptive Internal Model Control (FNAIMC)”, à paraître dans *European Journal of Control*, Vol. 7, N° 5, 2001.
- [R7] Evsukoff A., Branco A.C.S., Galichet S., “Structure identification and parameter optimization for non-linear fuzzy modeling”, *Fuzzy Sets and Systems*, accepté avec révision mineure.

### VI.2 Ouvrages collectifs

- [O1] Foulloy L., Galichet S., “Typology of Fuzzy Controllers”, in *Theoretical Aspects of Fuzzy Control* (H.T. Nguyen, M. Sugeno, R. Tong, and R. Yager Eds), John Wiley & Sons, 1995, pp. 65-90.
- [O2] Galichet S., Foulloy L., “Fuzzy Control with Non-Precise Inputs”, in *Fuzzy Logic and Soft Computing* (B. Bouchon-Meunier, R. Yager and L. Zadeh Eds), World Scientific, 1995, pp 146-157, (republication de [C9] à la demande de l’éditeur).
- [O3] Nakoula Y., Galichet S., Foulloy L., “Identification of Linguistic Fuzzy Models Based on Learning”, in *Fuzzy Model Identification - Selected Approaches*, (H. Hellendoorn, D. Driankov Eds), Springer Verlag, 1997, pp. 281-319.
- [O4] Foulloy L., Galichet S., “Fuzzy and Linear Controllers”, in *FUZZY SYSTEMS Modeling and Control* (H.T. Nguyen, M. Sugeno Eds), *The Handbooks of Fuzzy Sets Series*, Kluwer Academic Publishers, 1998, pp. 197-225.
- [O5] Dubois D., Foulloy L., Galichet S., Prade H., “Performing approximate reasoning with words?”, in *Computing with Words in Information/Intelligent Systems 1 - Foundations*, (L.A. Zadeh, J. Kacprzyk Eds), Physica-Verlag, 1999, pp. 24-49.

### **VI.3 Thèse**

- [T1] Galichet S., "Etude d'un système d'incrustation d'images vidéo-couleur", Université de Technologie de Compiègne, Déc. 89.

### **VI.4 Brevet**

- [B1] Grimaldi J.L., Galichet S., "Procédé et dispositif d'incrustation d'images vidéo-couleur auto-adaptatif", brevet déposé au nom de Thomson Vidéo Equipement, No d'enregistrement national 89 13085, No de publication 2 652 972, Oct. 89.

### **VI.5 Communications dans des congrès internationaux avec actes et comité de lecture**

- [C1] Galichet S., Dubuisson B., Grimaldi J. L., "Procédé d'incrustation d'images vidéo-couleur auto-adaptatif", Proc of the 1st Int. Conf. dedicated to professional image chains (IMAGE'COM 90), Bordeaux, France, Nov. 90, pp. 398-402.
- [C2] Foulloy L., Galichet S., "Fuzzy sensors and controllers", Proc. of the 8th Int. Conf. on Applications of Artificial Intelligence in Engineering (AIENG 93), Toulouse, France, Juillet 1993, pp. 7-20.
- [C3] Foulloy L., Galichet S., "Fuzzy Controllers Representation", Proc. of the 1st European Congress on Fuzzy Intelligent Technologies (EUFIT 93), Aachen, Germany, Septembre. 93, pp. 142-148.
- [C4] Galichet S., Foulloy L., "How to derive symbolic information from measurement", Proc. of the Int. Conf. on Fault Diagnosis (TOOLDIAG 93), Toulouse, France, April 1993, pp. 938-943.
- [C5] Galichet S., Foulloy L., "Fuzzy Equivalence of Classical Controllers", Proc. of the 1st European Congress on Fuzzy Intelligent Technologies (EUFIT 93), Aachen, Germany, Septembre 1993, pp. 1567-1573.
- [C6] Foulloy L., Galichet S., Jossierand J.F., "Fuzzy Components for Fuzzy Control", Proc. of the 2nd IFAC Symp. on Intelligent Components and Instruments for Control Applications (SICICA '94), Budapest, Hungary, June 1994, pp. 122-126.
- [C7] Benoit E., Foulloy L., Galichet S., Mauris G., "Fuzzy Sensor for the Perception of Colour", Proc. of the IEEE Conf. on Fuzzy Systems (FUZZ-IEEE '94), Orlando, USA, June 1994, pp. 2008-2013.
- [C8] Galichet S., Foulloy L., Chebre M., Beauchene J., "Fuzzy Logic Control of a Floating Level in a Refinery Tank", Proc. of the 3rd IEEE Conf. on Fuzzy Systems (FUZZ-IEEE '94), Orlando, USA, June 1994, pp. 1538-1542.
- [C9] Galichet S., Foulloy L., "Fuzzy Control with Non Precise Inputs", Proc. of the 5th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge Based Systems (IPMU '94), Paris, France, July 1994, pp. 84-89.
- [C10] Foulloy L., Galichet S., Benoit E., "Fuzzy Control with Fuzzy State Sensors", Proc. of the 2nd European Congress on Fuzzy Intelligent Technologies (EUFIT 94), Aachen, Germany, Sept. 94, pp. 1156-1160.
- [C11] Galichet S., Foulloy L., "State Feedback Fuzzy Controllers", Proc. of the 2nd Euro-

- pean Congress on Fuzzy Intelligent Technologies (EUFIT 94), Aachen, Germany, Sept. 94, pp. 1161-1167.
- [C12] Foulloy L., Galichet S., "Representations of Mamdani's Controllers", Proc. of the 6th Int. Fuzzy System Association World Congress (IFSA 95), Sao Paulo, Brasil, July 95, pp. 581-584.
- [C13] Nakoula Y., Galichet S., Foulloy L., "Learning of a Fuzzy Symbolic Rulebase", Proc. of the 3rd European Congress on Intelligent Techniques and Soft Computing (EUFIT 95), Aachen, Germany, August 1995, pp. 594-598.
- [C14] Josserand J.F., Galichet S., Benoit E., Foulloy L., "Fuzzy cells network for fuzzy control with a fuzzy state sensor", Proc. of the 3rd European Congress on Intelligent Techniques and Soft Computing (EUFIT 95), Aachen, Germany, August 1995, pp. 910-914.
- [C15] Carron T., Lambert P., Galichet S., "A colour region-growing segmentation with fuzzy aggregation", Proc. of the 3rd European Congress on Intelligent Techniques and Soft Computing (EUFIT 95), Aachen, Germany, August 1995, pp. 1366-1372.
- [C16] Dussud M, Simonin P., Galichet S., Foulloy L., "Fuzzy logic control of continuous casting installations", Proc. of the 13th IFAC World Congress, San Francisco, USA, July 96, Vol. M, pp 469-474.
- [C17] Josserand J.F., Galichet S., Foulloy L., "Implementation of fuzzy controllers with fuzzy components", Proc. of the 13th IFAC World Congress, San Francisco, USA, July 96, Vol. A, pp 415-420.
- [C18] Dussud M., Galichet S., Foulloy L., "Design and implementation of a fuzzy-logic controller for brass casting control", Proc. of the 4th European Congress on Intelligent Techniques and Soft Computing (EUFIT'96), Aachen, Germany, Sept. 96, pp. 1230-1236.
- [C19] Nakoula Y., Galichet S., Foulloy L., "Simultaneous Learning of Rules and Linguistic Terms", Proc. of the 5th IEEE Int. Conf. on Fuzzy Systems (FUZZ'IEEE 96), New Orleans, USA, Sept. 96, pp. 1743-1749.
- [C20] Dubois D., Foulloy L., Galichet S., Prade H., "Two different views of approximate reasoning", Proc. of the 7th Int. Fuzzy Systems Association World Congress (IFSA'97), Prague, Czech Republic, June 97, Vol. 1, pp. 238-242.
- [C21] Galichet S., Passaquay D., Foulloy L., "Rulebase structure identification or reduction in the framework of fuzzy learning", Proc. of the 7th Int. Fuzzy Systems Association World Congress (IFSA'97), Prague, Czech Republic, June 97, Vol. 1, pp. 363-368.
- [C22] Boukezzoula R., Galichet S., Foulloy L., "Fuzzy Input-Output Linearization of Non-linear Systems", Proc. of the 5th European Congress on Intelligent Techniques and Soft Computing (EUFIT'97), Aachen, Germany, Sept. 97, Vol. 2, pp. 1316-1320.
- [C23] Foulloy L., Galichet S., Combastel C., Gentil S., "Fuzzy Processing for Electrical Machines", Proc. of the IEEE Int. Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives (SDEMPED'97), Carry-le-Rouet, France, Sept. 97, pp. 94-101.

- [C24] Boukezzoula R., Galichet S., Foulloy L., “Fuzzy Learning Control of Nonlinear Systems Using Input-Output Linearization”, Proc. of the 1998 IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC’98), San Diego, California, USA, Oct. 98, pp. 2095-2100.
- [C25] Galichet S., Foulloy L., “Size Reduction in Fuzzy Rulebases”, Proc. of the 1998 IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC’98), San Diego, California, USA, Oct. 98, pp. 2107-2112.
- [C26] Boukezzoula R., Galichet S., Foulloy L., “Fuzzy Control of Nonlinear Systems Using Two Standard Techniques”, (FUZZ-IEEE’99), Seoul, Korea, August 99, Vol. II, pp. 875-880.
- [C27] Boukezzoula R., Galichet S., Foulloy L., “Fuzzy Nonlinear Adaptive Internal Model Control (FNAIMC) - Part I: Fuzzy Model Identification”, (ECC’99), Karlsruhe, Germany, Sept. 99, CDROM n° F874.
- [C28] Boukezzoula R., Galichet S., Foulloy L., “Fuzzy Nonlinear Adaptive Internal Model Control (FNAIMC) - Part II: Fuzzy Model Inversion”, (ECC’99), Karlsruhe, Germany, Sept. 99, CDROM n° F875.
- [C29] Charbonnier S., Galichet S., Mauris G., Siché J.P., “Statistical and Fuzzy Models of Ambulatory Systolic Blood Pressure for Hypertension Diagnosis”, Proc. of the 16th IEEE Int. Instrumentation and Measurement Technology Conf. (IMTC’99), Venice, Italy, Mai 99, pp. 103-108.
- [C30] Foulloy L., Galichet S., “Another Formulation of the Semantics of Fuzzy Rules”, Proc. of the IAR-ICD’99 Workshop on Intelligent Techniques for Information Processing and Applications, Nancy, France, Dec. 99, pp. 21-24.
- [C31] Dussud M., Galichet S., Foulloy L., "Fuzzy supervision for continuous casting mold level control", Proc. of the 2nd Conf. IFAC/IFIP/IEEE on Management and Control of Production and Logistics (MCPL2000), Grenoble, July 2000, CDROM P361.
- [C32] Galichet S., Charbonnier S., Mauris G., Siché J.P., “A Fuzzy Linguistic Model for Ambulatory Systolic Blood Pressure Variation Prediction”, Proc. of the IEEE Int. Conf. on Fuzzy Systems (FUZZ’IEEE 2000), San Antonio (TX), USA, May 2000, pp. 522-527.
- [C33] Miklos A.M., Galichet S., Foulloy L., Popescu D., “Fuzzy supervision for temperature control in a steam-electric power plant”, Proc. of the 2000 IAR-ICD workshop on Intelligent Control and Diagnosis, Nancy, Nov. 2000, pp. 7-12.
- [C34] Boukezzoula R, Galichet S., Foulloy L., “Exact inversion of Takagi-Sugeno fuzzy models”, Proc. of the 9th IFSA World Congress (IFSA2001), Vancouver, Canada, July 2001, pp. 2108-2113.
- [C35] Boukezzoula R, Galichet S., Foulloy L., “Robust fuzzy control for a class of continuous nonlinear systems using input-output linearization : Real-time implementation for a robot wrist”, Proc. of the 2001 IEEE Int. Conf. on Control Applications (CCA2001), Mexico, Mexique, Sept. 2001, pp. 311-316.
- [C36] Foulloy L., Galichet S., Boukezzoula R, “Model-based or heuristic-based fuzzy logic

controllers? Foundations and examples”, IFAC Workshop on Advanced Fuzzy-Neural Control 2001, Valence, Espagne, Oct. 2001, pp. 197-210.

- [C37] Boukezzoula R, Galichet S., Foulloy L., “Fuzzy adaptive control for nonlinear systems : Real-time implementation for a robot wrist”, 40th IEEE Conf. on Decision & Control (CDC2001), Orlando, USA, Dec. 2001, pp. 4364-4370.
- [C38] Boukezzoula R, Galichet S., Foulloy L., “Fuzzy robust control for discrete-time nonlinear systems using input-output linearization and  $H_{\infty}$  optimization”, 10th IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE2001), Melbourne, Australie, Dec 2001, P244.
- [C39] Evsukoff A., Branco A.C.S., Galichet S. , “Incremental fuzzy modelling”, 10th IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE2001), Melbourne, Australie, Dec 2001, P424.

## **VI.6 Communications dans des congrès nationaux avec actes et comité de lecture**

- [N1] Foulloy L., Galichet S., “Représentation des contrôleurs flous”, Actes des 2ièmes Journées Nationales sur les Applications des Ensembles Flous, Nîmes, France, Nov. 1992, pp. 129-136.
- [N2] Galichet S., Dussud M., Foulloy L., “Contrôleurs flous : équivalences et études comparatives”, Actes des 2ièmes Journées Nationales sur les Applications des Ensembles Flous, Nîmes, France, Nov. 1992, pp. 229-236.
- [N3] Carron Th., Lambert P., Galichet S., “Segmentation en régions d'images couleur par agrégation floue”, Actes des 4ièmes Journées Nationales sur les Applications des Ensembles Flous, Lille, France, Déc. 94, pp. 203-212.
- [N4] Nakoula Y., Galichet S., Foulloy L., “Modélisation symbolique floue par jeu de règles pondérées”, Actes des rencontres francophones sur la Logique Floue et ses Applications (LFA'95), Paris, France, Nov. 1995, pp. 57-64.
- [N5] Boukezzoula R., Galichet S., Foulloy L., “Commande à modèle interne flou”, Actes des rencontres francophones sur la Logique Floue et ses Applications (LFA'97), Lyon, France, Déc. 97, pp. 11-16.
- [N6] Boukezzoula R., Galichet S., Foulloy L., “Apprentissage de lois de commande floues pour des systèmes non linéaires (synthèse directe et indirecte)”, Actes des rencontres francophones sur la Logique Floue et ses Applications (LFA'98), Rennes, France, Nov. 98, pp. 19-27.
- [N7] Miklos A.M., Galichet S., Foulloy L., Popescu D., “Supervision floue d'une chaîne de régulation de température dans une centrale thermoélectrique”, Actes des rencontres francophones sur la Logique Floue et ses Applications (LFA'2000), La Rochelle, France, Oct. 2000, pp. 157-164.
- [N8] Galichet S., Trouvé E., “Communication entre applications sous Labview”, Actes du 3ième Colloque sur l'Enseignement des Technologies et des Sciences de l'Information et des Systèmes (CETSI'2001), Clermont-Ferrand, France, Oct. 2001, pp. 161-164.
- [N9] Boukezzoula R., Galichet S., Foulloy L., “Quelques propriétés des systèmes flous de

Takagi-Sugeno à conclusions constantes”, Actes des rencontres francophones sur la Logique Floue et ses Applications (LFA'2001), Mons, Belgique, Nov. 2001, pp. 241-248.

#### **VI.7 Divers (rapports internes - conférences sans comité de lecture)**

- [D1] Galichet S., Gounon P., “Bibliothèque SIMPA, Signal et IMage PAcKage”, Ass. Générale du GdR Traitement du Signal et Images, Batz sur Mer, Avril 91.
- [D2] Galichet S., “Guide d'utilisation de SIMPA - Bibliothèque image”, GdR Traitement du Signal et Images, Sept. 91, 104 pages.
- [D3] Galichet S., “SIMPA : Signal et IMage PAcKages”, Actes des journées UNIX de Grenoble, Grenoble, Nov. 91.
- [D4] Foulloy L., Galichet S. , “Contrôleurs flous: représentation, équivalences et études comparatives”, Rapport interne LAMII 92-04, Univ. de Savoie, Annecy, France, Juillet 1992, 14 pages. (Ce rapport a servi de base pour [N1] et [N2]).
- [D5] Galichet S., Foulloy L., “Fuzzy Control with Non-Precise Inputs”, Rapport interne LAMII 93-04, Univ. de Savoie, Annecy, Juillet 1993, 7 pages. (Ce rapport a servi de base pour [C9]).
- [D6] Galichet S., Foulloy L., “Fuzzy controllers: Synthesis and Equivalences”, Rapport interne LAMII 93-08, Univ. de Savoie, Annecy, France, Novembre 1993, 22 pages. (Ce rapport a servi de base pour [R2]).
- [D7] Galichet S., Foulloy L., “Evaluation du logiciel TILGEN”, Rapport interne LAMII 94-01, Univ. de Savoie, Annecy, France, Février 94, 11 pages.
- [D8] Galichet S., Mauris G., “Module de prédiction des temps de passage des rapports pour l'aide à la conduite”, Rapport de fin de contrat, Renault DR, Mars 2000.
- [D9] Galichet S., “Expression analytique unique de la sortie d'un contrôleur flou de Mamdani”, Rapport interne LAMII 01-02, Université de Savoie, Octobre 2001.



## **Partie B : Travaux de recherche**



## Introduction générale

*“First, computing with words is a necessity when the available information is too imprecise to justify the use of numbers, and second, when there is a tolerance for imprecision which can be exploited to achieve tractability, robustness, low solution cost, and better report with reality. Exploitation of the tolerance for imprecision is an issue of central importance in computing with words.”*

Cette citation extraite de [Zadeh96] situe le contexte général dans lequel s’inscrivent nos travaux. Appliquer à la commande des procédés l’idée de calculer avec les mots, telle est la démarche que nous avons suivie. Bien évidemment, le succès de l’entreprise est lié à une compréhension fine des contrôleurs flous, et plus particulièrement de la dualité numérique/linguistique sur laquelle ils reposent. C’est d’ailleurs cette dernière qui nous a conduits à structurer le présent document en deux chapitres.

Ainsi, le premier s’inscrit clairement dans une problématique numérique puisque les systèmes flous sont utilisés comme outil de représentation d’une fonction entrées/sortie numérique. L’objectif poursuivi est alors de fournir une méthode de construction du système flou permettant d’assurer une approximation satisfaisante du transfert entrées/sortie désiré. La stratégie proposée repose sur une technique d’interpolation, mise en oeuvre via le principe des équivalences modales. Celui-ci, développé au sein de l’équipe pour traiter des problèmes linéaires de faible dimension, est ici généralisé à des fonctions quelconques, dont la connaissance peut même n’être que partielle. De fil en aiguille, le premier chapitre propose ainsi un itinéraire conduisant des PID flous à la commande adaptative robuste des systèmes non linéaires par linéarisation entrées/sortie floue.

Si la démarche sous-jacente à l’écriture de ce premier chapitre nous a conduits à l’intituler “Interpolation numérique”, il est possible de lui trouver une autre clé de lecture. Ainsi, aurait-on pu le dénommer “De Mamdani à Sugeno” puisqu’il reflète le cheminement qu’a connu la discipline au cours des dix dernières années. Petit à petit, l’approche préconisée par Mamdani dès les années 70 a pratiquement disparu au profit d’une recherche et d’une garantie de performances numériques. Dans ce contexte, le formalisme analytique autorisé par le mécanisme de calcul des systèmes de Sugeno paraît mieux adapté puisqu’il permet une exploitation pratiquement immédiate des outils de la commande conventionnelle.

L’aspect linguistique est abordé dans le deuxième chapitre intitulé “Codage de l’expertise”. L’idée directrice est maintenant d’ajouter aux systèmes flous, développés selon une approche numérique, une dimension linguistique leur conférant lisibilité, transparence et pouvoir explicatif. L’ensemble des travaux présentés dans ce contexte repose sur une implantation entière-

ment symbolique des systèmes flous. Cette approche originale, mise en évidence par la typologie des contrôleurs flous que nous avons établie, nous permet d'assurer une implantation exacte des systèmes de Sugeno à conclusions constantes. Cette propriété n'est cependant valable que sous certaines hypothèses qui justifient a posteriori les contraintes que nous nous sommes imposées sur la synthèse des systèmes flous dans le premier chapitre. La cohérence ainsi établie entre les approches numérique et linguistique nous autorise à envisager divers schémas d'intégration des connaissances expertes disponibles sur le procédé à piloter. Il devient notamment possible d'intégrer une expertise dans un contrôleur numérique existant. Le choix d'une architecture globale d'implantation de la loi de commande finale se pose alors plus en termes de contraintes technologiques, stratégiques et économiques qu'en termes scientifiques.

Ce second chapitre se pose en pendant du premier et sa seconde clé de lecture pourrait être "De la théorie aux applications". En effet, les travaux conceptuels développés, mais aussi leur exploitation dans le cadre d'applications industrielles, nous amènent finalement à établir une démarche méthodologique au codage de l'expertise dans un contexte numérique.

Avant d'entrer dans le vif du sujet, il ne nous reste plus ici qu'à préciser quelques conventions que nous avons fixées et qui pourront sans doute éclairer la lecture des pages à venir.

Les paragraphes, écrits en italique et situés en retrait, présentent les idées directrices ou interrogations qui guident notre discours.

Les références bibliographiques, données au fil du texte, sont répertoriées en fin de document. Celles qui correspondent à des travaux développés au sein de l'équipe (articles de revue, congrès et thèses encadrées) sont différenciées par le fait qu'elles apparaissent en caractères gras au moment de leur citation.

# Chapitre I : Interpolation numérique

## I Introduction

Dans un cadre applicatif, l'élaboration d'un contrôleur flou est encore bien souvent appréhendée de façon empirique avec une démarche d'essais/erreurs. Une telle pratique est sans doute liée à l'absence d'une méthodologie simple de synthèse d'un contrôleur flou. Cette constatation ne date bien évidemment pas d'aujourd'hui. Dès les années 90, elle a d'ailleurs conduit de nombreux chercheurs à tenter d'établir un "pont" entre théorie du contrôle et contrôle flou. La problématique est alors classiquement posée en les termes suivants :

*Etant donné un contrôleur flou entièrement défini (base de règles, partitions floues associées aux variables numériques, opérateurs d'inférence, méthode de défuzzification), quelle est la forme analytique de la loi de commande associée ?*

Réponse a été donnée à cette question pour différents types de contrôleurs flous. En particulier, les contrôleurs flous de type P, PI, PD et PID, basés sur l'exploitation d'un signal d'erreur entre la sortie du procédé et la consigne, ont été abondamment étudiés. Les premiers travaux sur le sujet ([Siler89], [Ying90]) concernent l'analyse de bases de règles à deux entrées de taille restreinte (4 ou 9 règles). Ils mettent en évidence le fait que si aucune précaution particulière n'est prise lors de l'implantation du contrôleur flou, la commande engendrée présente des non-linéarités qui se superposent à un comportement linéaire global. Reste alors à étudier l'impact des non-linéarités générées et la façon dont celles-ci peuvent être exploitées pour satisfaire différents objectifs de régulation (stabilité, robustesse, ...). Si cette perspective est envisagée dans [Ying90], les premiers éléments de réponse avancés pour une base de 4 règles restent sans suite. Ils s'avèrent en fait difficilement généralisables à un nombre de règles plus élevé. Bien que non énoncée explicitement, la raison peut être trouvée dans des travaux ultérieurs ([Ying93], [El Hajjaji94], [El Hajjaji95]) qui donnent la forme analytique de la loi de commande obtenue avec une base de règles de taille quelconque. En fait, les non-linéarités générées sont similaires sur toutes les mailles floues et se reproduisent donc à l'identique dans les différentes zones de l'espace de phase. Leur exploitation devient alors délicate dans la mesure où aucun traitement local ne peut être effectué.

Les non-linéarités évoquées précédemment sont en réalité dues aux opérateurs d'inférence et à la méthode de défuzzification choisis. N'étant pas ajustables, elles peuvent apparaître comme indésirables. Elles risquent notamment de modifier d'autres non-linéarités créées volontairement par le biais des règles ou des fonctions d'appartenance. Apparaît alors la volonté de trouver un mécanisme de calcul qui garantisse la linéarité de l'interpolation sur une maille floue. Dans ce contexte, le choix du produit et de la somme bornée comme opérateurs d'inférence

permet de satisfaire la contrainte de linéarité lorsque la défuzzification est effectuée par la méthode du centre de gravité ou par la méthode des hauteurs ([Mizumoto93], [Galichet95], [Moon95]).

En résumé, toutes les études citées précédemment ont contribué à déterminer une forme de contrôleurs flous équivalents à des contrôleurs linéaires. Elles ont toutes en commun le fait que la base de règles analysée est supposée additive. Cette hypothèse est justifiée a posteriori puisque la loi de commande correspondante est effectivement linéaire lorsque les opérateurs flous sont correctement choisis. Elle est également confortée par le fait que la loi de commande limite, obtenue lorsque le nombre de symboles utilisés tend vers l'infini, s'avère linéaire indépendamment des opérateurs d'inférence utilisés ([Buckley89], [Bouslama92]). Bien que validée dans un contexte linéaire, l'hypothèse de règles additives, ou plus généralement de toute forme figée de règles, n'en demeure pas moins un handicap majeur à la généralisation de la démarche.

La stratégie que nous avons initiée puis développée s'affranchit de toute contrainte de forme sur les règles. Elle repose sur un nouveau principe, appelé principe des équivalences modales, introduit dans ([Galichet92], [Galichet93], [Galichet95]) puis exploité pour différents types de problèmes ([Galichet94a], [Galichet94b], [Boukezzoula97], [Foulloy98]). Le principe des équivalences modales, fil conducteur de ce chapitre, a été développé à l'origine dans l'objectif de synthétiser une loi de commande linéaire avec un contrôleur flou. En ce sens, il rejoint l'ensemble des travaux publiés à la même époque et aboutit bien évidemment à la même conclusion. Il diffère cependant de l'ensemble des études évoquées ci-dessus par le fait que la démarche envisagée est constructive et non plus restreinte à l'analyse d'un contrôleur flou existant. En d'autres termes, contrairement aux approches analytiques qui reposent toutes sur des règles additives, le principe des équivalences modales ne nécessite aucune hypothèse quant à la forme des règles. En contrepartie, la loi de commande que l'on cherche à obtenir est supposée connue. Ainsi, la problématique initiale est inversée et se pose en les termes suivants :

*Etant donnée une loi de commande cible, quel est le contrôleur flou qui permet de l'approcher ?*

Si à l'origine la stratégie de construction du contrôleur flou a été développée pour une loi linéaire, elle n'est pas pour autant dédiée à ce type de cible. Elle permet en effet d'aborder la génération de lois de commande quelconques, mais aussi l'apprentissage d'une base de règles à partir d'exemples. De manière générale, le principe des équivalences modales peut être vu comme un outil de construction d'un système flou, l'objectif étant de réaliser un transfert entrées/sortie numérique connu (forme analytique) ou partiellement connu (exemples).

A travers ce chapitre, sont présentées différentes utilisations possibles du principe des équivalences modales. Les références bibliographiques relatives aux domaines abordés sont insérées dans les paragraphes correspondants.

## **II Contrôleurs flous numériques**

La synthèse de lois de commande est envisagée pour deux types particuliers de contrôleurs flous, à savoir les contrôleurs de Mamdani et ceux de Sugeno. Bien qu'ils soient équivalents sous certaines conditions, ils sont dans un premier temps clairement dissociés et traités indépendamment les uns des autres. Ce choix est avant tout lié au fait que ces deux types de systèmes sont généralement exploités dans des contextes différents. En effet, les systèmes de Sugeno sont souvent associés à une recherche de performance numérique alors que ceux de Mamdani sont orientés vers une possible prise en compte de connaissances expertes. Dans les deux cas, la re-

cherche d'équivalence avec une loi cible peut donc être vue comme une phase d'initialisation du système flou qui devra ensuite être affiné.

Dans ce qui suit, la présentation des deux types de systèmes flous est réduite à celle des mécanismes de calcul associés. En effet, l'objectif visé dans un premier temps étant simplement la réalisation d'un transfert entrées/sortie numérique connu, nul besoin est d'étudier en détails le fonctionnement interne des contrôleurs flous considérés. Enfin, pour alléger les notations, la présentation est restreinte à des systèmes à deux entrées,  $x$  et  $y$ , et une sortie  $z$ .

## II.1 Contrôleurs flous de Mamdani

Un système flou de Mamdani est basé sur une collection de règles du type :

$$R_{(i,j,k)} : \mathbf{Si} \ x \text{ est } A_i \ \mathbf{et} \ y \text{ est } B_j \ \mathbf{alors} \ z \text{ est } C_k, \quad (1)$$

où  $A_i, i \in I_1, B_j, j \in I_2$ , et  $C_k, k \in I_3$ , représentent des termes linguistiques auxquels est attribuée une signification floue ([Zadeh71]) que l'on suppose normalisée<sup>(1)</sup>. Le jeu de règles est donné par la connaissance de l'ensemble  $I$  des triplets  $(i,j,k)$ , il sera dit complet si :

$$\forall (i,j) \in I_1 \times I_2 \quad \exists k \in I_3 \ / \ (i,j,k) \in I \quad (2)$$

Pour des entrées numériques précises,  $x_0$  et  $y_0$ , un sous-ensemble flou de sortie, noté  $F$ , est généré selon la relation :

$$\mu_F(z) = \max_{(i,j,k) \in I} \min(\mu_{A_i}(x_0), \mu_{B_j}(y_0), \mu_{C_k}(z)). \quad (3)$$

L'opération  $\max_{(i,j,k) \in I}$  correspond à l'application de l'opérateur max sur l'ensemble des règles définies. La valeur numérique de sortie délivrée par le système flou est obtenue par défuzzification, classiquement implantée par la méthode du centre de gravité :

$$z = \text{MAM}(x_0, y_0) = \frac{\int z \cdot \mu_F(z) \cdot dz}{\int \mu_F(z) \cdot dz}. \quad (4)$$

L'équation (3) repose sur l'utilisation des opérateurs flous min/max préconisés par Mamdani. Elle peut être généralisée en introduisant des opérateurs quelconques de type t-norme/t-conorme (voir par exemple [Dubois87], [Bouchon93]) et devient alors :

$$\mu_F(z) = \perp_{(i,j,k) \in I} \ T_2(T_1(\mu_{A_i}(x_0), \mu_{B_j}(y_0)), \mu_{C_k}(z)), \quad (5)$$

où  $T_1$  et  $T_2$  représentent des t-normes et  $\perp$  une t-conorme. Il est à noter que l'équation (5) correspond à une implantation conjonctive du "Si ... alors ..." et que le cas des graphes implicatifs, formulé dans [Foulloy95a] ou [Foulloy96], n'est donc pas abordé dans ce document.

## II.2 Contrôleurs flous de Sugeno

Les contrôleurs de Sugeno sont, comme ceux de Mamdani, construits à partir d'une base de règles "Si ... alors ...". Si les prémisses sont toujours exprimées linguistiquement et donc similaires à celles utilisées dans (1), les conclusions sont par contre de nature numérique. Elles peuvent prendre différentes formes : équations linéaires ou polynômiales, statiques ou dynamiques. L'étude est ici restreinte à des conclusions constantes, les règles étant donc formulées de la manière suivante :

$$R_{(i,j,k)} : \mathbf{Si} \ x \text{ est } A_i \ \mathbf{et} \ y \text{ est } B_j \ \mathbf{alors} \ z = c_k, \quad (6)$$

---

<sup>(1)</sup> Un sous-ensemble flou  $F$ , défini sur  $X$ , est normalisé si et seulement si

$$\exists x_m \in X \ / \ \mu_F(x_m) = 1.$$

Dans ce cas,  $x_m$  est appelé valeur modale de  $F$ .

où  $c_k$  représente une valeur réelle quelconque. Pour les valeurs d'entrée précises  $x_0$  et  $y_0$ , la sortie  $z$  est évaluée selon le mécanisme de calcul suivant :

$$z = \text{SUG}(x_0, y_0) = \frac{\sum_{(i,j,k) \in I} w_{i,j} \cdot c_k}{\sum_{(i,j,k) \in I} w_{i,j}} \quad \text{avec} \quad w_{i,j} = \mu_{A_i}(x_0) \cdot \mu_{B_j}(y_0). \quad (7)$$

### III Equivalences modales

La synthèse du système flou est envisagée selon une approche interpolative. L'objectif fixé est alors similaire à celui de toute méthode d'interpolation, à savoir :

*Etant donnée une forme de système (linéaire, polynomial, ..., flou, ...), comment fixer les paramètres de celui-ci pour qu'il "passe" exactement en un certain nombre de points fixés ?*

Le contrôleur flou est ainsi construit selon le mécanisme illustré à la Fig. 1. Par égalisation de la sortie désirée (sortie du contrôleur cible) avec celle du contrôleur flou, sont déduites des contraintes permettant de rétroagir sur le système flou à générer.

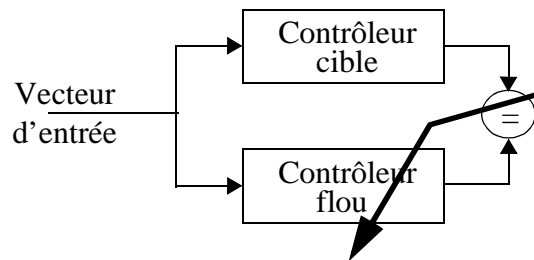


Fig. 1 : Stratégie de construction du contrôleur flou.

La mise en oeuvre de cette méthodologie nécessite néanmoins de pouvoir exprimer paramétriquement la sortie du système flou pour les vecteurs d'entrée considérés. La principale difficulté est alors liée à la nature même du système à construire, non exprimable sous une forme analytique simple, tout au moins en ce qui concerne les systèmes de Mamdani. Une façon de contourner le problème consiste à se restreindre dans le choix des vecteurs d'entrée et à ne considérer que des entrées modales, d'où l'appellation "équivalences modales". Ce choix repose en fait sur la constatation que pour des entrées modales, la sortie inférée par le contrôleur flou est généralement évaluable de façon simple. En d'autres termes, les opérations de t-conorme ou de sommation intervenant dans les équations (5) ou (7) se réduisent souvent à la prise en compte d'un seul terme facilement identifié, permettant ainsi de mettre en évidence une propriété des contrôleurs flous aux valeurs modales.

#### III.1 Propriété aux valeurs modales

On suppose que les significations floues définies sur  $X$  et  $Y$  sont normalisées et forment des partitions strictes<sup>(1)</sup>. Soient  $VX$  et  $VY$  l'ensemble des valeurs modales associées à ces partitions.

---

<sup>(1)</sup> Un ensemble de termes  $\{A_i\}_{i=1, \dots, n}$  assure un partitionnement strict de  $X$  si et seulement si :

$$\forall x \in X \quad \sum_{i=1}^n \mu_{A_i}(x) = 1.$$



Suivant le type de contrôleurs flous considéré, la propriété **[a]** ou **[b]** peut être établie.

**[a]** Si les significations floues définies sur  $U$  sont symétriques et unimodales, tout contrôleur de Mamdani dont le jeu de règles est complet est tel que :

$$\forall (vx, vy) \in VX \times VY, \exists vu \in VU \text{ tel que } vu = \text{MAM}(vx, vy),$$

où  $VU$  est l'ensemble des valeurs modales associées à  $U$ .

**[b]** Tout contrôleur de Sugeno dont le jeu de règles est complet est tel que :

$$\forall (vx, vy) \in VX \times VY, \exists cu \in CU \text{ tel que } cu = \text{SUG}(vx, vy),$$

où  $CU \subset \mathfrak{R}$  est l'ensemble des constantes utilisées dans les conclusions de règles.

En effet, soit  $x_m$  une valeur modale du symbole  $A_m$ , et  $y_n$  une valeur modale du symbole  $B_n$ , on a alors  $\mu_{A_m}(x_m) = \mu_{B_n}(y_n) = 1$ . Si l'on suppose des partitions strictes des univers de discours associés aux entrées, on a de plus  $\mu_{A_i}(x_m) = \mu_{B_j}(y_n) = 0$  pour tout  $i$  différent de  $m$  et  $j$  différent de  $n$ . La t-conorme ou la sommation sur toutes les règles (équations (5) et (7)) se ramène alors à la seule valeur non nulle, fournie par la règle de prémisse "Si  $x$  est  $A_m$  et  $y$  est  $B_n$ ".

Dans le cas d'un contrôleur de Sugeno, le résultat final est immédiat :

$$z = \text{SUG}(x_m, y_n) = c_l \quad \text{avec } (m, n, l) \text{ triplet indexant la seule règle activée.} \quad (8)$$

Pour un contrôleur de Mamdani, l'équation (3) devient tout simplement :

$$\mu_F(z) = \mu_{C_l}(z) \quad \text{avec } (m, n, l) \text{ triplet indexant la seule règle activée.} \quad (9)$$

Si l'on considère que les symboles de sortie ont des significations floues, symétriques et unimodales, la méthode du centre de gravité conduit finalement à :

$$z = \text{MAM}(x_m, y_n) = z_l \quad (10)$$

avec  $z_l$  valeur modale du symbole  $C_l$  et  $(m, n, l)$  triplet indexant la seule règle activée.

La Fig. 2 résume la propriété des contrôleurs flous aux valeurs modales pour les deux types de règles considérés. Quoique restrictives, les conditions de validité de **[a]** et **[b]** sont très largement vérifiées dans les applications décrites dans la littérature.

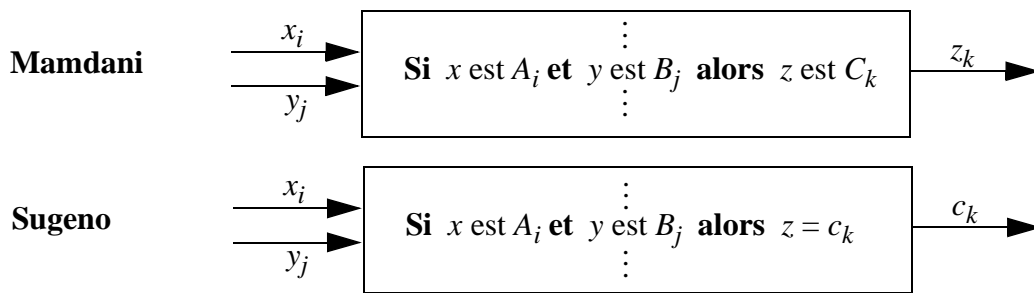


Fig. 2 : Comportement d'un contrôleur flou pour des entrées modales.

### III.2 Principe des équivalences modales

Soit le contrôleur cible défini par la fonction  $h: X \times Y \rightarrow U$ .

Le principe des équivalences modales consiste à trouver un contrôleur de Mamdani ou de Sugeno tel que :

$$\forall (vx, vy) \in VX \times VY, \text{MAM}(vx, vy) = h(vx, vy) \\ \text{ou } \text{SUG}(vx, vy) = h(vx, vy).$$

## IV Application du principe des équivalences modales à la synthèse d'un contrôleur flou de type PI

L'objectif de ce paragraphe est d'illustrer le principe des équivalences modales sur le cas particulier d'un contrôleur cible de type Proportionnel Intégral (PI).

L'équation en  $z$  du contrôleur cible est donc de la forme :

$$C(z) = \frac{u(z)}{\varepsilon(z)} = K_P + K_I \cdot \left( \frac{z}{z-1} \right), \quad (11)$$

où  $K_P$  et  $K_I$  représentent les gains, supposés connus, du contrôleur PI pour lequel un équivalent flou est recherché. Si l'on note respectivement  $\delta\varepsilon$  et  $\delta u$  les variations de l'erreur  $\varepsilon$  et de la commande  $u$ , l'équation (11) se traduit dans le domaine temporel par :

$$\delta u = K_P \delta\varepsilon + K_I \varepsilon. \quad (12)$$

D'un point de vue boîte noire, le contrôleur PI peut donc être assimilé à un bloc à qui l'on fournit deux entrées ( $\varepsilon$  et  $\delta\varepsilon$ ) et qui produit une sortie ( $\delta u$ ). Il paraît tout à fait naturel de conserver les mêmes entrées et sortie pour le contrôleur flou équivalent.

### IV.1 Partitionnement des univers de discours

Les entrées et sortie du contrôleur flou que l'on cherche à synthétiser étant définies, il s'agit maintenant de déterminer les symboles qui seront utilisés pour l'écriture de la base de règles. Leur exploitation ultérieure dans un monde numérique nécessite également de leur attribuer une signification floue. Cette étape correspond au partitionnement flou des différents univers numériques, chaque symbole étant représenté par un sous-ensemble flou défini sur l'univers de discours adéquat. Le contrôleur cible étant linéaire, il convient d'éviter au maximum d'introduire des non-linéarités dans le contrôleur flou. Dans cette optique, un partitionnement triangulaire régulier est effectué sur chacun des univers de discours.

La Fig. 3 illustre ce choix pour la variable d'erreur. La notation  $E_i$  représente le  $i$ ème symbole attaché à l'erreur.

En notant  $\Delta a$  l'écartement entre deux fonctions d'appartenance consécutives, la valeur modale du symbole  $E_i$ , notée  $\varepsilon_i$ , s'exprime selon la relation :

$$\varepsilon_i = i \Delta a + \varepsilon_0. \quad (13)$$

Un choix similaire pour les autres variables conduit à la la définition des symboles  $dE_j$  et  $dU_k$  dont les valeurs modales respectives vérifient :

$$\delta\varepsilon_j = j \Delta b + \delta\varepsilon_0 \quad (14)$$

$$\delta u_k = k \Delta c + \delta u_0 \quad (15)$$

avec  $\Delta b$  et  $\Delta c$  écarts respectifs entre deux fonctions d'appartenance consécutives relatives à la variation d'erreur et à la variation de commande. Par la suite, sans perte de généralité mais pour une simplification des calculs, la distribution des fonctions d'appartenance sera supposée symétrique par rapport à zéro, c'est-à-dire  $\varepsilon_0 = \delta\varepsilon_0 = \delta u_0 = 0$ . Il est à noter que la définition des symboles  $dU_k$  ne sera exploitée que dans le cas des contrôleurs flous de Mamdani.

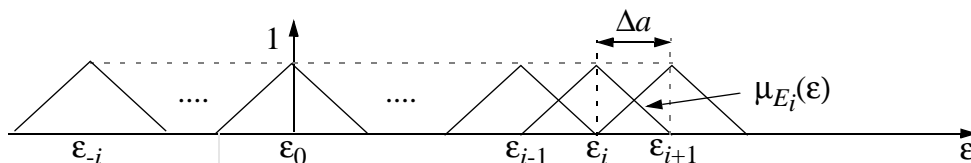


Fig. 3 : Partitionnement flou de l'univers de discours associé à l'erreur.

## IV.2 Synthèse d'un contrôleur flou de type Mamdani

### IV.2.1 Construction de la base de règles

Du choix des entrées et sortie du contrôleur flou et du partitionnement des univers de discours découle l'expression générique des règles qui, dans le cas d'un contrôleur de type Mamdani, est donnée par :

$$\text{Si } \varepsilon \text{ est } E_i \text{ et } \delta\varepsilon \text{ est } dE_j \text{ alors } \delta u \text{ est } dU_{f(i,j)}. \quad (16)$$

Aucune hypothèse n'étant faite sur le nombre de symboles, et donc de règles, les indices  $i$  et  $j$  sont considérés comme appartenant à  $\mathbb{Z}$ . A chaque règle, indiquée par le couple  $(i,j)$ , est associé un symbole de sortie. Celui-ci, supposé inconnu, est numéroté à l'aide d'une fonction  $f$  à valeur entière, qu'il convient de déterminer. Cette paramétrisation permet de transformer le problème de la synthèse de la base de règles en un problème d'identification de la fonction  $f$ .

Finalement, la paramétrisation mise en oeuvre, tant sur le partitionnement des univers de discours que sur l'expression de la base de règles, a permis de réduire le problème de synthèse du contrôleur flou à l'identification de  $f$ ,  $\Delta a$ ,  $\Delta b$  et  $\Delta c$ .

L'application du principe des équivalences modales, illustrée par la Fig. 4, conduit à la contrainte suivante :

$$\delta u_{f(i,j)} = K_P \delta\varepsilon_j + K_I \varepsilon_i, \quad (17)$$

soit, d'après (13), (14) et (15) :

$$f(i,j) \cdot \Delta c = K_P \cdot j \cdot \Delta b + K_I \cdot i \cdot \Delta a, \quad (18)$$

ou encore :

$$f(i,j) = j K_P (\Delta b / \Delta c) + i K_I (\Delta a / \Delta c). \quad (19)$$

Rappelons que dans la règle floue (16),  $f(i,j)$  représente l'indice du symbole de sortie et à ce titre, doit donc prendre une valeur entière quel que soit le couple d'entiers  $(i,j)$ . Pour satisfaire cette contrainte supplémentaire, il faut faire en sorte que les grandeurs  $K_P (\Delta b / \Delta c)$  et  $K_I (\Delta a / \Delta c)$  soient également entières. De plus, pour que tous les symboles de sortie puissent effectivement apparaître dans une conclusion de règle, il est nécessaire de choisir :

$$K_P (\Delta b / \Delta c) = \beta \quad \text{et} \quad K_I (\Delta a / \Delta c) = \alpha, \quad (20)$$

$\alpha$  et  $\beta$  étant des entiers vérifiant  $\text{PGCD}(\alpha, \beta) = 1$ .

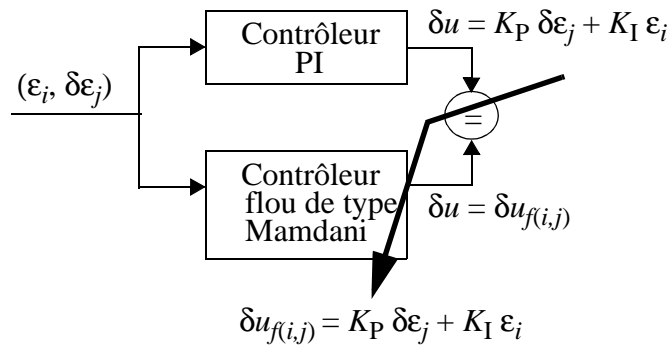


Fig. 4 : Application du principe des équivalences modales à la synthèse d'un PI.

En résumé, la synthèse du contrôleur flou requiert dans un premier temps de fixer des entiers  $\alpha$  et  $\beta$  tels que  $\text{PGCD}(\alpha, \beta) = 1$ . Ce choix étant fait, les règles sont ensuite automatiquement générées d'après (19) selon le schéma additif suivant :

$$f(i,j) = \alpha i + \beta j, \quad (22)$$

donnant ainsi lieu à des règles de la forme :

$$\text{Si } \varepsilon \text{ est } E_i \text{ et } \delta\varepsilon \text{ est } dE_j \text{ alors } \delta u \text{ est } dU_{\alpha i + \beta j}. \quad (23)$$

Enfin, la distribution des fonctions d'appartenance est réalisée de façon à vérifier les relations :

$$\frac{\Delta c}{\Delta a} = \frac{K_I}{\alpha} \quad \text{et} \quad \frac{\Delta c}{\Delta b} = \frac{K_P}{\beta}. \quad (24)$$

Le principe des équivalences modales permet de retrouver ici des résultats connus (règles additives, gains équivalents). Mais sa "force" réside avant tout dans le fait qu'il aboutit à une méthode constructive de la base de règles et à un réglage automatique du contrôleur flou associé. En cela, il s'inscrit donc dans une démarche orientée synthèse et se différencie ainsi des travaux existants, orientés analyse.

#### IV.2.2 Réglage du contrôleur synthétisé

La construction de la base de règles par application du principe des équivalences modales a permis d'établir des relations entre les paramètres du système flou et ceux du PI cible (cf équations (24)). L'implantation pratique du contrôleur flou nécessite maintenant de donner une valeur effective à chacun de ses paramètres ( $\Delta a$ ,  $\Delta b$ ,  $\Delta c$ ,  $\alpha$ ,  $\beta$ ). Si aucune contrainte n'est imposée sur la distribution des fonctions d'appartenance, en d'autres termes si le nombre final de règles n'est pas limité, un choix simple consiste à fixer  $\alpha = \beta = 1$ . On obtient alors la base de règles antidiagonale résumée dans le Tableau 1.

En associant un nom à chacun des symboles utilisés, il apparaît clairement que la base de règles ainsi générée n'est autre que celle proposée par Mac Vicar-Whelan [MacVicar76] et étudiée ultérieurement par différents auteurs ([Buckley89], [Siler89], [Ying90], [El Hajjaji95], [Mann99]). Le réglage des paramètres  $\Delta a$ ,  $\Delta b$  et  $\Delta c$  est ensuite effectué à partir des équations (24). On peut par ailleurs remarquer qu'un degré de liberté est laissé au concepteur puisque les trois paramètres ne sont soumis qu'à deux contraintes.

D'un point de vue pratique, il est souvent nécessaire de contraindre la synthèse du contrôleur flou par un nombre maximum de symboles autorisés pour décrire chacune des variables d'entrée. Les univers de discours associés étant des intervalles, les valeurs de  $\Delta a$  et  $\Delta b$  sont alors figées ou tout au moins ne peuvent être ajustées librement. Dans ce cas, la synthèse du contrôleur flou équivalent au PI cible nécessite de jouer sur les coefficients  $\alpha$  et  $\beta$  de façon à ce que l'égalité (24) puisse être satisfaite.

		Erreur $\varepsilon$								
		.....	$E_{-3}$	$E_{-2}$	$E_{-1}$	$E_0$	$E_1$	$E_2$	$E_3$	.....
Variation de l'erreur $\delta\varepsilon$	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....
	$dE_3$	.....	$dU_0$	$dU_1$	$dU_2$	$dU_3$	$dU_4$	$dU_5$	$dU_6$	.....
	$dE_2$	.....	$dU_{-1}$	$dU_0$	$dU_1$	$dU_2$	$dU_3$	$dU_4$	$dU_5$	.....
	$dE_1$	.....	$dU_{-2}$	$dU_{-1}$	$dU_0$	$dU_1$	$dU_2$	$dU_3$	$dU_4$	.....
	$dE_0$	.....	$dU_{-3}$	$dU_{-2}$	$dU_{-1}$	$dU_0$	$dU_1$	$dU_2$	$dU_3$	.....
	$dE_{-1}$	.....	$dU_{-4}$	$dU_{-3}$	$dU_{-2}$	$dU_{-1}$	$dU_0$	$dU_1$	$dU_2$	.....
	$dE_{-2}$	.....	$dU_{-5}$	$dU_{-4}$	$dU_{-3}$	$dU_{-2}$	$dU_{-1}$	$dU_0$	$dU_1$	.....
	$dE_{-3}$	.....	$dU_{-6}$	$dU_{-5}$	$dU_{-4}$	$dU_{-3}$	$dU_{-2}$	$dU_{-1}$	$dU_0$	.....
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	

Tableau 1 : Base de règles d'un PI flou.

Revenons rapidement sur la définition des entiers  $\alpha$  et  $\beta$ . A posteriori, les équations (24) permettent d'établir la relation suivante :

$$\frac{\Delta a}{\Delta b} = \frac{\alpha}{\beta} \cdot \frac{K_P}{K_I}. \quad (25)$$

Le rapport  $\alpha/\beta$  établit donc un coefficient de proportionnalité entre le rapport des gains du PI cible et  $\Delta a/\Delta b$ . Le PI cible étant supposé connu, la donnée de  $\alpha$  et  $\beta$  fixe a priori le rapport  $\Delta a/\Delta b$ . A l'opposé, la donnée d'un rapport  $\Delta a/\Delta b$  permet de déterminer les entiers  $\alpha$  et  $\beta$  de façon à satisfaire la relation (25).

### IV.2.3 Introduction de gains de normalisation

Une alternative à l'ajustement du système flou consiste à déporter le problème en introduisant des gains de normalisation sur les variables manipulées (cf Fig. 5) ([Ying90], [Ying93], [Ying94], [Li95], [Li97], [Yeh99]). L'objectif devient alors d'ajuster les gains équivalents du système global et non plus directement ceux du système flou.

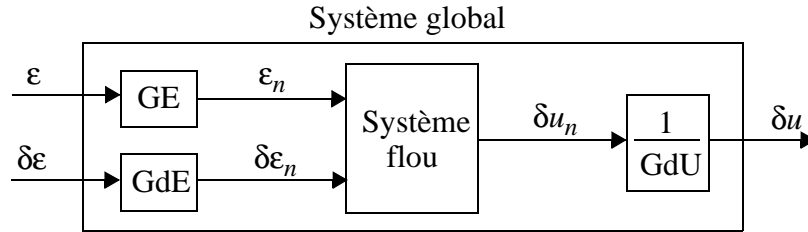


Fig. 5 : Normalisation des variables.

Soient  $\varepsilon_n$ ,  $\delta\varepsilon_n$  et  $\delta u_n$  les variables normalisées définies par :

$$\varepsilon_n = GE \cdot \varepsilon, \quad \delta\varepsilon_n = GdE \cdot \delta\varepsilon, \quad \delta u_n = GdU \cdot \delta u, \quad (26)$$

où GE, GdE et GdU représentent les gains de normalisation. Aux points modaux, le contrôleur flou engendre la commande :

$$\delta u_n = \alpha \cdot \frac{\Delta c}{\Delta a} \cdot \varepsilon_n + \beta \cdot \frac{\Delta c}{\Delta b} \cdot \delta\varepsilon_n, \quad (27)$$

et le système global produit donc la sortie :

$$\delta u = \alpha \cdot \frac{\Delta c}{\Delta a} \cdot \frac{GE}{GdU} \cdot \varepsilon + \beta \cdot \frac{\Delta c}{\Delta b} \cdot \frac{GdE}{GdU} \cdot \delta\varepsilon. \quad (28)$$

En égalisant l'équation (28) avec l'équation du PI cible, on obtient les relations suivantes, directement exploitables pour le réglage des gains de normalisation :

$$K_I = \alpha \cdot \frac{\Delta c}{\Delta a} \cdot \frac{GE}{GdU} \quad \text{et} \quad K_P = \beta \cdot \frac{\Delta c}{\Delta b} \cdot \frac{GdE}{GdU}. \quad (29)$$

L'introduction des gains de normalisation a permis d'ajouter trois degrés de liberté au système global. On peut alors complètement figer le système flou, c'est-à-dire choisir des valeurs quelconques pour  $\alpha$ ,  $\beta$ ,  $\Delta a$ ,  $\Delta b$  et  $\Delta c$ , tout en assurant une équivalence entre le système global et le PI cible. Notons que deux gains de normalisation suffisent à garantir l'existence d'un réglage. Au premier abord, l'introduction d'un troisième gain peut alors paraître inutile. Il n'en est rien puisqu'en fait, ce dernier peut être exploité pour augmenter le domaine de validité du PI flou synthétisé.

A titre d'illustration, supposons que  $\alpha=\beta=\Delta a=\Delta b=\Delta c=1$  et que le PI cible soit défini par  $K_P=1$  et  $K_I=2$ . Supposons également qu'aucun gain de normalisation ne soit utilisé sur la sortie du système flou ( $GdU=1$ ). La synthèse du système équivalent nécessite alors de régler les gains GE et GdE selon l'équation (29), soit  $GE=2$  et  $GdE=1$ . Le domaine de validité D du système

engendré dépend du nombre de règles du système flou. Ainsi pour un système à 3x3 règles, la variable  $\varepsilon$  doit appartenir à  $[-\Delta a/GE, \Delta a/GE]$  et  $\delta\varepsilon$  à  $[-\Delta b/GdE, \Delta b/GdE]$ , soit  $D=[-1/2, 1/2] \times [-1, 1]$ . L'extension de ce domaine peut être faite, soit par augmentation du nombre de règles, soit par introduction du gain GdU. Dans ce dernier cas, les gains GE et GdE peuvent en effet être multipliés par GdU sans pour autant changer les gains équivalents du système global. Ainsi, toujours pour un système à 3x3 règles, les réglages  $GE=1/5$ ,  $GdE=1/10$  et  $GdU=1/10$  permettent d'étendre D à  $[-5, 5] \times [-10, 10]$ .

En résumé, l'équation (29) donne une formulation complète des gains équivalents d'un contrôleur flou de type PI. De par son aspect générique, elle englobe les résultats obtenus par différents auteurs sur des cas particuliers, ainsi qu'illustré par le Tableau 2. La première colonne indique la référence bibliographique concernée. La seconde traduit, dans notre formalisme, les hypothèses de travail des auteurs. Rappelons que tous supposent des règles additives sans qu'aucune stratégie de construction des règles ne soit proposée dans les études considérées. Enfin, la troisième colonne fournit la loi de commande équivalente, déterminée par les auteurs, mais pouvant être également directement engendrée à partir de (29). Suivant les opérateurs d'inférence utilisés, cette dernière représente une expression analytique exacte ou un comportement limite.

Référence	Hypothèses	Loi de commande équivalente
[Siler89]	$\alpha=\beta=-1, GE=GdE=GdU=1, \Delta a=\Delta b=1, \Delta c=0.5$	$\delta u = -(\varepsilon+\delta\varepsilon)/2$
[Buckley89]	$\alpha=\beta=-1, GE=GdE=GdU=1, \Delta a=\Delta b=1/N, \Delta c=1/2N$	$\delta u \rightarrow -(\varepsilon+\delta\varepsilon)/2$ quand $N \rightarrow \infty$
[Ying90]	$\alpha=\beta=-1, \Delta a=\Delta b=2L, \Delta c=L$	$\delta u = - (GE \varepsilon + GdE \delta\varepsilon) / (2 GdU)$
[Bousslama92]	$\alpha=\beta=-1, GE=GdE=GdU=1, \Delta b=k\Delta a, \Delta c=v_0\Delta a,$	$\delta u \rightarrow -v_0(\varepsilon+\delta\varepsilon/k)$ quand $\Delta a \rightarrow 0$
[Ying93] [Ying94]	$\alpha=\beta=-1, \Delta a=\Delta b=L/J, \Delta c=H/(2J)$	$\delta u \rightarrow -H (GE \varepsilon + GdE \delta\varepsilon) / (2L GdU)$ quand $J \rightarrow \infty$
[Moon95]	$\alpha=\beta=-1, GE=GdE=GdU=1, \Delta c=M/m, \Delta a=\Delta c/K_I, \Delta b=\Delta c/K_P$	$\delta u = -K_I \varepsilon - K_P \delta\varepsilon$
[El Hajjaji95]	$\alpha=\beta=-1, GE=GdE=GdU=1, \Delta a=L/I, \Delta b=K/J, \Delta c=H/(I+J)$	$\delta u \rightarrow -(H/2L)\varepsilon - (H/2K)\delta\varepsilon$ quand $I \rightarrow \infty$ et $J \rightarrow \infty$
[Li97] [Li99]	$\alpha=\beta=1, \Delta a=\Delta b$	$\delta u \rightarrow \frac{\Delta c}{\Delta a} \left( \frac{GE \cdot \varepsilon + GdE \cdot \delta\varepsilon}{GdU} \right)$ quand nombre de règles $\rightarrow \infty$
[Mann99]	$\alpha=\beta=1, \Delta a=2/(N_1-1), \Delta b=2/(N_2-1), \Delta c=2/(N_1-N_2-2)$	$\delta u = u_{NL} + \frac{\Delta c}{\Delta a} \frac{GE}{GdU} \varepsilon + \frac{\Delta c}{\Delta b} \frac{GdE}{GdU} \delta\varepsilon$ avec $u_{NL}$ composante non linéaire

Tableau 2 : Positionnement dans notre formalisme de travaux issus de la littérature.

### IV.3 Synthèse d'un contrôleur flou de type Sugeno

La méthodologie de synthèse développée précédemment est maintenant appliquée au cas des contrôleurs flous de type Sugeno pour lesquels l'expression générique des règles devient :

$$\text{Si } \varepsilon \text{ est } E_i \text{ et } \delta\varepsilon \text{ est } dE_j \text{ alors } \delta u = g(i,j). \quad (30)$$

La fonction  $g$  est ici utilisée pour paramétrer les conclusions numériques des règles et prend donc ses valeurs dans  $R$ . Un partitionnement triangulaire régulier (cf. Fig. 3) est à nouveau imposé sur les univers de discours relatifs à  $\varepsilon$  et  $\delta\varepsilon$ . La synthèse du contrôleur flou se résume donc à déterminer  $g$ ,  $\Delta a$  et  $\Delta b$ .

Le principe des équivalences modales conduit à une contrainte unique, relative à  $g$  :

$$g(i,j) = K_P \delta\varepsilon_j + K_I \varepsilon_i = K_P \cdot j \cdot \Delta b + K_I \cdot i \cdot \Delta a. \quad (31)$$

Le choix de  $\Delta a$  et  $\Delta b$  n'est en fait sujet à aucune restriction. Entière liberté est donc laissée à l'utilisateur pour exploiter ces paramètres, notamment pour fixer le nombre de symboles désiré sur les univers de discours considérés.

Il apparaît ainsi que la synthèse d'un contrôleur de Sugeno est plus simple à mettre en oeuvre que celle d'un contrôleur de Mamdani. Cette constatation émane tout simplement du fait qu'aucune contrainte n'est imposée sur la répartition des conclusions de règles. En d'autres termes, la contrainte d'équirépartition (18) utilisée dans le cadre des systèmes de Mamdani est ici absente. La possibilité de la relaxer pour les contrôleurs de Mamdani est évoquée dans le deuxième chapitre de ce manuscrit (cf paragraphe IV.1).

### IV.4 Effet de l'interpolation entre les points modaux dans un contrôleur de Mamdani

La méthode de construction explicitée dans le paragraphe précédent permet d'assurer l'égalité de sortie du PI cible et du contrôleur flou généré, uniquement pour des entrées modales. Entre les points modaux, le contrôleur flou produit une interpolation naturelle qu'il convient maintenant d'analyser. Il est possible de déterminer analytiquement l'expression de la sortie fournie par le contrôleur flou pour un vecteur d'entrée quelconque  $(\varepsilon, \delta\varepsilon)$  appartenant à la maille floue rectangulaire définie par les entrées modales  $\varepsilon_i, \varepsilon_{i+1}$  et  $\delta\varepsilon_j$  et  $\delta\varepsilon_{j+1}$  ([El Hajjaji95], [Galichet01]). Cette approche s'avère toutefois fastidieuse dans le cas des contrôleurs de Mamdani, où l'utilisation des opérateurs min/max nécessite de considérer huit cas selon la position du vecteur d'entrée  $(\varepsilon, \delta\varepsilon)$  dans la maille concernée.

Il est ici choisi d'aborder le problème de manière pratique à partir de simulations permettant de mettre en évidence l'effet de l'interpolation entre points modaux et ce, en fonction des réglages du système flou. Le contrôleur flou est ainsi utilisé en boucle fermée de manière à réguler un système pour lequel un PI cible est déterminé au préalable. Les sorties obtenues avec le PI cible et le contrôleur flou équivalent sont alors comparées de façon à isoler l'effet des non-linéarités introduites par l'interpolation floue.

Le souhait de se focaliser sur ce dernier point nous a conduits au choix délibéré d'un procédé extrêmement simple, à savoir un premier ordre, caractérisé par une constante de temps  $\tau_{Proc}=5s$  et un gain statique de 2. La fonction de transfert correspondante est alors donnée par :

$$H_{Proc}(s) = \frac{2}{1 + 5s}. \quad (32)$$

Le PI cible est fixé de manière à obtenir en boucle fermée un premier ordre de gain unitaire et de constante de temps  $\tau_{BF}=0.25s$ . La fonction de transfert du PI cible est donc la suivante :

$$H_{PI}(s) = 10 + \frac{2}{s}. \quad (33)$$

L'implantation des régulateurs PI cible et PI flou étant réalisée en discret sur le procédé échan-

tillonné avec la période  $T_{ech}=0.1s$ , les gains du PI cible (cf équation (12)) sont finalement :

$$K_P = 10 \text{ et } K_I = 0.2.$$

La simulation est effectuée à partir du schéma Simulink illustré à la Fig. 6., le système flou étant intégré via une S-fonction. La consigne est formée de deux échelons unitaires, l'un à l'instant initial (échelon positif), l'autre à l'instant  $t = 6s$  (échelon négatif). Les univers de discours associés à l'erreur et à sa variation sont donc fixés à  $[-1, 1]$ .

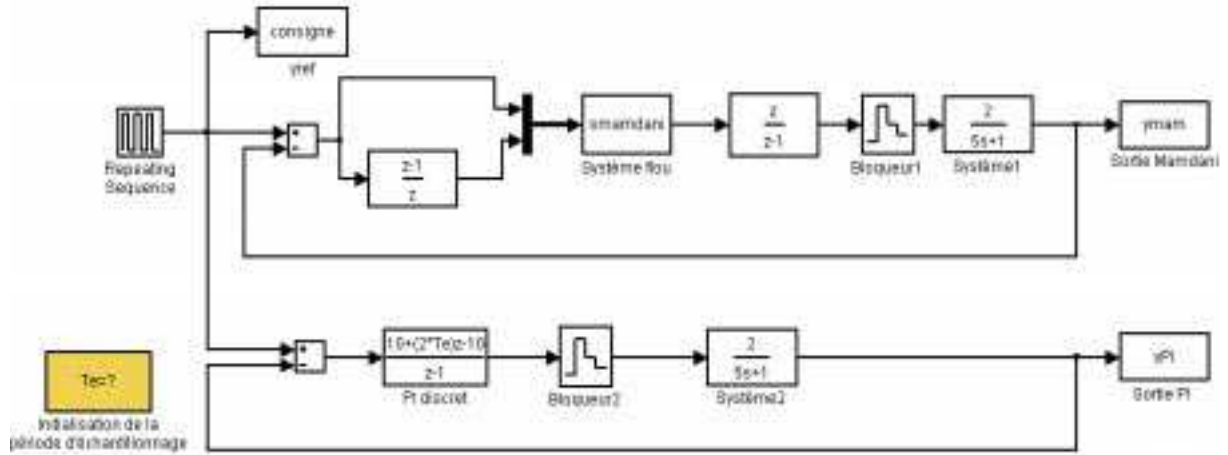
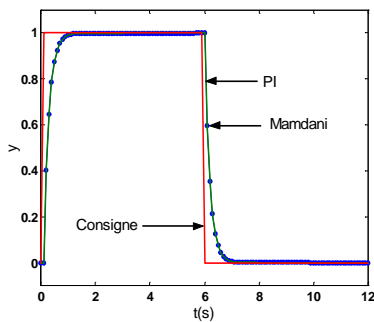


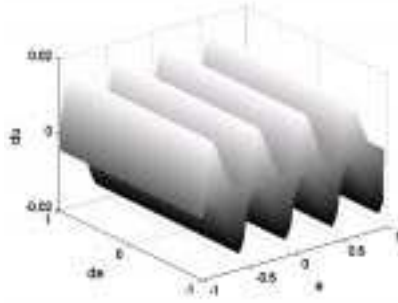
Fig. 6 : Schéma de simulation sous Simulink.

Dans un premier temps, un contrôleur flou de Mamdani est mis en oeuvre. Son réglage est d'abord effectué de manière minimale ( $\alpha=\beta=1$ , aucun gain de normalisation). Dans ce cas, selon l'équation (25), le rapport des gains du PI cible induit un rapport de 50 entre  $\Delta a$  et  $\Delta b$ . Ainsi, si l'on choisit  $\Delta a = 0.5$  (5 symboles pour décrire l'erreur),  $\Delta b$  doit être fixé à 0.01. Si PI et contrôleur flou présentent des performances similaires en régulation (cf Fig. 7(a)), il n'en demeure pas moins que ce dernier exploite une base de  $5 \times 201$  règles à laquelle aucune lisibilité ne peut être accordée. La Fig. 7(b) illustre l'erreur entre les deux surfaces de contrôle (différence entre les  $\delta u$  produits par le contrôleur de Mamdani et le PI) et met en évidence un comportement périodique selon  $\varepsilon$  correspondant aux 4 intervalles délimités par les 5 valeurs modales définies sur  $\varepsilon$ . Bien que non perceptible sur la surface d'erreur globale, un effet similaire est présent selon  $\delta\varepsilon$ . Ce dernier devient visible lorsque la représentation de la surface d'erreur est restreinte à une seule maille floue (cf Fig. 7(c)).

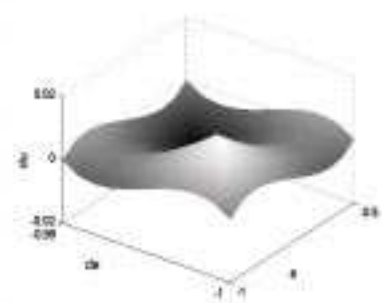
Réglages :  $\Delta a = 0.5, \Delta b = 0.01, \Delta c = 0.1, \alpha = \beta = 1, GE = GdE = GdU = 1$



(a) Sortie du système régulé



(b) Surface d'erreur



(c) Zoom sur une maille floue

Fig. 7 : Contrôleur de Mamdani (5 x 201 règles).



La seconde étape dans la détermination d'un réglage intéressant du contrôleur de Mamdani consiste à tenter une diminution du nombre de règles nécessaires et donc une augmentation de  $\Delta b$ . Différents cas sont présentés à la Fig. 8. La différence de réponse des systèmes bouclés est mise en évidence en fin de montée du premier échelon. L'utilisation de 9 symboles pour  $\delta\epsilon$  permet d'obtenir une réponse relativement satisfaisante, bien que différente de celle produite par un bouclage avec le PI cible. La montée est plus rapide mais au détriment du temps nécessaire à l'annulation de l'erreur statique. Ce phénomène est encore amplifié par une réduction à 5 du nombre de symboles concernant  $\delta\epsilon$ .

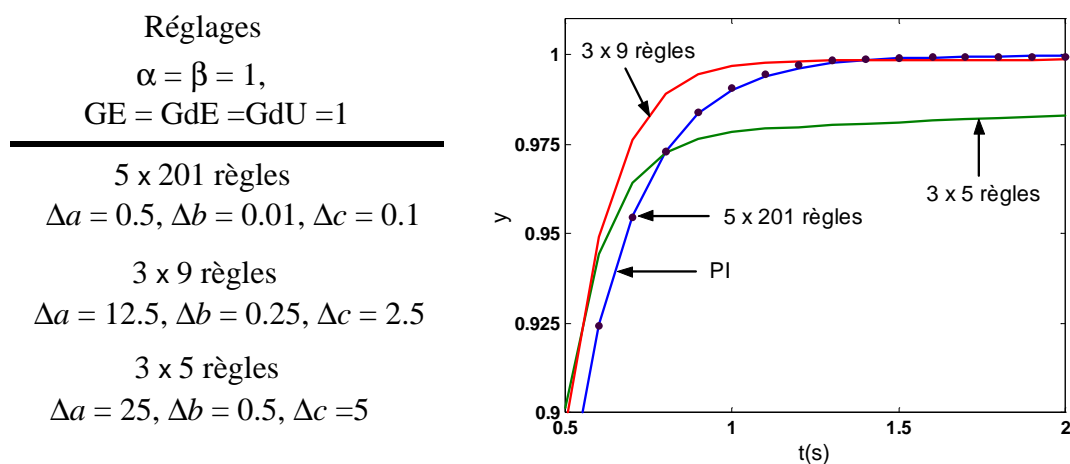


Fig. 8 : Performances en suivi selon le nombre de règles.

D'un point de vue global, il semble que la limitation du nombre de règles a entraîné une augmentation du gain proportionnel et une diminution du gain intégral du contrôleur flou. Toutefois, à l'examen des surfaces d'erreur (cf Fig. 9), il apparaît que cette constatation n'est plus valide localement. Un comportement périodique dominant est maintenant engendré selon  $\delta\epsilon$ . A nouveau, il ne peut être que subi et interdit tout réglage local. Il est à noter que les réglages de  $\Delta a$  font en sorte que l'erreur n'évolue que dans une infime partie des mailles floues définies de part et d'autre de  $\epsilon = 0$ . L'effet périodique selon  $\epsilon$  a ainsi disparu sur l'univers de discours considéré.

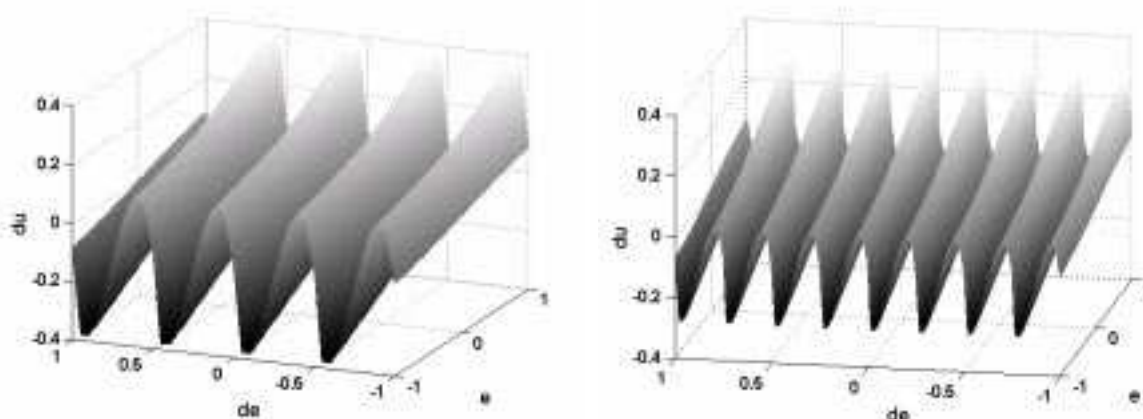


Fig. 9 : Erreur de contrôle selon le nombre de règles.

Reste enfin à souligner que même si l'erreur de suivi engendrée par un système de Mamdani de taille restreinte peut paraître satisfaisante sur un échelon unitaire, l'utilisation du contrôleur

correspondant nécessite une extrême prudence. En effet, dans la mesure où le système bouclé n'est plus tout à fait linéaire, son comportement ne peut plus être jugé par une simple réponse indicielle. La Fig. 10 illustre cet état de fait sur une échelle normalisée pour des échelons d'amplitude différente dans le cas du contrôleur à 3 x 5 règles. Les non-linéarités générées par l'interpolation floue introduisent clairement des réponses de nature différente selon la localisation de la trajectoire dans le plan de phase ( $\epsilon$ ,  $\delta\epsilon$ ).

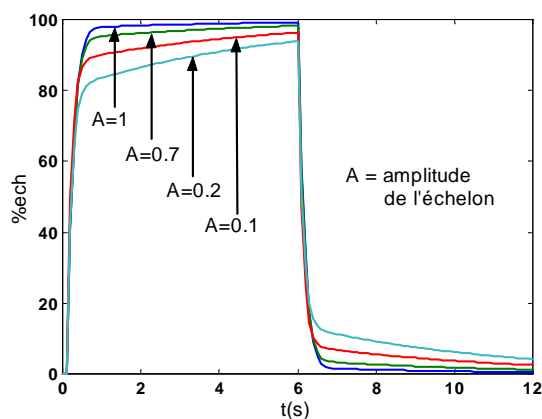


Fig. 10 : Réponse à des échelons d'amplitude différente (3 x 5 règles).

Il s'avère donc impossible d'obtenir un contrôleur de Mamdani de taille restreinte qui soit un équivalent "proche" du PI cible en exploitant simplement les réglages de base sur  $\Delta a$ ,  $\Delta b$  et  $\Delta c$ . L'étape suivante dans notre démarche consiste donc à introduire les paramètres  $\alpha$  et  $\beta$  de façon à voir s'ils permettent d'atteindre l'objectif fixé. A nouveau, les résultats obtenus sont décevants ainsi que l'illustre la Fig. 11. Les surfaces d'erreur en incrément de commande (cf Fig. 12) témoignent d'un comportement périodique à la fois sur  $\epsilon$  et  $\delta\epsilon$ .

L'introduction de  $\alpha$  et  $\beta$  a en fait permis, par une "mise à l'échelle" des conclusions de règles, de mélanger les non-linéarités engendrées sur chacune des entrées. L'exploitation des lois de commande floues paraît d'autant plus délicate que les effets non linéaires induits ne sont pas ajustables localement.

Réglages	
GE = GdE = GdU = 1	
5 x 21 règles	
$\Delta a = 0.5$ , $\Delta b = 0.1$ , $\Delta c = 0.1$	$\alpha = 1$ , $\beta = 10$
5 x 9 règles	
$\Delta a = 0.5$ , $\Delta b = 0.25$ , $\Delta c = 0.1$	$\alpha = 1$ , $\beta = 25$
5 x 5 règles	
$\Delta a = 0.5$ , $\Delta b = 0.5$ , $\Delta c = 0.1$	$\alpha = 1$ , $\beta = 50$

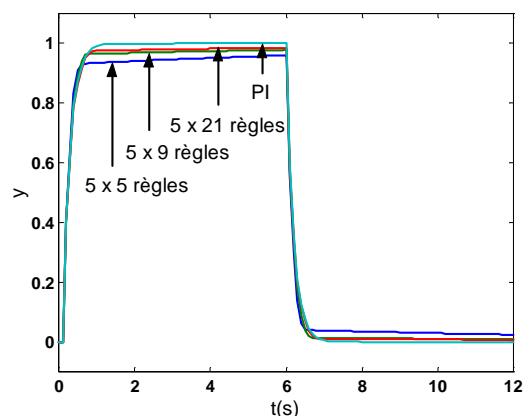


Fig. 11 : Erreur de suivi pour différents réglages de  $\alpha$  et  $\beta$ .

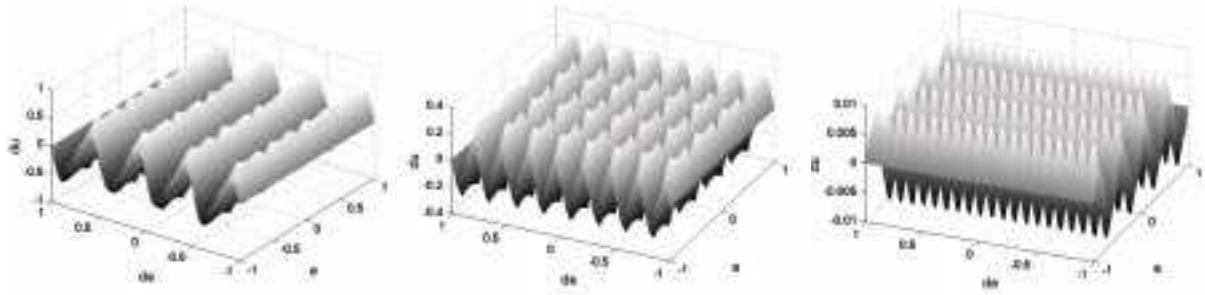


Fig. 12 : Surfaces d'erreur en incrément de commande pour différents réglages de  $\alpha$  et  $\beta$ .

La dernière tentative consiste à prendre en compte les paramètres de réglage supplémentaires que sont les gains GE, GdE et GdU. Il s'avère en fait qu'une utilisation conjointe des écarts entre valeurs modales et des gains de normalisation ne modifie en rien les résultats obtenus précédemment. Ainsi, pour des valeurs de  $\alpha$  et  $\beta$  données, seul le réglage des rapports  $\frac{GE}{\Delta a}$ ,  $\frac{GdE}{\Delta b}$  et  $\frac{GdU}{\Delta c}$  influe sur la surface de commande obtenue. Cette constatation mise en évidence aux valeurs modales dans l'équation (28) reste valide pour des entrées quelconques. Ainsi, le choix d'une mise à l'échelle des variables d'entrée et de sortie, soit de façon externe via les gains de normalisation, soit de façon interne via les partitions floues, n'a aucune incidence sur la surface de commande engendrée. Seul un ajustement via les règles par l'introduction des  $\alpha$  et  $\beta$  modifie la nature de l'interpolation entre les valeurs modales.

Les simulations mises en oeuvre permettent de tirer différentes conclusions résumées ci-dessous.

- L'obtention d'un équivalent linéaire satisfaisant nécessite un nombre de règles élevé.
- Les non-linéarités introduites par l'interpolation floue ne sont pas contrôlables. De plus, leur périodicité induit des effets indésirables.
- Seule l'introduction des paramètres  $\alpha$  et  $\beta$  permet de fixer le nombre de règles et les univers de discours effectivement exploités tout en garantissant les gains équivalents du contrôleur flou.
- L'usage conjoint des paramètres de répartition des fonction d'appartenance et des gains de normalisation n'apportent rien.

#### IV.5 Obtention d'un équivalent linéaire exact

Dans la mesure où les non-linéarités introduites par l'interpolation floue ne sont pas utilisables, l'objectif est maintenant de les éliminer de façon à obtenir une implantation floue exacte du PI cible.

Dans le cas des systèmes de Sugeno, il s'avère que le contrôleur flou synthétisé selon l'équation (31) est linéaire ([Mizumoto93], [Galichet95], [Sugeno99]). A titre d'illustration, on peut ainsi vérifier que le système de Sugeno caractérisé par la base de règles de la Fig. 13 est équivalent au PI cible. D'un point de vue expérimental, une différence de sortie de l'ordre de  $10^{-15}$  est constatée entre les deux systèmes régulés. Cette erreur négligeable, imputable au bruit de calcul numérique, valide en simulation le résultat théorique d'une équivalence exacte entre les deux contrôleurs implantés. Remarquons que le nombre de symboles définis sur les partitions d'entrée peut être choisi de manière quelconque et n'influe aucunement sur la surface de contrôle obtenue. En fait, il suffit de deux symboles sur chaque univers de discours (quatre points

modaux dans le plan de phase) pour construire le système flou.

Dans le cas des systèmes de Mamdani, seule l'utilisation des opérateurs min/max induit les non-linéarités de la loi de commande. En effet, il est possible de montrer que le choix du produit comme t-norme et de la somme bornée à 1 comme t-conorme permet d'obtenir une interpolation linéaire entre les points modaux ([Galichet95]). Ce résultat généralise à une défuzzification par centre de gravité celui préalablement établi dans [Mizumoto93] avec la méthode des hauteurs. Ainsi, sous réserve d'exploiter des opérateurs "linéaires", le contrôleur de Mamdani défini par  $\Delta a = \Delta b = 1$ ,  $\Delta c = 0.2$ ,  $\alpha = 1$  et  $\beta = 50$  et composé de  $3 \times 3$  règles (cf Fig. 14) est strictement équivalent au PI cible sur l'univers de discours  $[-1, 1] \times [-1, 1]$ . Si cet état de fait est confirmé par la simulation, il n'en demeure pas moins que l'implantation numérique de la défuzzification par centre de gravité engendre une erreur d'approximation. D'un point de vue pratique, l'obtention d'une erreur satisfaisante requiert l'utilisation d'un pas de discrétisation faible. Ainsi, pour un écart de l'ordre de  $10^{-7}$  sur la sortie du système régulé, il est nécessaire de fixer un pas de discrétisation déjà très faible sur  $\delta u$  ( $10^{-3}$ ).

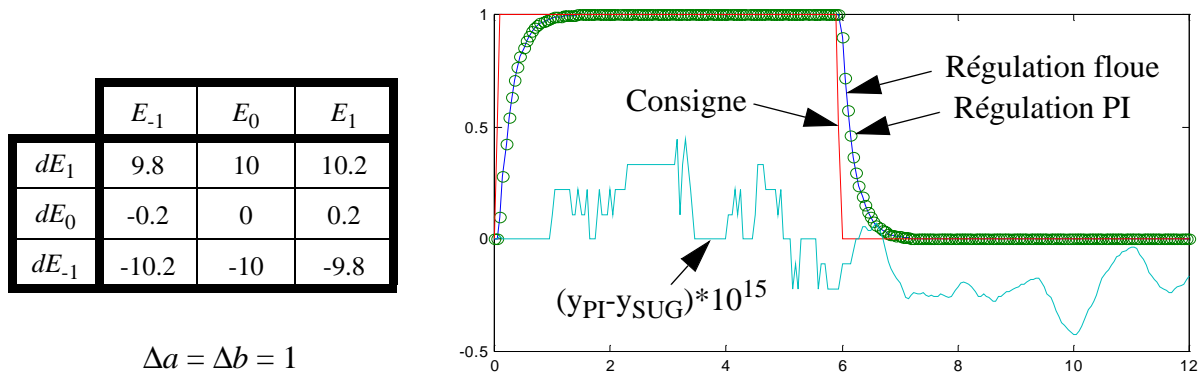


Fig. 13 : Contrôleur flou de Sugeno (base de règles et réponse indicielle).

L'objectif initial d'obtention d'un contrôleur flou implantant une loi de commande linéaire est maintenant atteint à la fois pour des systèmes de Mamdani et de Sugeno. De plus, les paramètres de réglage associés permettent de dimensionner les systèmes flous synthétisés en termes de nombre de règles. Reste toutefois à souligner que dans le cas des systèmes de Mamdani, la base de règles générée (cf Fig. 14) n'exploite qu'une très faible proportion des symboles définis sur l'univers de discours associé à la sortie et s'avère donc difficilement justifiable d'un point de vue lisibilité et interprétation. Dans toute la suite de ce chapitre, la synthèse de contrôleurs de Mamdani est donc abandonnée au profit des systèmes de Sugeno, plus facilement réglables. Cette mise à l'écart ne sera cependant que provisoire puisque le chapitre suivant mettra en évidence la possibilité de transformation d'une base de règles de Sugeno en son homologue de Mamdani.

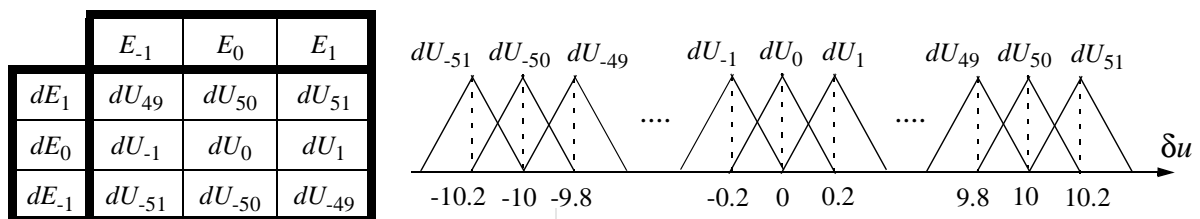


Fig. 14 : Contrôleur de Mamdani équivalent au PI cible.

## V Extension à la synthèse de contrôleurs flous linéaires par retour d'état

Les travaux développés ci-dessus ont été restreints à la construction d'un contrôleur flou de type PI. Ils peuvent bien évidemment être adaptés pour générer un contrôleur flou de type PD, auquel cas la variable de sortie du contrôleur flou devient la commande elle-même et non plus sa variation.

De manière plus générale, la méthodologie utilisée s'applique sans difficulté à la synthèse d'un système flou implantant une relation entrées/sortie linéaire du type  $s = K_1 e_1 + K_2 e_2$ , où  $e_1$  et  $e_2$  représentent les entrées du système et  $s$  la sortie. Les variables d'entrée du système sont alors utilisées dans les prémisses de règles alors que la variable de sortie apparaît dans les conclusions. Dans ce cas, le système flou implanté peut être vu comme un contrôleur, mais aussi comme un modèle ou, de façon plus large, comme une fonction nommée  $SF$  permettant de déterminer la sortie  $s$  à partir des entrées  $e_1$  et  $e_2$ , c'est-à-dire :  $s = SF(e_1, e_2)$ . La mise en oeuvre du principe des équivalences modales permet de garantir que la fonction  $SF$  produit la même sortie que la fonction linéaire cible pour des entrées modales. L'interpolation floue produite par un système de Sugeno étant linéaire, la représentation obtenue est exacte.

Le principe des équivalences modales peut également être appliqué à la construction d'un système flou permettant d'approximer une fonction linéaire de  $n$  variables. La problématique est alors la suivante :

*Soit la fonction cible  $f_L$  définie par  $s = f_L(e_1, e_2, \dots, e_n) = K_1 e_1 + K_2 e_2 + \dots + K_n e_n$ . Comment déterminer le système flou  $SF$  permettant d'approcher  $f_L$  ?*

Le système flou recherché prend en compte les mêmes variables que le système cible linéaire et est donc construit à partir de règles du type :

Si  $e_1$  est  $A_1^{i_1}$  et  $e_2$  est  $A_2^{i_2}$  et ..... et  $e_n$  est  $A_n^{i_n}$  alors  $s = g(i_1, i_2, \dots, i_n)$ , (34)  
pour un système de Sugeno.

Comme précédemment les partitions des univers de discours sont supposées triangulaires avec une répartition régulière des fonctions d'appartenance associées aux symboles. Si l'on note  $\Delta a_k$ ,  $k=1, \dots, n$ , la distance entre deux valeurs modales consécutives relatives à la variable  $e_k$ , l'application du principe des équivalences modales aboutit à la relation suivante :

$$g(i_1, i_2, \dots, i_n) = \sum_{k=1}^n K_k \cdot e_k^{i_k} = \sum_{k=1}^n K_k \cdot i_k \cdot \Delta a_k, \quad (35)$$

où  $e_k^{i_k}$  représente la  $i_k$  ième valeur modale de la variable  $e_k$ . Il est démontré dans [Ying97] que la forme de règles engendrée (cf équations (34) et (35)) est une condition nécessaire et suffisante à l'obtention d'un système flou globalement linéaire.

Ce résultat s'applique de manière immédiate à la synthèse d'un contrôleur flou de type retour d'état ([Galichet94b], [Foulloy98]). En effet, soit le système linéaire multi-entrées, multi-sorties défini par la représentation d'état suivante :

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad \text{avec} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} \quad u = \begin{bmatrix} u_1 \\ u_2 \\ \dots \\ u_m \end{bmatrix} \quad \text{et} \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_p \end{bmatrix}, \quad (36)$$

où  $x$  représente l'état du système,  $u$  son entrée et  $y$  sa sortie. Le principe de la compensation par retour d'état (cf Fig. 15) consiste à définir la loi de commande  $u$  sous la forme linéaire suivante :

$$u = Lr - Kx, \quad (37)$$

où  $r \in \mathfrak{R}^p$  est la nouvelle entrée du système,  $K \in M^{m,n}$  la matrice de retour d'état et  $L \in M^{m,p}$  la matrice permettant de fixer le comportement statique du système compensé.

L'implantation floue d'une telle commande nécessite l'élaboration d'un système flou à  $p+n$  entrées et  $m$  sorties. Dans la mesure où les entrées relatives à la nouvelle entrée  $r$  et celles relatives à l'état  $x$  peuvent être clairement dissociées, l'implantation floue est en fait réalisée avec deux systèmes flous distincts. Le premier, en charge du retour d'état, se substitue à la matrice  $K$  dans la Fig. 15, alors que le second remplace la matrice  $L$ . Cette façon de faire permet de "coller" au plus près de la structure de commande d'origine mais aussi de réduire le nombre de règles global. Dans la suite de ce paragraphe, le problème est donc restreint à la synthèse de la loi  $v = Kx$ , une méthodologie strictement similaire pouvant être appliquée au transfert  $Lr$ .

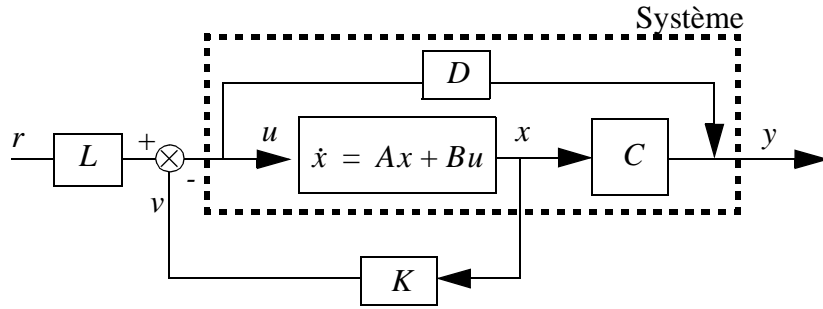


Fig. 15 : Principe de la commande par retour d'état.

Dans le cas où le système à commander ne possède qu'une seule entrée ( $m=1$ ), le résultat obtenu précédemment (équation (35)) est directement applicable en remplaçant les entrées génériques  $e_1, e_2, \dots, e_n$  exploitées dans les prémisses de règles (34) par les composantes du vecteur d'état  $x$ . Le système flou produit alors la sortie escomptée, à savoir  $v = Kx$ , dans la mesure où la contrainte (35) est satisfaite.

Quant au cas multi-variable, il nécessite la synthèse d'un système flou à  $m$  sorties basé sur des règles du type :

$$\begin{aligned} \text{Si } x_1 \text{ est } A_1^{i_1} \text{ et } x_2 \text{ est } A_2^{i_2} \text{ et } \dots \text{ et } x_n \text{ est } A_n^{i_n} \text{ alors} \\ v_1 = g_1(i_1, i_2, \dots, i_n) \text{ et } v_2 = g_2(i_1, i_2, \dots, i_n) \text{ et } \dots \text{ et } v_m = g_m(i_1, i_2, \dots, i_n). \end{aligned} \quad (38)$$

Un tel système peut en fait être vu comme une collection de systèmes flous à sortie unique. Chacun d'entre eux repose alors sur une même structure de règles dont seul le paramètre de sortie diffère. Chaque sous-système est ensuite réglé selon l'équation (35), ce qui génère l'ensemble de contraintes suivant :

$$\forall l \in \{1, 2, \dots, m\} \quad g_l(i_1, i_2, \dots, i_n) = \sum_{k=1}^n K_{l,k} \cdot x_k^{i_k} = \sum_{k=1}^n K_{l,k} \cdot i_k \cdot \Delta a_k. \quad (39)$$

Il apparaît clairement qu'imposer une même partition des entrées pour chaque sous-système flou, i.e.  $\Delta a_k$  indépendant de  $l$ , ne pose aucune difficulté pour la détermination des conclusions de règles. Cette constatation n'est cependant plus valide pour des systèmes de Mamdani pour lesquels la synthèse des règles selon l'approche développée précédemment devient extrêmement délicate, voire impossible dans la mesure où le nombre de paramètres de réglage devient inférieur au nombre de contraintes à satisfaire ([Galichet94b], [Foulloy98]).

## VI Application à la synthèse de contrôleurs flous non linéaires

L'idée développée dans ce paragraphe est tout simplement d'exploiter le principe des équivalences modales pour la synthèse d'un contrôleur non linéaire. La présentation s'articule autour de deux applications distinctes qui se différencient essentiellement de par leur objectif. En effet, si la première concerne l'intégration d'une expertise humaine dans une loi de commande analytique, la seconde est avant tout un problème d'optimisation d'une fonction coût exprimée numériquement. Le contexte de ces deux applications est également très différent. Dans un cas, il s'agit d'apporter une réponse à un problème industriel. Dans l'autre, le travail est réalisé en simulation et doit être considéré comme une pré-étude visant à tester la faisabilité d'une approximation floue. Malgré ces divergences de nature, les deux applications présentées ont en point commun le fait que la fonction cible est non linéaire et de forme analytique connue.

Ces deux études s'inscrivent donc dans la problématique plus générale de l'approximation de fonctions. Dans les années 90, la bibliographie traitant du sujet a essentiellement abordé l'aspect "théorique" de l'approximation. Ainsi, de nombreux auteurs se sont attachés à démontrer que telle ou telle famille de systèmes flous possède la propriété d'approximateurs universels ([Buckley92], [Wang92a], [Buckley93]). Basés sur le théorème de Stone-Weierstrass, ces travaux n'apportent aucune aide pratique quant à la synthèse d'un système flou permettant d'approximer une fonction cible. Des publications plus récentes tentent cependant de donner des éléments de réponse sous la forme de contraintes que doit satisfaire le système flou. Dans [Zeng96b], il est par exemple montré qu'une approximation plus précise est généralement obtenue si les partitions des univers d'entrée sont telles que les fonctions d'appartenance se croisent au point milieu de l'intersection de leur support. [Kosko95] préconise qu'une règle floue soit associée à chaque extremum de la fonction à approximer. Dans le même courant de pensée, [Ying97] et [Ding00] mettent en évidence la nécessité qu'un point modal soit effectivement défini en tout extremum de la fonction cible. Bien que ces résultats soient constructifs vis-à-vis de l'élaboration du système flou, ils ne constituent pas encore une méthodologie de synthèse complète, telle que celle proposée par [Lygeros97] dans le cas particulier des fonctions polynomiales à coefficients rationnels.

Les travaux présentés ci-dessous n'ont pas la prétention de pallier ce manque. Ils abordent uniquement l'aspect pratique de l'approximation de fonctions et tentent simplement d'apporter une solution à des problèmes concrets.

### VI.1 Régulation de niveau en raffinerie

Cette application concerne la mise en oeuvre d'un contrôleur flou pour la régulation de niveau flottant dans une capacité tampon en raffinerie. Cette étude ([Galichet94a]), menée en collaboration avec le centre de recherche de ELF Solaize, exploite le principe des équivalences modales pour synthétiser le contrôleur flou à partir d'un contrôleur non linéaire existant. Celui-ci donne satisfaction en mode de fonctionnement normal mais est incapable de traiter d'éventuelles perturbations. Le contrôleur flou est donc modifié localement pour prendre en compte le contexte de fonctionnement du système.

La solution proposée s'inscrit donc dans une stratégie en deux étapes :

- copie de l'existant pour construire le contrôleur flou,
- amélioration du contrôleur flou initial par introduction d'une connaissance experte.

Il est à noter que cette démarche ne peut être envisagée que si l'on dispose effectivement d'une connaissance experte sur la façon d'améliorer la régulation. Dans ce cas, la traduction de la loi

de commande non linéaire en base de règles floues permet d'exploiter dans un même contexte, des connaissances de nature différente : forme analytique de la loi de commande et expertise sur la conduite du procédé. L'aspect "intégration de l'expertise", deuxième phase dans le processus de construction du contrôleur flou, n'est pas abordé ici mais sera détaillé dans le second chapitre de ce mémoire.

Le procédé considéré est illustré à la Fig. 16. L'objectif de la régulation est de maintenir le niveau courant  $L$  entre les bornes  $L_{min}$  et  $L_{max}$ . Si aucune consigne précise n'est imposée sur le niveau, il est important de minimiser les variations du débit de sortie, seule variable de commande autorisée. Dans ce contexte, le contrôleur existant, dit contrôleur algorithmique, évalue la variation de débit de sortie ( $\delta F$ ) de façon à stabiliser le niveau avant qu'il n'atteigne une borne limite.

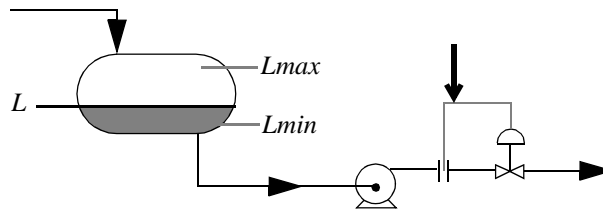


Fig. 16 : Régulation de niveau flottant.

Les équations suivantes avaient ainsi été déterminées par les ingénieurs :

$$\begin{aligned} \delta F &= K \cdot \frac{(\delta L)^2}{L_{max} - L} & \text{si} & \quad \delta L \geq 0, \\ \delta F &= K \cdot \frac{(\delta L)^2}{L_{min} - L} & \text{si} & \quad \delta L \leq 0, \end{aligned} \quad (40)$$

où  $\delta L$  représente la variation de niveau et  $K$  une constante caractéristique de la cuve et de la période d'échantillonnage. Un seuillage est ensuite appliqué à la valeur calculée de façon à garantir :

$$|\delta F| \leq G. \quad (41)$$

Le choix des variables d'entrée du contrôleur flou est guidé par l'équation (40). Si une utilisation directe de la variable  $L$  semble appropriée, il n'en est pas de même pour la variable  $\delta L$ . En effet,  $\delta L^2$  étant l'unique occurrence de la variation de niveau dans (40), l'élévation au carré peut être traitée à l'extérieur du système flou. Cette approche requiert toutefois de préserver l'information sur le signe de  $\delta L$  et conduit donc au choix de la variable  $\text{signe}(\delta L) * \delta L^2$  comme seconde entrée du système flou.

La base de règles construite par application du principe des équivalences modales est donnée au Tableau 3 pour  $\Delta a = 25$ ,  $\Delta b = 100/3$ ,  $L_{min} = 0$ ,  $L_{max} = 100$  et  $K = 1$ . Les symboles  $A_0$  et  $A_4$  ont pour valeurs modales respectives 0 et 100, c'est-à-dire  $L_{min}$  et  $L_{max}$ , et conduisent donc suivant le signe de  $\delta L$  à des conclusions de règles qui tendent vers plus ou moins l'infini. L'introduction de l'étape de saturation (41) dans le contrôleur flou permet de remplacer la notation  $\infty$  par la valeur  $G$  dans le Tableau 3. Les cases  $(A_0, B_0)$  et  $(A_4, B_0)$  mettent en évidence une discontinuité dans l'expression de la commande  $\delta F$ . Si l'on désire représenter au mieux cette dernière, il est nécessaire de couper le symbole  $B_0$  en  $B_0^-$  et  $B_0^+$ . Dans notre cas, la base de règles sera retravaillée avec un expert procédé de façon à améliorer le comportement global de la commande.



		→ signe( $\delta L$ )* $\delta L^2$						
		$B_{-3}$	$B_{-2}$	$B_{-1}$	$B_0$	$B_1$	$B_2$	$B_3$
$Lmin$ $\downarrow$ $L$ $\downarrow$ $Lmax$	$A_0$	$-\infty$	$-\infty$	$-\infty$	$-\infty/0$	$1/3$	$2/3$	$1$
	$A_1$	$-4$	$-8/3$	$-4/3$	$0$	$4/9$	$8/9$	$4/3$
	$A_2$	$-2$	$-4/3$	$-2/3$	$0$	$2/3$	$4/3$	$2$
	$A_3$	$-4/3$	$-8/9$	$-4/9$	$0$	$4/3$	$8/3$	$4$
	$A_4$	$-1$	$-2/3$	$-1/3$	$0/+\infty$	$+\infty$	$+\infty$	$+\infty$

Tableau 3 : Base de règles synthétisée.

La Fig. 17 donne à titre illustratif les surfaces de commande associées au contrôleur algorithmique et au contrôleur flou (choix d'une commande nulle dans les cas indéterminés et  $G=10$ ). La grille sur laquelle sont répartis les points modaux est visualisée sur la surface floue. Notons que le choix de la variable d'entrée  $\text{signe}(\delta L) * \delta L^2$  a induit une distribution quadratique des valeurs modales associées à  $\delta L$ .

Bien qu'un même comportement global soit obtenu, il est clair que les deux surfaces de commande restent significativement différentes, notamment lorsque le niveau s'approche des bornes fixées. Le contrôleur algorithmique ne donnant pas satisfaction en de telles circonstances, aucun effort n'est fait pour le reproduire plus fidèlement. A ce stade de la construction du contrôleur flou, la qualité de copie obtenue est donc jugée suffisante pour pouvoir procéder à la phase ultérieure d'amélioration de la commande par intégration de l'expertise ([Galichet94a]). Cette étape nécessitant une manipulation des règles à un niveau purement linguistique ne sera introduite que dans le second chapitre de ce document.

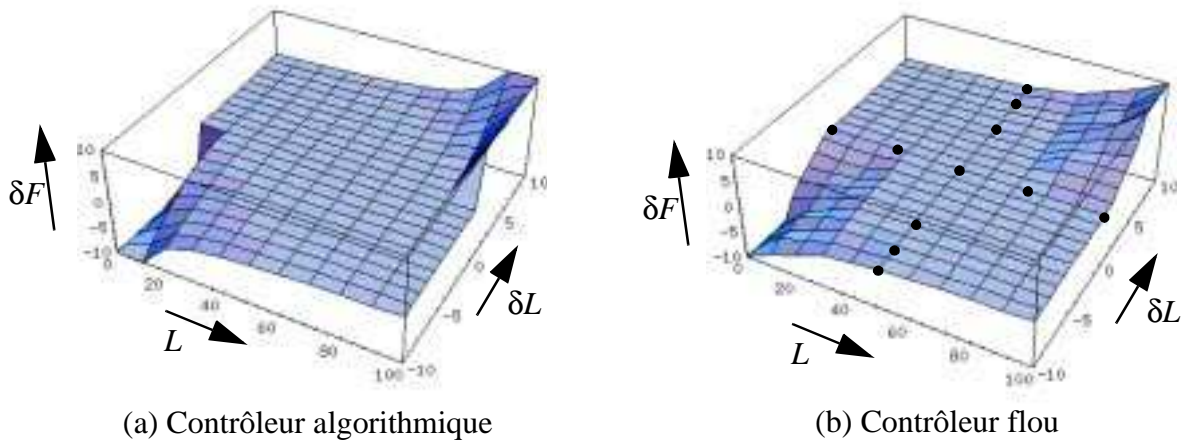


Fig. 17 : Surfaces de commande.

## VI.2 Transformation linéarisante

L'approche développée a pour but d'implanter à l'aide d'un système flou une transformation linéarisante d'un système non linéaire mono-entrée, mono-sortie. Ce dernier, par hypothèse linéarisable, est ici supposé connu et exprimé dans une représentation d'état sous la forme :

$$\begin{cases} \dot{x} = f(x) + g(x) \cdot u, \\ y = h(x), \end{cases} \quad (42)$$

où  $x = [x_1, x_2, \dots, x_n]^T$  est le vecteur d'état du système,  $u$  son entrée et  $y$  sa sortie. Pour alléger les notations, les développements sont effectués pour un système de degré relatif  $r$  égal à  $n$ . La

démarche peut cependant être étendue à tout système dont la dynamique de zéros est stable. Dans le cas où  $r = n$ , la relation entrée-sortie associée au système (42) est donnée par :

$$y^{(n)} = L_f^{(n)}h(x) + L_gL_f^{(n-1)}h(x)u, \quad (43)$$

où la notation  $L_a b(x)$  représente une dérivée de Lie et signifie la dérivée de  $b$  dans la direction de  $a$ . Selon une approche de linéarisation entrée-sortie, une nouvelle entrée  $v$  est introduite et la commande :

$$u = -\frac{L_f^{(n)}h(x)}{L_gL_f^{(n-1)}h(x)} + \frac{1}{L_gL_f^{(n-1)}h(x)} \cdot v = \alpha(x) + \beta(x) \cdot v, \quad (44)$$

est appliquée au système. On obtient ainsi le système linéaire  $y^{(n)} = v$ , dont la représentation d'état s'écrit :

$$\begin{cases} \dot{z} = Az + Bv, \\ y = Cz, \end{cases} \quad (45)$$

avec

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \\ 1 \end{bmatrix}, C^T = \begin{bmatrix} 1 \\ 0 \\ \dots \\ 0 \\ 0 \end{bmatrix}.$$

Le nouveau vecteur d'état  $z$  est obtenu par transformation de coordonnées à l'aide du difféomorphisme  $\Phi$  défini par :

$$z = \Phi(x) = \begin{bmatrix} h(x) \\ L_f h(x) \\ \dots \\ L_f^{(n-1)}h(x) \end{bmatrix}. \quad (46)$$

Il convient maintenant de stabiliser le système linéarisé (chaîne d'intégrateurs asymptotiquement instable). Un simple retour d'état, c'est-à-dire  $v = -Kz$ , permet d'atteindre cet objectif dans la mesure où  $K$  est choisi de façon à ce que la matrice  $A-BK$  ait toutes ses valeurs propres à partie réelle négative. En résumé, la commande  $u = \alpha(x) - \beta(x) K \Phi(x)$  permet de linéariser et de stabiliser le système considéré. C'est cette dernière qui devient la fonction cible que l'on cherche à approximer par un système flou qui sera donc basé sur une collection de règles de la forme :

$$\text{Si } x_1 \text{ est } A_1^{i_1} \text{ et } x_2 \text{ est } A_2^{i_2} \text{ et } \dots \text{ et } x_n \text{ est } A_n^{i_n} \text{ alors } u = g(i_1, i_2, \dots, i_n), \quad (47)$$

avec  $g(i_1, i_2, \dots, i_n) = \alpha(x^{(i_1, i_2, \dots, i_n)}) - \beta(x^{(i_1, i_2, \dots, i_n)}) K \Phi(x^{(i_1, i_2, \dots, i_n)})$ ,  $x^{(i_1, i_2, \dots, i_n)}$  étant le vecteur d'entrée modal associé à la règle concernée, c'est-à-dire le vecteur composé des valeurs modales associées aux symboles  $A_1^{i_1}, A_2^{i_2}, \dots, A_n^{i_n}$ . Pour un partitionnement donné des univers de discours d'entrée, la synthèse de la base de règles est donc immédiate dans la mesure où  $\alpha, \beta, \Phi$  et  $K$  sont supposés connus.

Pour une mise en oeuvre pratique, il est maintenant nécessaire de se poser les questions suivantes :

*Combien faut-il définir de symboles pour décrire chaque composante du vecteur d'état ?*

*Comment faut-il répartir ces derniers sur les différents univers de discours ?*

Le problème est ici restreint au choix du nombre de symboles, ceux-ci étant distribués uniformément. Cette façon de faire est bien évidemment critiquable, mais elle permet de se faire une opinion rapide sur la faisabilité de l'approximation floue. En effet, si pour une répartition régu-

lière des symboles, on peut atteindre une approximation floue correcte avec un nombre limité de symboles, un partitionnement optimal des univers de discours ne pourra qu'améliorer la qualité du résultat final. Cet état de fait est illustré dans [Pedrycz97] sur différents exemples d'approximation de fonctions non linéaires. Une optimisation des partitions y est proposée selon un critère combinant partage équitable de la variabilité de la fonction cible entre règles et distinguabilité<sup>(1)</sup> des régions floues considérées.

Le choix du nombre de symboles est ici abordé de manière pragmatique selon une approche incrémentale. Ainsi, la base de règles, initialisée avec un nombre de symboles minimal (deux pour chaque variable), est élargie de manière itérative. A chaque étape, le système flou est testé en boucle fermée sur le procédé (Fig. 18(b)). Le signal de sortie généré est alors comparé à celui obtenu avec la loi de commande cible (Fig. 18(a)). Si le résultat est jugé insuffisant, le nombre de symboles est augmenté, mais uniquement pour la composante du vecteur d'état qui engendre l'erreur maximale sur l'intervalle de temps considéré. Dans le cas contraire, la synthèse est terminée.

La Fig. 19 illustre les résultats obtenus sur le système suivant :

$$\begin{cases} \dot{x}_1 = \sin(x_2) \\ \dot{x}_2 = x_1^4 \cdot \cos(x_2) \\ y = x_1 \end{cases} \quad (48)$$

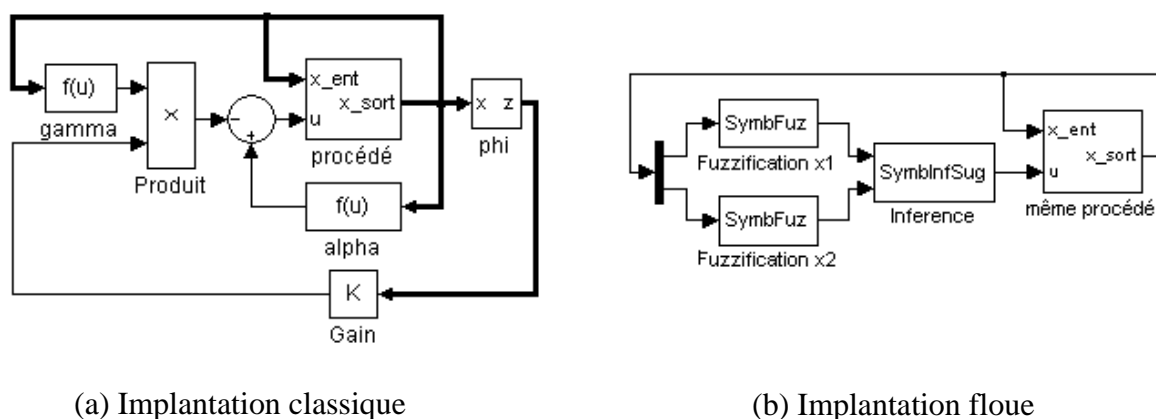


Fig. 18 : Linéarisation entrée/sortie et stabilisation.

La Fig. 19(a) présente la surface de commande obtenue par linéarisation entrée/sortie et placement de pôles, c'est-à-dire  $u = \alpha(x) - \beta(x) K \Phi(x)$  avec  $K = [2, 3]$ , pour  $x_1 \in [0, \pi/4]$  et  $x_2 \in [-\pi/8, 3\pi/8]$ . Son approximation floue est présentée à la Fig. 19(b). Elle correspond à une base de 21 règles (3 symboles sur  $x_1$ , 7 sur  $x_2$ ). Ce choix, établi après 6 itérations selon la méthodologie décrite ci-dessus, s'avère satisfaisant ainsi qu'en attestent les réponses temporelles de la Fig. 19(c) pour des conditions initiales fixées à 0.5 sur  $x_1$  et 1.1 sur  $x_2$ . Le contrôleur flou permet de rallier le point d'équilibre (0, 0) en suivant des trajectoires très proches de celles obtenues en utilisant la loi de commande cible.

(1) Par ce néologisme, nous reprenons ici le terme "distinguishability" largement utilisé dans la littérature de langue anglaise ([deOliveira95], [Pedrycz97]).

Les résultats de cette pré-étude nous ont encouragés à poursuivre plus en avant dans la voie tracée, à savoir l'exploitation de contrôleurs de Sugeno à conclusions constantes pour approximer des lois non linéaires. Les paragraphes qui suivent abordent à nouveau la synthèse du système flou mais cette fois, lorsque la fonction non linéaire cible est inconnue. On a alors recours à des techniques d'apprentissage ou d'adaptation suivant la nature de l'information disponible.

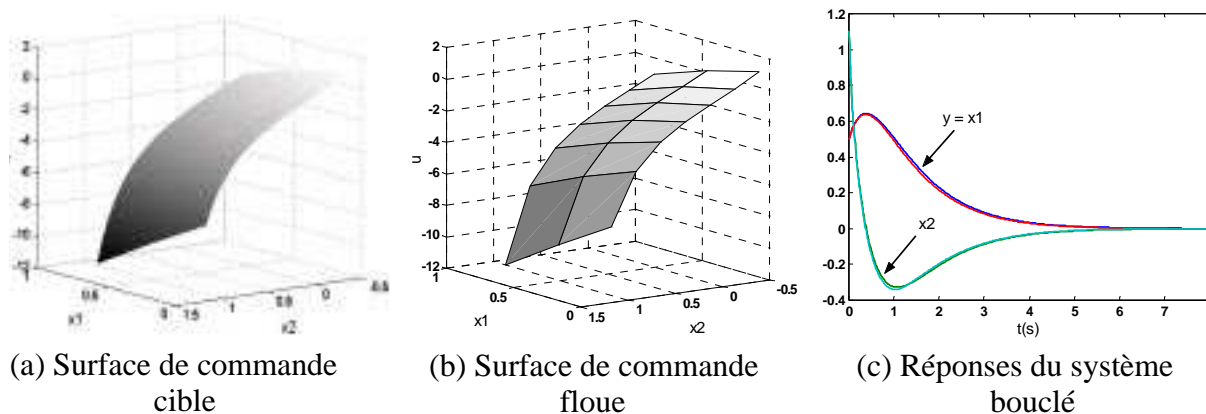


Fig. 19 : Surfaces de commande et réponse du système bouclé.

## VII Apprentissage d'une base de règles à partir d'exemples

### VII.1 Préambule

Les travaux présentés ici ont été réalisés dans le cadre de la thèse de Yassar Nakoula ([Nakoula97a], [Nakoula97b]). Ils sont relatifs à l'apprentissage d'une base de règles à partir d'exemples et s'inscrivent donc dans la lignée de [Pedrycz84], [Takagi85] ou [Wang92b]. Ces dernières années, la littérature sur le sujet a été particulièrement abondante. Deux courants de pensée peuvent être distingués selon que l'on cherche ou non à privilégier lisibilité et interprétabilité du système flou appris. Bien qu'aucune définition précise de ces concepts ne se soit réellement imposée, ils sont aujourd'hui associés à la notion de localité des règles ([Bikdash99], [Jin00]) plus qu'à leur nature même (Takagi-Sugeno ou Mamdani). Dans ce contexte, la partition des univers de discours est essentielle, notamment les sous-ensembles flous associés aux entrées doivent rester "distingables" pour pouvoir être étiquetés ([deOliveira95], [Pedrycz97]). L'identification de structure du système flou, ou plus précisément la construction des partitions d'entrée, est alors difficilement réalisable par des méthodes de coalescence de type FCM (Fuzzy C-Means) exploitées à profit quand seule la précision numérique est recherchée ([Yoshinari93], [Dickerson96], [Wang96], [Kim97]). Restent donc envisageables deux approches antagonistes du partitionnement : la première consiste à augmenter petit-à-petit le nombre de sous-ensembles flous exploités ([Takagi85], [Higgins94], [Rojas00]), alors que la seconde vise au contraire à diminuer ce nombre par sélection ou regroupement de règles ([Nie96], [Jin00], [Cordon00]). Dans le premier cas, une initialisation minimale est réalisée alors que dans le second, la base de règles est au départ de taille importante.

Une structure de base de règles étant fixée, un réglage fin du système flou est généralement réalisé selon une approche paramétrique. L'utilisation de différents outils d'optimisation (descente de gradient [Guély94], [Kim97], moindres carrés [Sugeno91], [Wang96], méthodes neuronales [Jang93], [Lin95], algorithmes génétiques [Farag98], [Cordon99], ...) est envisageable. Si des résultats numériques satisfaisants peuvent être obtenus, c'est souvent au détriment de la lisibilité du système flou appris.

La méthodologie proposée est en fait une émanation du principe des équivalences modales qui repose sur les constatations suivantes :

*Un partitionnement régulier des univers de discours associés aux variables d'entrée n'est pas souhaitable lorsque la loi cible est non linéaire.*

*La mise en oeuvre du principe des équivalences modales ne nécessite pas la connaissance totale de la loi cible puisque seuls les points modaux sont utilisés pour générer la base de règles.*

La première remarque conduit à réfléchir sur la façon de partitionner un univers numérique en sous-ensembles flous. Il apparaît clairement que le choix d'une partition détermine les valeurs modales de la variable associée. A l'inverse, le choix d'un ensemble de valeurs modales permet de construire une partition si la forme et le recouvrement des fonctions d'appartenance sont supposés connus. La prise en compte de la seconde constatation permet alors de réduire le problème de la synthèse d'un système flou à celui de la sélection d'un ensemble de points modaux. Dans une problématique d'apprentissage, cette dernière peut être réalisée en choisissant les points modaux parmi les points d'apprentissage. La solution la plus simple semble être d'utiliser tous les points d'apprentissage comme points modaux. Elle s'avère cependant peu efficace dans la mesure où le système flou construit comportera un nombre de règles très élevé. Dans un contexte bruité, il aura de plus perdu toute capacité de généralisation de par le fait qu'il "colle" aux données (problème de surapprentissage). Reste donc à fixer un critère de décision sur le fait qu'un point d'apprentissage doive devenir un point modal.

## VII.2 Technique d'apprentissage

La méthodologie proposée s'inscrit dans un mécanisme de construction incrémentale du système flou. Elle repose sur un enrichissement itératif du système flou par ajout de nouveaux symboles et des règles correspondantes et ce, jusqu'à obtention d'une erreur d'apprentissage satisfaisante. La base de règles est initialisée de manière minimale en considérant uniquement deux symboles sur chaque partition d'entrée ([Higgins94], [Jin00]). Le système flou ainsi obtenu est testé sur l'ensemble d'apprentissage de façon à déterminer le point générant l'erreur d'approximation maximale. Celui-ci est alors choisi pour devenir point modal ce qui permet d'annuler l'erreur d'approximation au point considéré. Le système flou est donc augmenté afin d'introduire le nouveau point modal. En pratique, cela se traduit par la définition d'un nouveau symbole sur chaque partition d'entrée et l'ajout d'une nouvelle règle. Le processus d'affinage du système flou est ensuite réitéré jusqu'à ce que des performances satisfaisantes soient atteintes. Il est à noter que la méthodologie proposée est cohérente avec la condition nécessaire d'approximation universelle établie dans [Ying97] et [Ding00]. Celle-ci exprime la nécessité qu'à tout extrémum d'une fonction à approximer soit associé un point modal.

La Fig. 20 illustre le processus d'insertion d'un nouveau symbole de valeur modale fixée ( $x^*$ ) dans une partition.

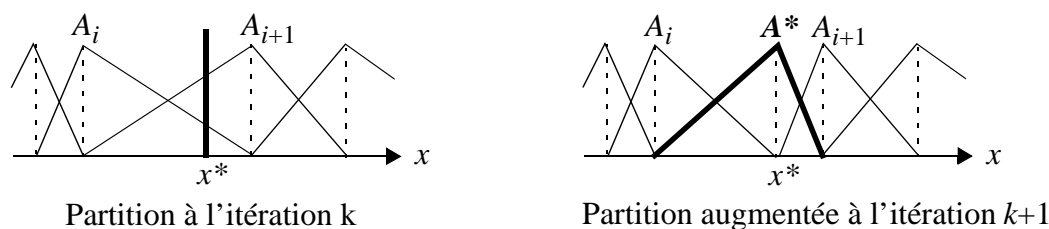


Fig. 20 : Ajout d'un nouveau symbole sur une partition.

Si l'on considère un problème d'apprentissage à deux entrées et une sortie, l'intégration d'un nouveau point modal  $P(x^*, y^*, z^*)$  dans le système flou nécessite :

- la création d'un symbole  $A^*$  de valeur modale  $x^*$  sur la partition des  $x$ ,
- la création d'un symbole  $B^*$  de valeur modale  $y^*$  sur la partition des  $y$ ,
- l'introduction de la règle “ **Si**  $x$  est  $A^*$  et  $y$  est  $B^*$  **alors**  $z = z^*$  ” dans la base.

La réalisation de ces trois étapes induit une base de règles incomplète (cf Fig. 21). Deux situations se présentent alors selon que les régions floues concernées contiennent ou non des points d'apprentissage. Dans le premier cas, il est indispensable de compléter les règles correspondantes pour pouvoir procéder à l'itération suivante. Le mécanisme mis en oeuvre consiste alors à approximer la sortie inconnue du point modal par celle du point d'apprentissage le plus “ressemblant”. Ce dernier est sélectionné par maximisation du degré d'appartenance à la région floue considérée, autrement dit du degré d'activation de la règle à compléter ([Wang92b], [Sudkamp94]). Une approche moins sensible au bruit est préconisée dans [Sudkamp94] et [Ishibuchi94] où la conclusion de règle est déterminée par moyenne pondérée des sorties des points d'apprentissage appartenant à la région floue considérée. [Nozaki97] introduit dans la définition des poids un paramètre permettant d'unifier les deux stratégies (maximum et moyenne pondérée) dans un formalisme unique.

	$A_i$	$A^*$	$A_{i+1}$	
	....	....		....
$B_j$	....	$z_{i,j}$		$z_{i+1,j}$
$B^*$			$z^*$	
$B_{j+1}$	....	$z_{i,j+1}$		$z_{i+1,j+1}$
	....	....		....

Fig. 21 : Base de règles à compléter.

Dans le cas où aucun point d'apprentissage n'est localisé dans le sous-espace d'entrée afférent, il n'est pas nécessaire de compléter la base de règles pour pouvoir poursuivre l'apprentissage dans la mesure où cette règle ne sera jamais activée. Toutefois, lorsque le système appris sera utilisé en généralisation, il est possible que l'absence de règles interdise le calcul de la sortie. Il est toujours envisageable de procéder à un remplissage ultérieur de la base de règles ([Sudkamp94]) ou d'interpoler directement au niveau des règles ([Tikk00]). Nous préférons traiter le problème au moment de l'apprentissage où l'intégralité des données est encore disponible, plutôt qu'a posteriori où ne reste accessible qu'un “résumé” de l'information initiale sous la forme d'une base de règles incomplète. La méthode de remplissage mise en oeuvre consiste à remplacer la conclusion manquante par la sortie du point d'apprentissage le plus proche de la région floue concernée, au sens de la distance à son point modal.

La technique proposée permet de réaliser un prototypage rapide du jeu de règles mais aussi des partitions d'entrée. Elle permet notamment de dimensionner le système flou en termes de nombre de symboles et donc de règles. Il est à noter qu'une version améliorée de l'algorithme est présentée dans [Galichet97]. Cette dernière autorise un nombre de termes linguistiques différent pour chaque variable d'entrée. Un nouveau symbole n'est alors ajouté sur une partition d'entrée que si sa non-cr ation emp che de r duire suffisamment l'erreur d'approximation au point d'apprentissage consid r . D'un point de vue num rique, le syst me flou obtenu peut  tre optimis  ult rieurement. On obtient alors une construction en deux  tapes du syst me flou telle que pr conis e dans [Ishibuchi94]. Une autre possibilit  consiste   introduire une optimi-

sation de paramètres à chaque itération de la construction du système flou. Cette approche, développée en collaboration avec A. Evsukoff ([Evsukoff02]) conduit à des résultats satisfaisants, tant en terme de précision numérique qu'en terme de lisibilité de la base de règles.

### VII.3 Application : prédiction de la tension artérielle

La technique développée a été utilisée pour la prédiction des variations de la pression artérielle dans le cadre d'une collaboration avec le LAG, le CHU de Grenoble et l'entreprise ELA Recherche, spécialisée dans l'appareillage médical. Cette étude a été soutenue par la région Rhône-Alpes sous la forme d'un projet Emergence. Globalement, le projet vise à améliorer le traitement de l'hypertension par un suivi continu de la pression artérielle du malade. Cet objectif nécessite une évaluation du niveau tensionnel en ambulatoire. L'interprétation d'une mesure tensionnelle réalisée en ambulatoire s'avère cependant délicate en raison d'une forte corrélation de celle-ci avec l'activité physique du patient au moment de la mesure. Ainsi, il est important de "corriger" la mesure selon le contexte dans lequel elle a été effectuée. La stratégie développée consiste en fait à évaluer la variation de pression (*VP*) pouvant être imputée à l'activité du sujet, cette dernière étant "reconstruite" à partir de mesures d'accélération corporelle (*ACC*) et de fréquence cardiaque (*FC*). Dans le cadre du projet, un protocole expérimental a été mis en oeuvre et a permis d'effectuer une première campagne de mesures sur 16 sujets sains normotendus. Pour chacun d'entre eux, deux enregistrements sur 24h sont disponibles regroupant les mesures de *VP*, *ACC* et *FR*. Les données relatives au premier jour sont alors utilisées pour l'apprentissage alors que celles du deuxième jour sont exploitées pour tester les performances en prédiction du système flou appris.

Dans un premier temps, des modèles individuels ont été construits en utilisant la technique d'apprentissage développée [Charbonnier00]. La Fig. 22 illustre les résultats obtenus sur un patient. Si l'on considère les performances en termes de moyennes horaires, les erreurs de prédiction sont satisfaisantes puisqu'elles sont de l'ordre de la précision de la mesure (+/- 7mmHg).

Dans un deuxième temps, un modèle de population a été appris à partir de l'ensemble des données d'apprentissage et testé individuellement sur chaque patient [Galichet00]. Les performances obtenues restent satisfaisantes. Notons que les utilisations potentielles des deux types de modèles sont différentes. Si les modèles individuels sont exploitables pour évaluer l'efficacité d'un traitement thérapeutique, les modèles de population peuvent, quant à eux, servir de référence à atteindre dans un système d'injection automatique de produits médicamenteux.

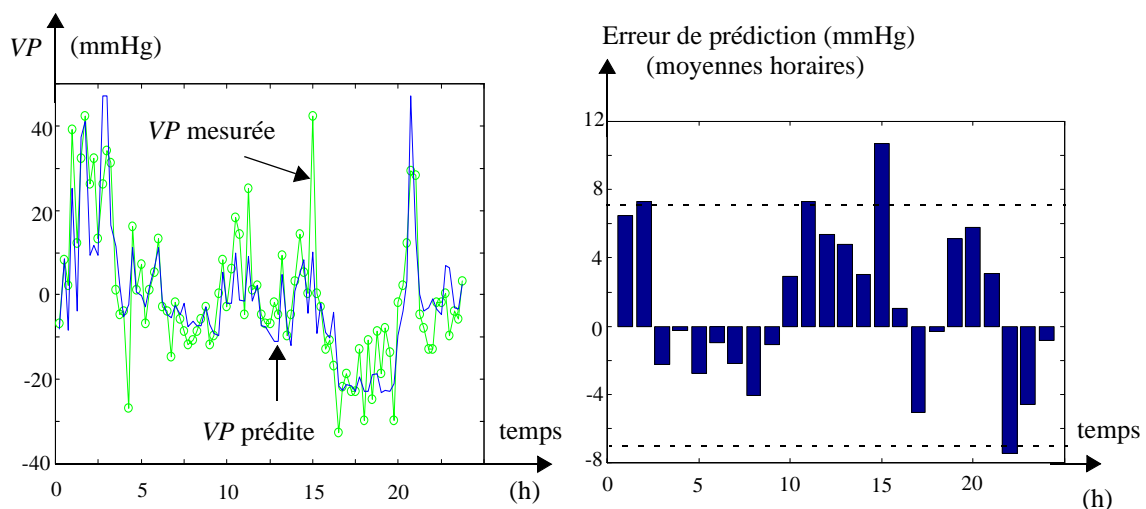


Fig. 22 : Prédiction des variations de pression (modèle individuel).

## VIII Propriétés structurelles des systèmes de Sugeno à conclusion constante

Depuis le début de ce chapitre, la synthèse des systèmes flous a été abordée d'un point de vue global. Autrement dit, on a cherché à construire des systèmes flous pour lesquels un comportement global est assuré. Selon le principe des équivalences modales, il apparaît qu'un positionnement correct des points modaux permet d'atteindre cet objectif. Reste maintenant à analyser les comportements locaux générés, c'est-à-dire la nature de l'interpolation entre points modaux. Une première approche dans ce sens a été faite dans le cas des systèmes linéaires puisqu'il a été montré qu'une interpolation linéaire est obtenue lorsque les points modaux sont répartis sur un hyperplan (cf paragraphe V). On se propose ici d'étendre le problème au cas d'une distribution quelconque des points modaux de façon à obtenir une expression analytique de la sortie d'un système flou. Cette démarche s'inscrit dans l'optique d'aborder par la suite la synthèse de contrôleurs flous permettant d'assurer stabilité et robustesse des systèmes bouclés. Cet aspect sera d'ailleurs développé dans le paragraphe suivant. L'ensemble de ces travaux a été réalisé dans le cadre de la thèse de Reda Boukezzoula ([Boukezzoula00]).

### VIII.1 Décomposition en sous-systèmes élémentaires

L'hypothèse d'un partitionnement triangulaire strict des univers de discours permet de garantir que chaque entrée est décrite par au plus 2 symboles. Ainsi, dans le cas d'un système à  $n$  entrées, au maximum  $2^n$  règles sont activées pour un vecteur d'entrée quelconque. La sortie générée par le système flou se réduit alors à celle produite par le sous-système flou composé des  $2^n$  règles activées. Le système flou global peut donc être vu comme une collection de sous-systèmes flous élémentaires. Chacun de ceux-ci est défini sur la "maille" floue (rectangle, parallélépipède, ...) construite à partir des entrées modales la délimitant. Dans ce contexte, l'étude d'un système flou peut être appréhendée à partir d'une analyse générique d'un sous-système flou élémentaire. Cette approche classiquement utilisée ([Zeng96c], [Li97], [Ying97], [Li99], [Sugeno99]) n'est d'ailleurs pas restreinte à un partitionnement triangulaire strict des univers d'entrée. Ainsi, elle est exploitée dans [Bikdash99] dans le cas d'un partitionnement strict sous la contrainte que les fonctions d'appartenance aient une dérivée nulle aux points modaux (splines, cosinus carré, ...). La contrainte d'une partition stricte peut également être levée ([Ding00]) sans remettre en question la décomposabilité du système flou global.

Dans la suite de ce paragraphe, toutes les propriétés énoncées correspondent à un partitionnement triangulaire strict et sont valides pour des systèmes flous élémentaires à  $n$  entrées. Toutefois, pour alléger le texte, il n'en est donné ici aucune formulation analytique, ni démonstration. Ces dernières sont présentées en détails dans [Boukezzoula00]. Seul un exemple sur un cas à 2 entrées est présenté de façon à illustrer les propriétés considérées. Celui-ci est introduit via la Fig. 23 qui regroupe la définition de la maille floue concernée ( $[0, 1] \times [-1, 1]$ ), la base de règles correspondante et une visualisation de la surface 3D engendrée.

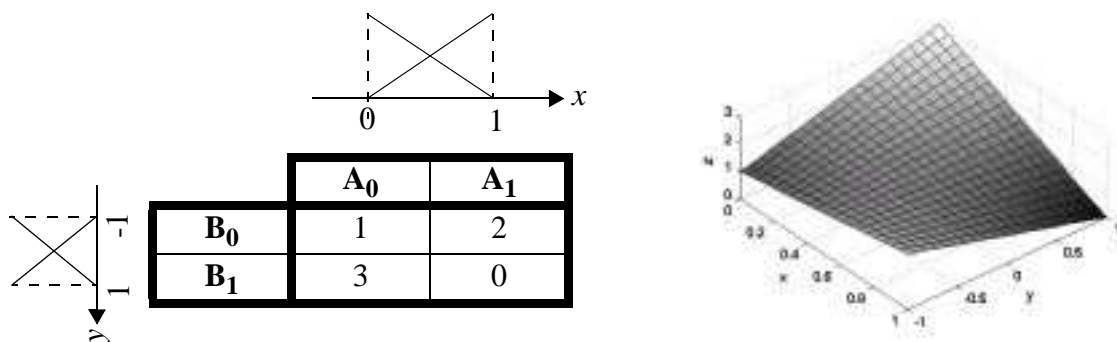


Fig. 23 : Exemple de sous-système flou élémentaire.



## VIII.2 Propriétés d'un sous-système élémentaire

L'idée de ce sous-paragraphe est de résumer un certain nombre de propriétés des systèmes flous élémentaires. Ces dernières seront par la suite exploitées, d'une part, pour proposer un algorithme d'inversion numérique d'un système flou, et d'autre part, pour établir un "pont" avec l'automatique conventionnelle, notamment via les techniques adaptatives.

### Paramétrisation

- [a] *La sortie d'un système flou élémentaire peut s'écrire sous une forme linéaire en les paramètres, c'est-à-dire comme étant le produit d'un vecteur ligne, dit vecteur de régression, et d'un vecteur colonne formé des conclusions de règles.*

L'équation (49) illustre ce type de paramétrisation sur l'exemple de la Fig. 23. Le vecteur  $W(x,y)$  regroupe les degrés de validité des prémisses de règles alors que le vecteur  $\Phi$  est formé des paramètres que représentent les sorties modales. Cette propriété est essentielle puisqu'elle permet une exploitation directe de bon nombre de travaux développés dans le cadre de l'identification et de la commande adaptative.

$$z = \left[ \frac{1}{2}(1-x)(1-y) \quad \frac{1}{2}(1-x)(1+y) \quad \frac{1}{2}x(1-y) \quad \frac{1}{2}x(1+y) \right] \begin{bmatrix} 1 \\ 2 \\ 3 \\ 0 \end{bmatrix} = W(x,y) \cdot \phi. \quad (49)$$

- [b] *La sortie d'un système flou élémentaire peut s'écrire sous une forme linéaire en les composantes d'un vecteur d'entrée étendu. Celui-ci regroupe un terme unitaire, toutes les entrées ainsi que tous les produits possibles d'entrées.*

Cette propriété est établie dans [Zeng96a] par une approche bornante de l'erreur d'approximation. L'équation (50) met en évidence ce type de décomposition sur l'exemple précédent. Le vecteur d'entrée étendu est formé de tous les termes  $x^i y^j$ ,  $(i, j) \in \{0,1\} \times \{0,1\}$ .

$$z = \begin{bmatrix} 2 & -1 & 1 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ xy \end{bmatrix}. \quad (50)$$

Cette paramétrisation permet de montrer facilement que tout système flou élémentaire est monotone, c'est-à-dire ne possède pas d'extrémum ([Ying97]).

### Affinité en chacune des entrées

- [c] *Un système flou élémentaire est affine en chacune de ses entrées.*

Cette propriété est illustrée par les équations (51) et (52) :

$$z = (y+2) + x(-1-2y) = \alpha_x(y) + x \beta_x(y) \quad \text{forme affine en } x, \quad (51)$$

$$z = (-x+2) + y(1-2x) = \alpha_y(x) + y \beta_y(x) \quad \text{forme affine en } y. \quad (52)$$

### Condition de linéarité en les entrées

- [d] *Un système flou élémentaire est linéaire si la différence de paramètres associée à un changement de symboles sur une entrée est identique pour toute combinaison symbolique des autres entrées et ce, quelle que soit l'entrée considérée.*

Cette condition est démontrée dans [Ying97] pour un système à deux entrées. La démarche mise en oeuvre est basée sur la paramétrisation  $[b]$  et consiste tout simplement à annuler le coefficient associé au terme croisé  $xy$ .

La non-linéarité du sous-système exemple est matérialisée à la Fig. 24. Un changement de symbole sur  $x$  (passage de  $A_0$  à  $A_1$ ) conduit à une différence de paramètre égale à  $\Delta_x^0$  lorsque  $y$  est décrit par  $B_0$  et égale à  $\Delta_x^1$  lorsque  $y$  est décrit par  $B_1$ . Ces deux valeurs étant différentes, le système considéré n'est pas linéaire. La même constatation peut être établie en considérant un changement de symbole sur  $y$ . On peut vérifier que les systèmes synthétisés en début de chapitre dans un contexte linéaire (cf Fig. 13) sont effectivement linéaires.

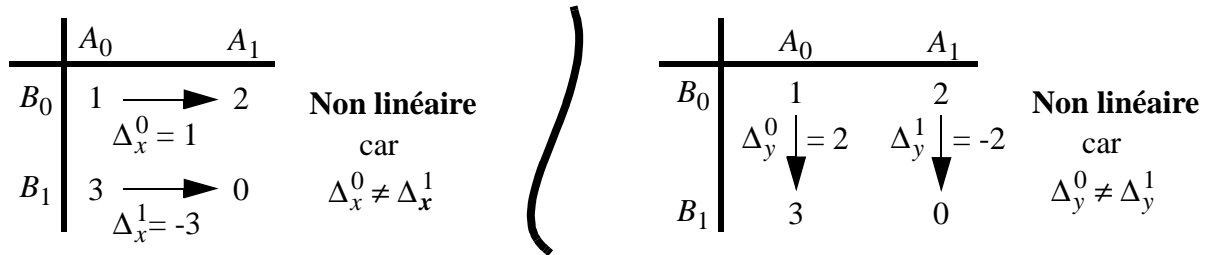


Fig. 24 : Non-linéarité d'un système flou élémentaire.

La Fig. 25 illustre graphiquement la condition de linéarité pour un système à 3 entrées, chaque point du parallélépipède correspondant à un paramètre. Dans la mesure où  $\Delta e_3$  peut être déterminé à partir de  $\Delta e_1$  et  $\Delta e_2$ , il est inutile d'analyser les variations de paramètres selon la variable  $e_3$ . De manière plus générale, pour un système à  $n$  entrées il est suffisant de vérifier les variations de paramètres relatives à  $n-1$  variables.

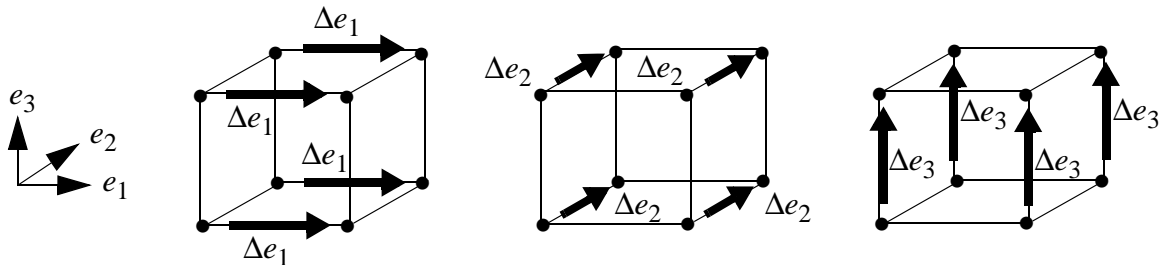


Fig. 25 : Condition de linéarité d'un sous-système élémentaire à 3 entrées.

### Transformation d'un système à prémisses composées en un système à prémisses simples

[e] *Un système flou élémentaire à prémisses composées peut être transformé en un système à prémisses simples mais à conclusion fonctionnelle.*

Pour un système à  $n$  entrées, les  $n-1$  variables éliminées des prémisses de règles sont réintroduites dans les conclusions. La base de règles à l'origine composée de  $2^n$  règles est réduite à 2 règles.

L'équation (53) illustre cette transformation sur le système exemple lorsque la variable  $y$  est éliminée des prémisses de règles.

$$\begin{aligned} \text{Si } x \text{ est } A_0 \text{ alors } z &= y+2, \\ \text{Si } x \text{ est } A_1 \text{ alors } z &= 1-y. \end{aligned} \quad (53)$$

Le même mécanisme aboutit au système (54) lorsque c'est la variable  $x$  qui est considérée.

$$\begin{aligned} \text{Si } y \text{ est } B_0 \text{ alors } z &= 1+x, \\ \text{Si } y \text{ est } B_1 \text{ alors } z &= 3-3x. \end{aligned} \quad (54)$$

### VIII.3 Inversion d'un système flou

L'objectif de l'inversion est illustré à la Fig. 26 dans le cas d'un système à  $n$  entrées que l'on cherche à inverser par rapport à la variable  $e_n$ . La problématique associée peut s'énoncer de la manière suivante :

*Etant donné un système flou connu, implantant le transfert  $f$  entre les entrées  $e_1, e_2, \dots, e_{n-1}, e_n$  et la sortie  $s$ , comment réaliser le transfert inverse  $g$  qui, pour les entrées  $e_1, e_2, \dots, e_{n-1}$  et  $s$ , produit une sortie  $y$  qui soit égale à  $e_n$  ?*

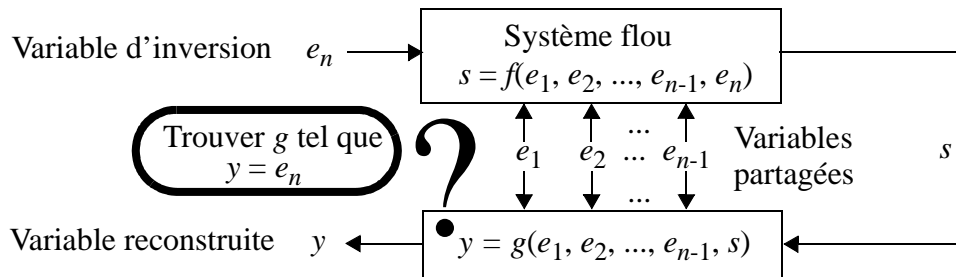


Fig. 26 : Inversion d'un système flou.

La première stratégie envisageable consiste à rechercher le transfert  $g$  sous la forme d'un système flou ([Babuska95], [Baranyi96], [Baranyi97]). Cette approche met en oeuvre un mécanisme d'inversion règle par règle par permutation des parties prémisses et conséquences. Si une telle procédure permet l'obtention d'une solution exacte dans le cas de systèmes mono-entrée, mono-sortie ([Babuska95]), son exploitation pour des systèmes multi-entrées induit des bases de règles incomplètes. Leur remplissage requiert l'utilisation de symboles "fictifs" ([Baranyi96], [Baranyi97]) puisque la définition de ces derniers repose sur des fonctions d'appartenance dont le support sort des univers de discours fixés. Dans le cas de systèmes de Takagi-Sugeno à conclusions constantes, la recherche d'un système flou inverse est en fait vouée à l'échec ([Baranyi97]). En effet, si l'on se réfère à la propriété **[b]** énoncée ci-dessus, il est clair que l'inversion analytique de  $f$  aboutit à une fonction  $g$  de nature homographique et donc non représentable de façon exacte par un système flou. A titre d'exemple, l'inversion par rapport à  $x$  du système flou élémentaire analysé précédemment conduit à la fonction  $g(y, z) = (y-z+2)/(1+2y)$  non représentable par un système flou.

Une seconde stratégie d'inversion consiste à utiliser une méthode itérative pour réaliser le transfert inverse  $g$  ([Kueon95], [Varkonyi99], [Park00]). Diverses techniques peuvent alors être exploitées (recherche dichotomique [Kueon95], descente de gradient [Park00], combinaison d'un algorithme génétique et de la méthode de Newton [Varkonyi99]). L'inconvénient de ce type d'approche est que la solution fournie n'est pas exacte et qu'aucune garantie ne peut être apportée quant à la convergence des algorithmes. De plus, la nature itérative de ces derniers induit un temps de calcul important qui doit être répété pour chaque nouveau vecteur d'entrées présenté.

Enfin, quelle que soit la méthodologie choisie, le cas des solutions multiples n'est pas traité. En ce qui concerne l'inversion directe des règles, elle est généralement réservée aux bases de règles monotones pour lesquelles l'unicité de solution est garantie ([Babuska95]). Quant aux méthodes itératives, l'existence de solutions multiples se traduit par une convergence vers des solutions différentes selon les conditions initiales imposées. Pour prendre en compte l'éventualité de plusieurs solutions admissibles, l'inversion est ici abordée de manière locale, c'est-à-dire sur les sous-systèmes élémentaires aptes à fournir une solution. L'inversion du système

global est ainsi appréhendée par inversion de certains de ses constituants.

La stratégie proposée pour inverser un sous-système élémentaire s'appuie sur une démarche purement analytique et s'écarte donc des deux approches évoquées précédemment. Elle repose en fait sur une exploitation directe de la propriété [c] des systèmes flous élémentaires. En effet, dans la mesure où les paramètres du système flou sont connus, son expression analytique peut être déterminée et mise sous une forme affine en l'entrée  $e_n$ , c'est-à-dire :

$$s = \alpha_{e_n}(e_1, e_2, \dots, e_{n-1}) + e_n \beta_{e_n}(e_1, e_2, \dots, e_{n-1}). \quad (55)$$

L'inversion par rapport à  $e_n$  devient alors une simple inversion de l'équation (55), à savoir :

$$e_n = g(e_1, e_2, \dots, e_{n-1}, s) = \frac{s - \alpha_{e_n}(e_1, e_2, \dots, e_{n-1})}{\beta_{e_n}(e_1, e_2, \dots, e_{n-1})} \quad (56)$$

L'existence d'une solution inverse unique nécessite d'une part, que l'équation (56) soit calculable ( $\beta_{e_n}(e_1, e_2, \dots, e_{n-1}) \neq 0$ ), d'autre part que le résultat évalué appartienne effectivement à la maille floue considérée. Le cas particulier où  $\beta_{e_n}(e_1, e_2, \dots, e_{n-1}) = 0$  se traduit, selon la valeur de  $s - \alpha_{e_n}(e_1, e_2, \dots, e_{n-1})$ , soit par une infinité de solutions, soit par l'inexistence de solution. Un résultat similaire peut être obtenu à partir de la propriété [e] des sous-systèmes élémentaires. La démarche consiste alors à transformer le système initial en un système composé de 2 règles à conclusion fonctionnelle. Les variables intervenant dans la partie conséquence des règles étant de valeur connue, l'inversion peut être réalisée par simple permutation entre prémisse et conclusion ([Babuska95]). A noter toutefois que les règles ainsi obtenues ne sont valables que pour des valeurs données de  $e_1, e_2, \dots, e_{n-1}$  et ne peuvent donc être interprétées comme constituant une base de règles inverses générique.

D'un point de vue global, si l'on considère que les variables  $e_1, e_2, \dots, e_{n-1}$  ont une valeur connue, il est possible de déterminer les mailles floues auxquelles appartient le vecteur  $[e_1, e_2, \dots, e_{n-1}]$ . Elles sont au nombre de  $N_{e_n}-1$  si  $N_{e_n}$  représente le nombre de symboles utilisés pour décrire la variable d'inversion  $e_n$ . Les sous-systèmes correspondants sont donc inversés selon le mécanisme introduit précédemment (équation (56)). Ainsi, le problème global pourra ne pas avoir de solution, avoir une solution unique ou encore avoir de multiples solutions. La stratégie proposée présente l'avantage de fournir toutes les solutions exactes au problème d'inversion. D'un point de vue implantation pratique, elle permet une pré-détermination des expressions analytiques sur chaque maille afin de limiter les calculs à répéter pour chaque nouveau vecteur d'entrées.

## **IX Synthèse d'une loi de commande par inversion de modèle flou : approche par linéarisation entrée/sortie**

L'objectif de ce dernier paragraphe est d'effectuer la synthèse d'un contrôleur flou pour réaliser un suivi de trajectoire dans un contexte non linéaire. Les développements présentés s'appuient sur les résultats des sections précédentes dans la mesure où le contrôleur flou est utilisé en tant qu'approximateur d'une loi de commande non linéaire. Ils se différencient cependant par le fait que la loi de commande, de même que l'expression analytique du modèle de procédé, sont maintenant supposées inconnues. Les seules informations disponibles sont alors les grandeurs mesurées en ligne sur le procédé.

La problématique du suivi de trajectoire consiste à déterminer une loi de commande  $u$  permettant d'assurer que le système bouclé reste stable et que sa sortie  $y$  suive une trajectoire désirée  $y_d$ . Pour atteindre cet objectif, une structure de contrôle traditionnelle est ici exploitée dans un contexte flou.

## IX.1 Principe

L'approche par linéarisation entrée/sortie, déjà évoquée dans le paragraphe VI.2 pour un problème de stabilisation, est ici adaptée au suivi de trajectoire donnant lieu à la structure de commande illustrée à la Fig. 27.

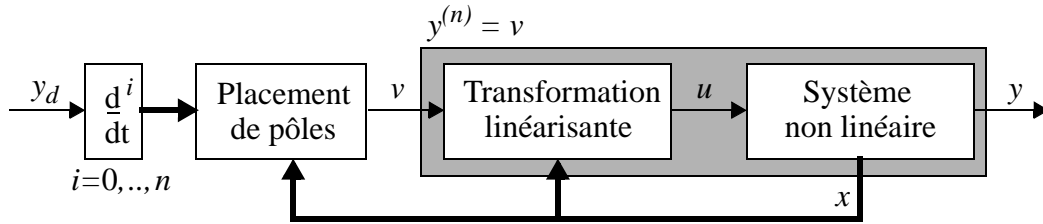


Fig. 27 : Commande par linéarisation entrée/sortie d'un système non linéaire.

Selon l'équation (43), reflétant la relation entrée/sortie associée au système non linéaire considéré, il apparaît clairement que la transformation linéarisante (44) est obtenue par inversion par rapport à la variable  $u$ . Ainsi, si un modèle flou de type TSK à conclusion constante représente de façon exacte la relation entrée/sortie du procédé, son inversion par rapport à  $u$  correspond à la transformation linéarisante. En pratique, la représentation floue de l'équation (43) donne généralement lieu à une erreur d'approximation  $\Delta$ , c'est-à-dire :

$$y^{(n)} = \text{TSK}(x, u) + \Delta. \quad (57)$$

L'introduction de la commande  $u = \text{TSKINV}(x, v)$ , obtenue par inversion du système flou (cf paragraphe VIII.3), conduit à l'obtention du système :

$$y^{(n)} = v + \Delta. \quad (58)$$

La structure de commande correspondante est illustrée à la Fig. 28. Sa mise en oeuvre pose deux difficultés majeures, à savoir l'obtention du modèle flou et la prise en compte de l'erreur de modélisation non assurée par l'architecture de base de la Fig. 28. La résolution de ces problèmes nous a conduits à intégrer de nouveaux composants fonctionnels dans la structure établie.

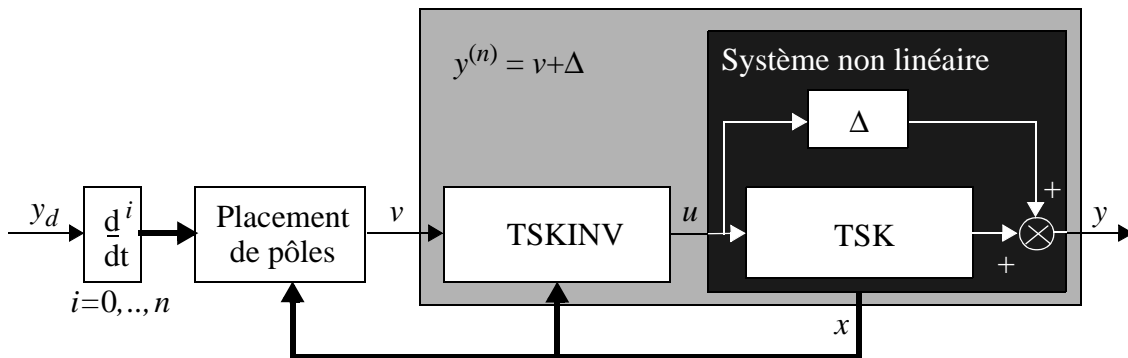


Fig. 28 : Commande par linéarisation entrée/sortie d'un système non linéaire représenté par un modèle flou entrée/sortie augmenté d'une incertitude additive.

## IX.2 "Robustification" de la structure de commande

Supposons dans un premier temps que le modèle flou du procédé soit connu ou tout au moins ait été identifié à l'aide d'une procédure d'apprentissage hors ligne telle que celle proposée dans le paragraphe VII. Selon la propriété [c] des systèmes flous exploités (affinité en chacune des entrées d'un sous-système), l'équation (57) peut être réécrite sous la forme :

$$y^{(n)} = \text{TSK}(x, u) + \Delta = \alpha_u(x) + u \beta_u(x) + \Delta. \quad (59)$$

Il est à noter que l'équation (59) ne signifie pas que le modèle flou est affine en commande. Par

contre, pour toute paire  $(x, u)$ , il est possible de déterminer quel système flou élémentaire est activé et donc d'exploiter les expressions analytiques correspondantes de  $\alpha_u$  et  $\beta_u$ . Le modèle global est en fait affine en commande par morceaux.

La prise en compte des incertitudes  $\Delta$  est réalisée via une composante de commande additive  $u_a$  selon le schéma de la Fig. 29, c'est-à-dire :

$$u = \text{TSKINV}(x, v) + u_a / \beta_u(x). \quad (60)$$

Par substitution de  $u$  dans la relation entrée/sortie (60), on obtient finalement :

$$y^{(n)} = v + u_a + \Delta. \quad (61)$$

Si l'on fixe  $u_a = -\Delta$ , il est clair que la composante additive compense intégralement l'incertitude. Toutefois, celle-ci étant inconnue, la loi de commande optimale n'est pas implantable et il est donc nécessaire de lui substituer une version dégradée n'exploitant que des informations disponibles. Deux approches ont été développées à cet effet dans [Boukezzoula00]. Par souci de lisibilité, seul leur principe est exposé ici, les démonstrations et développements correspondant pouvant être trouvés dans la référence citée.

La première stratégie de "robustification" proposée exploite une technique de mode de glissement. La composante additive est alors exprimée sous la forme :

$$u_a = \Delta_0 \text{signe}(e^T b), \quad (62)$$

avec  $\Delta_0$  borne supérieure de l'incertitude, i.e.  $|\Delta| \leq \Delta_0$ ,

$e = [e_0, \dot{e}_0, \dots, e_0^{(n-1)}]^T$  et  $e_0 = y_d - y$ ,  
et  $b$  vecteur ligne à  $n$  composantes.

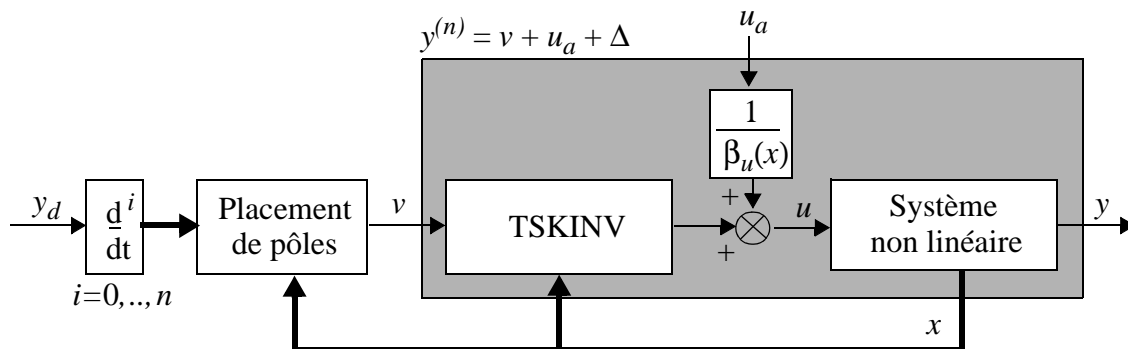


Fig. 29 : Intégration d'une composante de commande additive de "robustification".

La détermination de  $b$  est effectuée par une synthèse de Lyapunov permettant ainsi de garantir la stabilité de la structure de commande et la convergence vers zéro de l'erreur de suivi de trajectoire  $e_0$ .

La seconde technique développée est basée sur une stratégie  $H_\infty$ . L'objectif de la commande additive est alors de contraindre la norme infinie du transfert entre  $\Delta$  et  $e_0$ . La composante  $u_a$  est recherchée sous la forme :

$$u_a = G e, \quad (63)$$

$G$  étant un vecteur ligne à  $n$  composantes, déterminé par une optimisation  $H_\infty$  en utilisant les équations de Ricatti.

Ces deux méthodes de "robustification" ont été testées tant en simulation que sur un procédé réel. Leur comparaison s'avère délicate dans la mesure où l'approche  $H_\infty$  n'est pas toujours applicable. En effet, pour une contrainte d'atténuation fixée, il est impossible de garantir que l'optimisation  $H_\infty$  admette une solution. Toutefois, lorsque cette dernière existe, de bonnes

performances de suivi sont obtenues, notamment en régime transitoire. Quant à la synthèse par mode de glissement, elle présente l'avantage d'être toujours exploitable mais engendre des signaux haute fréquence indésirables. De plus, sa mise en oeuvre nécessite de fixer a priori la borne d'incertitude  $\Delta_0$ .

La Fig. 30 présente l'architecture utilisée pour tester les structures de commande développées sur des procédés réels [Boukezzoula01]. La machine client transmet au serveur la loi de commande, écrite sous la forme d'un schéma bloc Simulink, via une connection TCP-IP. Ce dernier, équipé du noyau temps réel XPCTarget et de cartes d'entrée/sortie, prend en charge le pilotage du procédé.

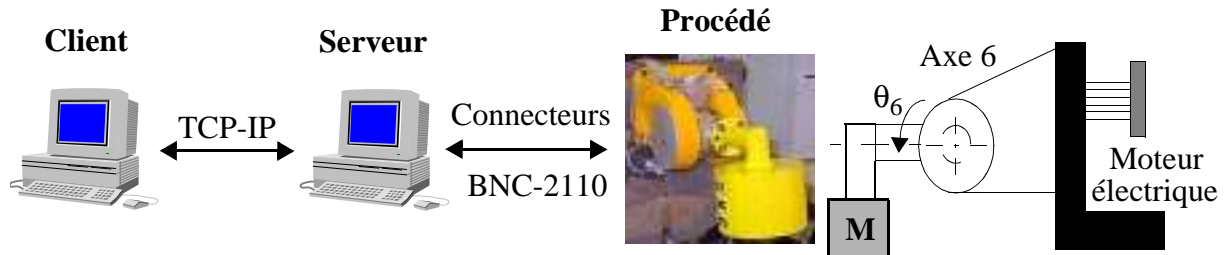


Fig. 30 : Implantation des structures de commande sur un poignet de robot.

La Fig. 31 regroupe les résultats obtenus par application d'une commande linéarisante floue, augmentée d'une composante  $H_\infty$ , dans le cadre d'un problème de suivi de trajectoire. Le procédé considéré est l'axe 6 d'un poignet de robot à six axes, ainsi qu'illustré à la Fig. 30. L'objectif de la commande est de forcer l'angle  $\theta_6$  à suivre la trajectoire sinusoïdale définie par  $y_d = (\pi/2) \sin t$ . Le modèle flou utilisé, composé de 12 règles, exploite les entrées  $\theta_6$  et  $\dot{\theta}_6$ . La vitesse angulaire n'étant pas mesurée est reconstruite par une dérivée numérique filtrée.

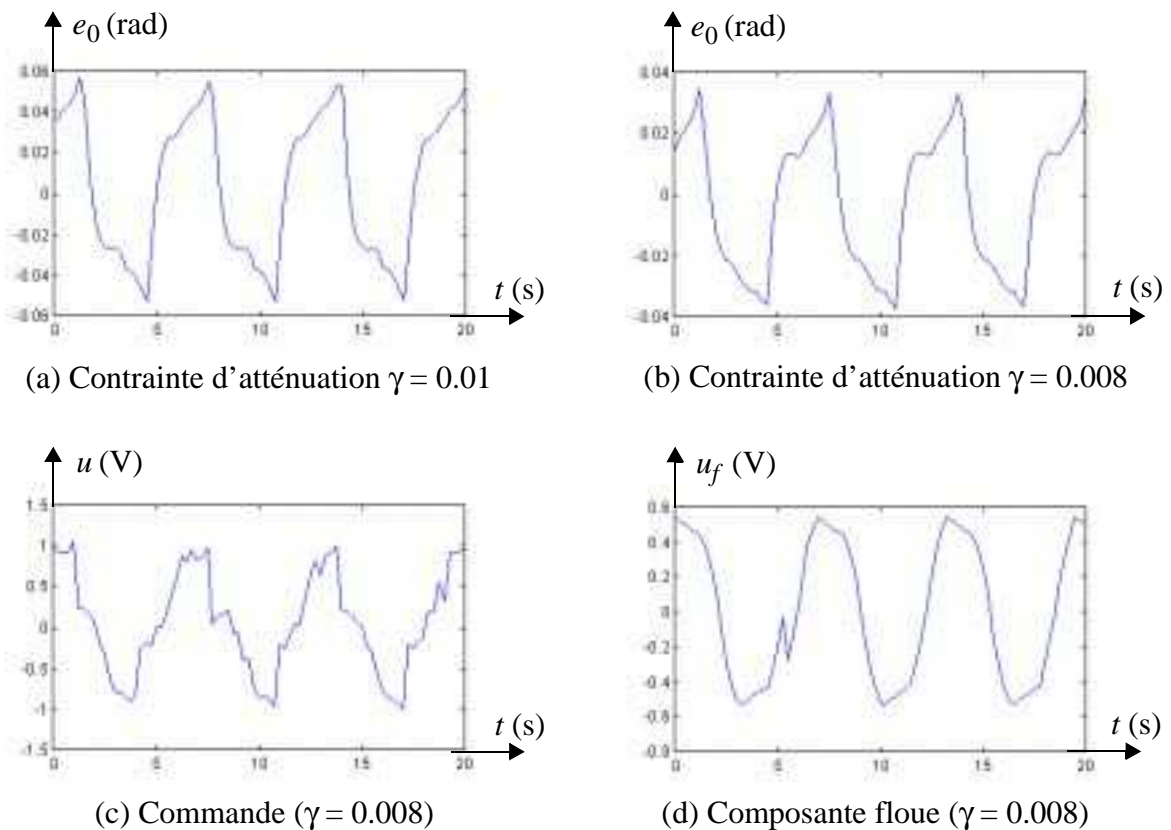


Fig. 31 : Commande de l'axe 6 d'un poignet de robot avec "robustification"  $H_\infty$

Les Fig. 31 (a) et (b) représentent l'évolution temporelle de l'erreur de suivi pour deux synthèses différentes de la composante  $H_\infty$ , les contraintes d'atténuation étant respectivement fixées à 0.01 et 0.008. Il est clair qu'un durcissement de la contrainte a permis d'améliorer la qualité du suivi de trajectoire sans qu'aucune dégradation de la commande n'apparaisse (cf Fig. 31 (c)). Reste enfin à souligner que l'usage exclusif d'une commande linéarisante floue (cf Fig. 31 (d)) ne permet d'atteindre de telles performances qu'au prix d'une augmentation significative du nombre de règles. Se pose alors de façon cruciale la question de l'identification de modèle abordée dans le paragraphe suivant.

### IX.3 Identification du modèle flou

Dans tout ce qui précède, le modèle flou exploité pour construire la loi de commande par inversion, est supposé disponible. Si ce n'est pas le cas, il est possible de l'identifier en ligne avec cependant le prérequis qu'au moins la structure du système soit connue. Sous cette contrainte, l'identification est ramenée à un problème d'estimation de paramètres appréhendable par une approche adaptative.

Suivant la paramétrisation [a] introduite dans le paragraphe VIII.2, le modèle flou inconnu exploité dans l'équation (59) s'exprime sous la forme :

$$\text{TSK}(x, u) = W(x, u) \phi, \quad (64)$$

où  $\phi$  représente le vecteur de paramètres constitué des conclusions de règles.

De la même manière, le système flou adapté s'écrit :

$$\hat{\text{TSK}}(x, u) = W(x, u) \hat{\phi}, \quad (65)$$

dans la mesure où il ne diffère du précédent que par ses conclusions de règles maintenant adaptées et regroupées dans le vecteur  $\hat{\phi}$ . Si l'on note  $\tilde{\phi}$  l'erreur de paramètres, c'est-à-dire  $\tilde{\phi} = \phi - \hat{\phi}$ , l'équation (59) devient finalement :

$$y^{(n)} = \text{TSK}(x, u) + \Delta = \hat{\text{TSK}}(x, u) + W(x, u) \tilde{\phi} + \Delta. \quad (66)$$

La loi de commande est construite de la même manière que précédemment à la nuance près que c'est maintenant le système flou adapté qui est inversé :

$$u = \hat{\text{TSKINV}}(x, v) + u_a / \hat{\beta}_u(x). \quad (67)$$

L'introduction de  $u$  dans la relation entrée/sortie du procédé conduit à l'obtention du système bouclé :

$$y^{(n)} = v + u_a + \Delta + W(x, u) \tilde{\phi}. \quad (68)$$

Reste maintenant à déterminer une loi d'adaptation du vecteur de paramètres permettant de satisfaire les objectifs fixés en termes de stabilité et de convergence de l'erreur de suivi. Deux approches différentes sont proposées et détaillées dans [Boukezzoula00]. Elles sont reprises succinctement ici.

La première consiste à déterminer la loi d'adaptation de façon à minimiser l'erreur de poursuite entre la sortie du procédé et la trajectoire désirée. Celle-ci, construite par une synthèse de Lyapunov selon une approche basée sur la théorie de la passivité, s'exprime sous la forme :

$$\dot{\hat{\phi}} = -\dot{\tilde{\phi}} = -\Lambda W(x, u) e^T b, \quad (69)$$

avec  $\Lambda$  matrice diagonale définie positive et  $b$  vecteur ligne à  $n$  composantes.

La seconde stratégie repose sur une minimisation conjointe de l'erreur de suivi de trajectoire et de l'erreur de prédiction. Elle reprend la philosophie de base de la méthode composite proposée dans [Slotine91] et étendue aux systèmes flous dans [Boukezzoula98]. L'algorithme initial, de type moindres carrés récursifs avec facteur d'oubli, s'avère cependant délicat à exploiter dans la mesure où les conditions de convergence annoncées sont difficilement véri-



fiables. Une version modifiée de la méthode composite est donc proposée dans [Boukezzoula00]. Celle-ci reprend les outils de synthèse évoqués précédemment (Lyapunov, passivité) pour construire une loi d'adaptation s'exprimant sous la forme :

$$\dot{\hat{\phi}} = -\dot{\tilde{\phi}} = -\Lambda W(x, u) [e^T b - e_p], \quad (70)$$

où  $e_p$  représente l'erreur de prédiction associée au modèle flou.

## X Conclusion

Ce chapitre nous a permis d'aborder différents aspects numériques liés au développement des contrôleurs flous. A titre de conclusion, il nous paraît maintenant utile de revenir sur quelques points importants avant d'envisager les perspectives que l'on peut associer à ce travail.

La première remarque concerne le choix d'implantation d'un contrôleur flou. Si l'on se cantonne aux aspects numériques, il est à nos yeux préférable d'utiliser un système de type Sugeno plutôt qu'un système de type Mamdani. Ce choix permet en effet d'éliminer les difficultés liées à l'exploitation de gains de normalisation ou encore de paramètres additionnels nécessaires à l'écriture des règles additives (constantes  $\alpha$  et  $\beta$  introduites dans (23)). Nous sommes d'ailleurs confortés dans cette opinion par le fait que nous n'avons trouvé aucun avantage à exploiter les opérateurs min/max préconisés par Mamdani, tout au moins dans un contexte de commande. Au contraire, l'usage de ces derniers entraîne l'apparition de non-linéarités qui s'avèrent incontrôlables et dont l'aspect répétitif sur chaque maille floue nous semble préjudiciable. Enfin, en anticipant quelque peu sur le chapitre à venir, vous verrez que le choix des systèmes de Sugeno à conclusions constantes est également validé dans un cadre linguistique.

Le second point sur lequel il nous paraît essentiel d'insister est que, sous l'hypothèse d'un partitionnement triangulaire strict des univers de discours d'entrée, un système flou de Sugeno à conclusions constantes n'est autre qu'un système multi-linéaire par morceaux. Ce résultat, probablement connu par les spécialistes du domaine, n'est que très rarement énoncé de façon explicite. Il est pourtant fondamental et se décline sous différents points de vue, ainsi qu'illustré par les deux paragraphes qui suivent.

En posant des contraintes sur les conclusions des règles, il est aisé d'obtenir une implantation purement linéaire d'un système de Sugeno. Cette façon de faire permet notamment d'aboutir à la réalisation exacte de PID flous ou encore de contrôleurs linéaires par retour d'état flous. Cet état de fait devrait clore la première génération de travaux liés à l'implantation floue de PI, PD, PID ou de tout autre loi de commande linéaire. Dès à présent, on voit d'ailleurs apparaître une seconde génération de PID flous, dans lesquels sont introduites des non-linéarités maîtrisées ([Hu99], [Mudi99], [Tao00]).

Toute fonction non linéaire peut être approximée par un système multi-linéaire par morceaux. La qualité de l'approximation dépend alors du nombre de morceaux considérés c'est-à-dire, dans un contexte flou, du nombre de règles. Il est donc possible de construire un modèle flou approximatif de tout système non linéaire. Ce dernier est exprimable sous une forme analytique, qui plus est, s'avère linéaire en les paramètres que représentent les conclusions de règles. Ces caractéristiques permettent d'imaginer l'intégration de modèles flous dans bon nombre de structures de commande conventionnelles. C'est ainsi que nous avons, par exemple, pu développer une structure de commande floue robuste adaptative par linéarisation entrées/sortie.

Cette dernière remarque nous conduit tout naturellement à envisager des perspectives immédia-

tes de développement. En effet, au même titre que les réseaux de neurones se sont “fuzzifiés”, il est tout à fait probable de voir apparaître dans un futur proche le qualificatif “flou” en association de divers types de commande. D’ailleurs, bien que cela n’ait pas été présenté ici, nous avons déjà entrepris des démarches en ce sens, notamment dans le cadre de la commande à modèle interne ([Boukezzoula02]). Si cette approche par “fuzzification” des architectures de commande conventionnelles n’a rien de révolutionnaire, elle présente cependant l’avantage de pouvoir aborder des problèmes non linéaires de façon relativement simple. Il n’en demeure pas moins que l’erreur d’approximation qu’elle peut induire rend essentielle la phase de synthèse d’une composante de “robustification”. Reste enfin à souligner que l’approximation, résultant de l’usage d’un système flou de Sugeno à conclusions constantes, différencie la stratégie évoquée ici de celle mise en oeuvre dans une implantation de type PDC (Parallel Distributed Controllers) où le modèle flou global est généralement supposé exact.

Pour une vision à plus long terme, il est sans doute important de réfléchir à la signification du qualificatif flou lorsqu’il est attribué à une commande ou à un modèle. Actuellement, il indique simplement que, d’un point de vue interne, certains transferts entrées/sortie sont exprimés sous forme de règles. D’un point de vue externe, il est impossible de différencier un module flou d’un quelconque mécanisme de calcul. Cette remarque révèle que l’aspect imprécis ou incertain n’est aucunement pris en compte dans les travaux actuels. A nos yeux, d’importants efforts sont encore à fournir pour analyser l’apport potentiel du flou dans le cadre de la commande des systèmes incertains. Dans ce contexte, il est nécessaire d’entamer une réflexion de fond sur différents points, parmi lesquels on peut citer :

- l’utilisation et la manipulation des équations différentielles floues,
- leur lien avec les systèmes à base de règles,
- l’exploitation possible de l’imprécis pour la synthèse d’un contrôleur,
- la remise en question de l’étape de défuzzification.

Enfin, il est clair qu’un développement conséquent des travaux nécessite également de se positionner par rapport aux avancées actuelles de la commande des systèmes incertains et notamment de situer l’outil flou par rapport aux approches probabilistes ou par intervalles.

## Chapitre II : Codage de l'expertise

### I Introduction

*“The past decade has witnessed a rapidly growing mathematization of control theory, with the result that the relation between the theory and its application has become more tenuous than ever.”*

Cette réflexion ne date pas d'aujourd'hui puisque qu'elle est extraite d'un papier de Zadeh paru en 72 ([Zadeh72]). Cette dernière est utilisée par l'auteur pour justifier le besoin de développer de nouvelles approches de commande, appropriées à la résolution de problèmes réels, difficilement abordables par le biais d'une analyse mathématique. Dans ce contexte, les premières applications de contrôle flou voient le jour. Elles reposent sur l'exploitation d'une expertise humaine, formalisée à partir de règles floues ([Mamdani75]). En ce sens, elles se distinguent effectivement des stratégies conventionnelles basées sur l'utilisation d'un modèle mathématique du procédé à piloter. La loi de commande est spécifiée de manière imprécise à l'aide de mots issus du langage naturel. Son implantation numérique reste cependant précise dans la mesure où elle intègre des interfaces numérique/symbolique et symbolique/numérique permettant de se ramener à des mondes numériques précis.

Trente ans après ces premiers développements, force est de constater que la remarque initiale de Zadeh s'applique maintenant au contrôle flou. La discipline a en effet évolué vers un usage pratiquement exclusif des systèmes flous à conclusions numériques de type TSK. Si ceux-ci présentent de bonnes capacités d'interpolation numérique, ils s'éloignent cependant des préoccupations initiales de l'approche linguistique préconisée par Zadeh ([Zadeh73]). Le tableau 4 illustre notre propos par une analyse rapide de la littérature récente du domaine. Il concerne exclusivement les dix derniers numéros de la revue IEEE Trans. on Fuzzy Systems. Durant la période considérée (février 2000 à août 2001), 126 articles ont été publiés parmi lesquels 46 traitent de contrôle flou. Dans le tableau 4, ces derniers sont répertoriés selon la nature des travaux qu'ils présentent et selon l'approche qui est mise en oeuvre. S'il convient bien évidemment d'éviter toute conclusion hâtive, ces quelques chiffres nous amènent cependant à établir deux constatations. La première concerne le faible taux de publication d'applications industrielles. Elle est néanmoins à prendre avec une extrême prudence dans la mesure où une seule revue est ici considérée. La seconde est relative à l'énorme proportion de travaux théoriques ou méthodologiques basés sur une approche numérique de la commande floue. Diverses raisons peuvent être trouvées à cet état de fait. On peut notamment invoquer les habitudes de travail des automaticiens, les difficultés à formaliser une approche linguistique générique du contrôle, la volonté de justifier les nouvelles stratégies par rapport à l'existant, l'absence de critères qualitatifs de performances, le manque d'outils de simulation adaptés, ... Toujours est-il qu'il sem-

ble que le contrôle flou soit en train de progressivement rejoindre la communauté de l'automatique conventionnelle.

Nature	Approche	Nombre de papiers
Théorie et méthodologie	Numérique	39
	Linguistique	2
Applications industrielles	Numérique	3
	Linguistique	2

Tableau 4 : Classification des articles de contrôle parus dans IEEE Transactions on Fuzzy Systems (10 numéros parus de février 2000 à août 2001)

Il est légitime de se réjouir de cette évolution puisqu'elle permet à la fois d'asseoir les travaux issus de la commande floue et d'ouvrir de nouvelles pistes d'exploration en commande non linéaire. Il nous semble cependant essentiel de veiller à ne pas retomber dans les travers d'une mathématisation à outrance, obstacle à la prise en compte des préoccupations de terrain. C'est l'idée développée dans le présent chapitre qui s'écarte en effet du monde numérique en autorisant l'intégration de connaissances expertes dans les structures de commande proposées. Ainsi, si le premier chapitre a eu pour objectif unique de construire des systèmes flous garantissant un comportement numérique, le second vise maintenant à les enrichir en y introduisant l'expertise disponible sur les procédés considérés.

D'un point de vue concret, cette possibilité est offerte par une implantation symbolique des systèmes flous. Cette dernière, issue de la typologie des contrôleurs flous que nous avons développée ([Foulloy95a]), s'avère donc être le fil conducteur des différents paragraphes qui suivent. Ainsi, dans un premier temps, elle est présentée en détails de façon à pouvoir étudier son utilisation potentielle pour des entrées numériques imprécises, notamment dans le cadre du raisonnement approximatif ([Dubois99]). Les liens qui la rattachent aux contrôleurs TSK à conclusions constantes sont ensuite exhibés. Ils en résultent une méthodologie de conversion permettant de transformer, de manière exacte, un système TSK à conclusions constantes en son homologue linguistique.

L'implantation linguistique des systèmes flous nous permet d'aborder le codage de l'expertise selon deux approches distinctes. La première consiste à réaliser la construction du système flou directement au niveau linguistique alors que la seconde exploite une synthèse préalable du contrôleur flou au niveau numérique. Cette dernière façon de faire permet de capitaliser la connaissance numérique acquise sur le contrôle d'un procédé et de la faire évoluer par intégration d'une expertise humaine. Enfin, l'implantation linguistique d'un système flou autorise une décomposition de la tâche globale en termes de raisonnement et de conversions numérique/symbolique et symbolique/numérique. Cette caractéristique permet de distribuer les traitements inhérents à chaque sous-tâche et par là même d'intégrer l'expertise disponible dans chacun des composants élémentaires. Une telle pratique autorise notamment le développement de structures hiérarchiques manipulant des variables de description intermédiaires auxquelles ne sont associées aucunes grandeurs numériques.

Ainsi qu'évoqué ci-dessus, l'intégration d'une expertise dans une structure de contrôle peut être réalisée de différentes façons et à divers niveaux. Ce chapitre aborde la problématique à partir de cas réels ([Galichet94a], [Dussud98]), mais avec le souci d'une structuration des différentes approches permettant d'envisager une démarche méthodologique générique.

## II Manipulation d'informations symboliques

### II.1 Description et signification

Le partitionnement d'un univers de discours numérique en sous-ensembles flous et l'affectation d'un label à ces derniers permet d'interfacer les mondes numérique et linguistique. Les mots se voient attribuer une signification sur l'univers numérique de référence et il devient possible de décrire une valeur numérique à l'aide de mots.

D'un point de vue mathématique, les opérations conjointes de partitionnement flou et d'étiquetage linguistique définissent une relation floue  $R$  sur le produit cartésien  $X \times LX$ , où  $X$  représente l'univers numérique et  $LX$  l'ensemble des mots associés. Cette relation induit une correspondance entre termes linguistiques et valeurs numériques. A ce titre, elle est assimilée à un langage dans [Zadeh71], où sont définies les notions complémentaires de signification et de description. La signification d'un terme linguistique de  $LX$  est engendrée par projection de  $R$  sur  $X$  et correspond donc à une fonction  $M$  définie de  $LX$  vers  $F(X)$ , où  $F(X)$  est l'ensemble des parties floues définies sur  $X$ . De même, la description d'un nombre est obtenue par projection de la relation  $R$  sur  $LX$ . C'est donc une fonction  $D$  définie de  $X$  vers  $F(LX)$  où  $F(LX)$  représente l'ensemble des parties floues définies sur  $LX$ . Ces définitions conduisent à l'égalité suivante :

$$\forall x \in X, \forall L \in LX, \quad \mu_{D(x)}(L) = \mu_R(x, L) = \mu_{M(L)}(x), \quad (71)$$

illustrée à la Fig. 32 dans le cas où la variable  $x$ , représentant la température de l'eau du lac d'Annecy, est égale à 23°C. L'ensemble des termes linguistiques utilisé pour décrire  $x$  est composé des mots **G**laciaire, **F**roide, **Ba**ignable, **Bo**nnie, **C**haude, c'est-à-dire  $LX = \{G, F, Ba, Bo, C\}$ . La description floue de la température 23°C est le sous-ensemble flou linguistique  $D(23) = 0.4/Bo + 0.6/C$ . Cette notation introduite par Zadeh permet d'exprimer un sous-ensemble flou défini sur un univers discret sans faire référence explicite à sa fonction d'appartenance. Elle consiste en fait à spécifier l'ensemble des couples  $(\mu_{D(x)}(L), L)$  sous la forme  $\alpha/L$  où  $\alpha$  représentant le degré d'appartenance de  $L$  à la partie floue considérée. Pour simplifier l'écriture, les éléments de degré d'appartenance nul, c'est-à-dire  $0/L$ , sont généralement éliminés. L'opérateur  $+$  est une notation commode pour représenter un sous-ensemble flou linguistique. Il est interprété par Zadeh comme une union sur la collection des parties floues élémentaires ([Zadeh73]). De manière générale, si  $LX = \{L_1, L_2, \dots, L_N\}$ , tout élément  $E$  de  $F(LX)$  peut être noté :

$$E = \sum_{i=1, N} \alpha_i / L_i. \quad (72)$$

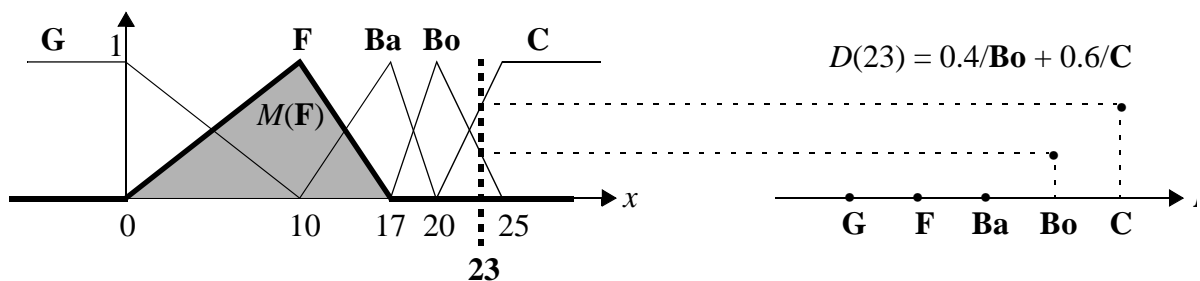


Fig. 32 : Description d'un nombre et signification d'un terme linguistique.

Par souci de simplicité, la notation  $\mu_{M(L)}(x)$  (cf équation (71)) est souvent abrégée en  $\mu_L(x)$  comme cela a d'ailleurs été fait dans tout le premier chapitre. Ceci correspond en fait à assimiler un

mot à sa signification floue (cf note <sup>(1)</sup> extraite de [Zadeh71] où  $x$  représente un terme linguistique).

## II.2 Typologie des systèmes flous

Les systèmes flous de Mamdani, que nous avons utilisés dans le premier chapitre, sont exprimés à partir de règles du type "Si ... alors ...." dans lesquelles prémisses et conclusions sont spécifiées de manière symbolique. Dans le cadre de la commande, l'exploitation d'une telle connaissance nécessite la mise en place d'interfaces numérique/symbolique et symbolique/numérique. Ces dernières sont en effet des passerelles indispensables à l'établissement d'un lien entre la base de règles et le procédé, sur lequel seules des mesures et actions numériques sont envisageables. De manière classique, le fonctionnement interne des systèmes flous est décomposé en trois étapes tel qu'illustré à la figure 33.

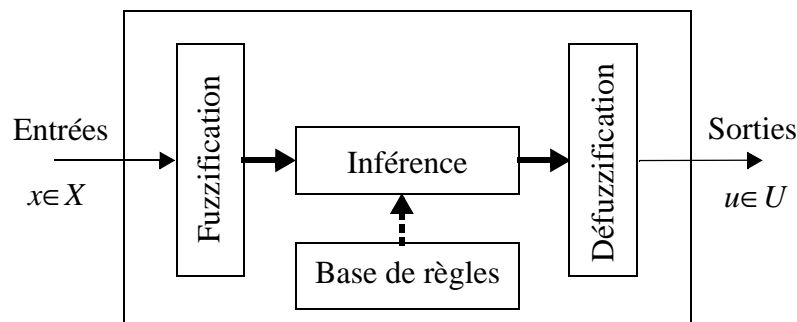


Fig. 33 : Structuration interne d'un système flou.

L'opération de fuzzification consiste à transformer les entrées numériques disponibles en parties floues. Celles-ci alimentent alors le mécanisme d'inférence qui détermine les parties floues relatives aux sorties selon la base de règles fournie. Enfin, la défuzzification convertit ces dernières en sorties numériques. Si cette structuration indique clairement les organes réalisant les tâches de codage et décodage flou, elle ne spécifie aucunement à quel niveau se situent les interfaces numérique/symbolique (NS) et symbolique/numérique (SN). En d'autres termes, les données entrant et sortant du bloc d'inférence sont-elles numériques ou symboliques ? Réponse est donnée à cette question sous la forme d'une typologie des systèmes flous, mettant en évidence différentes implantations possibles ([Foulloy95a]). La figure 34 présente la classification proposée dans laquelle sont différenciés deux types de fuzzification ( $\varphi_1$  et  $\varphi_2$ ), quatre types d'inférence ( $\rho_1$ ,  $\rho_2$ ,  $\rho_3$  et  $\rho_4$ ) et deux types de défuzzification ( $\delta_1$  et  $\delta_2$ ). Chacune de ces fonctionnalités est définie comme une application d'un ensemble source vers un ensemble destination. Ainsi, la fuzzification prend en entrée un élément de  $X$  et fournit en sortie une partie floue de  $X$  ou de  $LX$  suivant qu'elle est numérique ou linguistique, elle sera alors respectivement notée  $\varphi_1$  ou  $\varphi_2$ . À l'inverse, la défuzzification produit un élément de  $U$ , soit à partir d'une partie floue de  $U$ , soit à partir d'une partie floue de  $LU$ , et sera respectivement notée  $\delta_1$  ou  $\delta_2$ . Quant à l'inférence, elle peut prendre quatre formes différentes selon la nature numérique ou linguistique des parties floues d'entrée et de sortie.

(1) "Although  $x$  and  $M(x)$  are entirely different entities, it is expedient to abbreviate  $M(x)$  to  $x$ , relying on the context for the determination of whether  $x$  stands for a term or for its meaning  $M(x)$ . That is what we usually do in every day discourse, because in such discourse it is rarely necessary to differentiate between  $x$  and  $M(x)$ . On the other hand, it is important to differentiate -or at least to understand the difference- between  $x$  and  $M(x)$  in the case of programming languages, machine translation of languages, and other areas in which ambiguity of interpretation can lead to serious errors."

Le choix de trois constituants élémentaires compatibles entre eux conduit finalement à quatre implantations possibles des systèmes flous :

$$\begin{aligned} \text{SF1 : } & u = (\varphi_1 \circ \rho_1 \circ \delta_1)(x), \\ \text{SF2 : } & u = (\varphi_1 \circ \rho_2 \circ \delta_2)(x), \\ \text{SF3 : } & u = (\varphi_2 \circ \rho_3 \circ \delta_1)(x), \\ \text{SF4 : } & u = (\varphi_2 \circ \rho_4 \circ \delta_2)(x). \end{aligned}$$

L'implantation SF1 est dite numérique dans la mesure où les données échangées entre les différents constituants sont des sous-ensembles flous construits sur des référentiels numériques. Selon la même convention, un système SF4 est dit linguistique. Quant aux configurations SF2 et SF3, elles correspondent à des solutions hybrides qualifiées de numérico-symbolique et de symbolico-numérique.

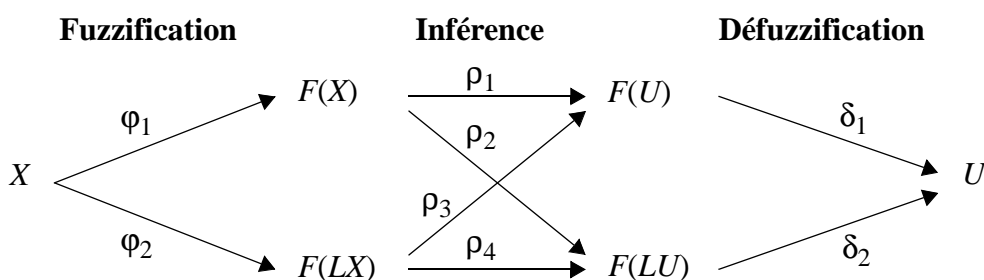


Fig. 34 : Typologie des systèmes flous de Mamdani.

Cette typologie permet d'isoler rapidement les organes qui prennent en charge les conversions NS et SN. Ainsi, dans une implantation SF4, ce sont clairement les blocs de fuzzification et de défuzzification qui réalisent les interfaces NS et SN en codant puis décodant les données manipulées. Dans le cas des systèmes SF1, aucune conversion n'est réalisée au niveau des données et l'inférence est en fait purement numérique. L'interfaçage avec le monde symbolique est alors réalisé par le biais d'une traduction de la base de règles en une relation floue numérique. La figure 35 illustre d'un point de vue conceptuel ces différences de principe sur les cas typiques que sont les SF1 et SF4.

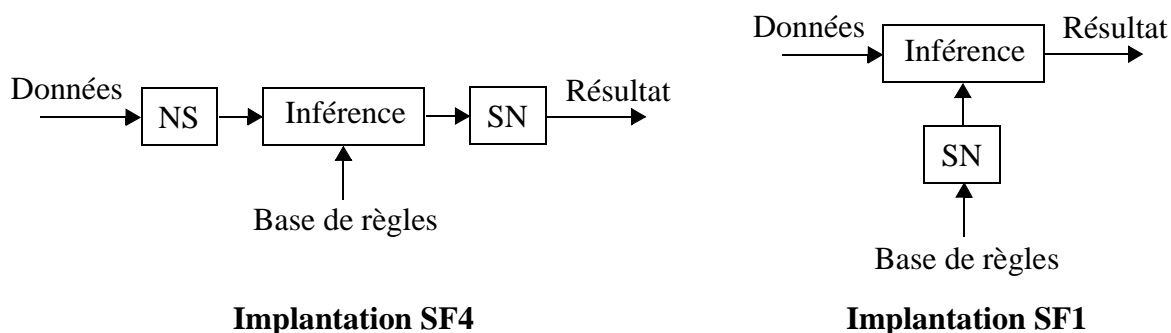


Fig. 35 : Différents principes d'implantation.

Il est montré dans [Foulloy95b] que les systèmes classiques de Mamdani peuvent être réalisés de façon exacte dans chacun des formalismes proposés. Se pose alors la question de l'utilité de distinguer les différentes implantations possibles si elles sont finalement toutes équivalentes. Trois éléments de réponse peuvent être énoncés sous la forme de simples états de fait :

*L'implantation SF1 des systèmes flous de Mamdani n'exploite qu'une forme restreinte de la fuzzification numérique dans laquelle l'entrée, considérée précise, est transformée en un singleton.*

L'implantation SF1 des systèmes flous de Mamdani est basée sur une interprétation spécifique des règles "Si ... Alors ..." puisqu'elle repose implicitement sur une vision conjonctive de ces dernières .

L'implantation SF4 des systèmes flous de Mamdani utilise une version approuvée de l'inférence  $\rho_4$  dans la mesure où le graphe de la relation liant les symboles d'entrée aux symboles de sortie est net.

Il ressort de ces constatations que la typologie offre un cadre beaucoup plus large au développement des systèmes flous que ne le permet le formalisme de base associé aux systèmes de Mamdani. Notamment, l'implantation SF1 inclut de façon naturelle le traitement des entrées imprécises ainsi que le cas des règles implicatives. Quant à l'implantation SF4, elle permet de mener à bien l'inférence directement au niveau symbolique et autorise ainsi l'exploitation de règles linguistiques pondérées. Cette dernière solution s'inscrit dans la ligne directrice de Zadeh résumée sous la formule "Computing with Words". Présentant l'intérêt d'une réelle simplicité d'inférence, cette approche purement symbolique guide les travaux décrits dans la suite de ce chapitre. Dans un premier temps, les systèmes SF4 sont donc détaillés de façon à pouvoir ensuite étudier leur exploitation à la fois dans le cadre du raisonnement approximatif et en tant que systèmes de Sugeno. Dans un second temps, leur intégration dans des structures de commande est abordée dans l'optique de faciliter, mais aussi de structurer, le codage de l'expertise disponible.

### II.3 Implantation SF4

Les SF4 sont ici présentés pour deux entrées  $x$  et  $y$  et une sortie  $z$ . Les règles considérées sont identiques à celles exploitées dans le cadre des systèmes de Mamdani au détail près qu'un poids est maintenant associé à chaque règle. Celles-ci se présentent alors sous la forme :

$$\text{Si } x \text{ est } A_i \text{ et } y \text{ est } B_j \text{ alors } z \text{ est } C_k \text{ avec le poids } \alpha_{ijk}, \quad (73)$$

avec  $LX = \{A_i\}_{i=1, \dots, I}$ ,  $LY = \{B_j\}_{j=1, \dots, J}$  et  $LZ = \{C_k\}_{k=1, \dots, K}$ .

Chaque règle, indicée par le triplet  $(i, j, k)$ , est vue comme une relation liant les mots  $A_i$  et  $B_j$ , associés aux entrées, au mot  $C_k$  décrivant la sortie. Le poids  $\alpha_{ijk} \in [0, 1]$  définit la force de la relation entre  $A_i$ ,  $B_j$  et  $C_k$ , c'est-à-dire :

$$\mu_R(A_i, B_j, C_k) = \alpha_{ijk}. \quad (74)$$

Les règles de type Mamdani représentent un cas particulier de règles pondérées, chaque règle se voyant affecter un poids de 1. On peut par ailleurs remarquer qu'il est possible que plusieurs règles aient la même prémisse mais une conclusion différente. Dans ce cas, elles pourront être regroupées dans une règle compacte unique de façon à simplifier l'écriture :

$$\text{Si } x \text{ est } A_i \text{ et } y \text{ est } B_j \text{ alors } z \text{ est } \sum_{k=1, K} \alpha_{ijk} / C_k, \quad (75)$$

avec  $\alpha_{ijk} = 0$  lorsque la règle correspondante n'existe pas.

Supposons maintenant que le couple  $(x_0, y_0)$  soit présenté au système flou. La sortie  $z$  est calculée selon le mécanisme illustré à la figure 36.

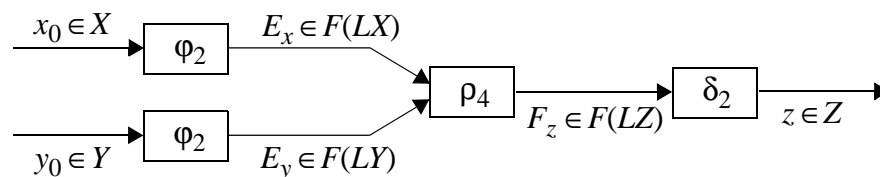


Fig. 36 : Implantation d'un SF4 à deux entrées.



L'étape de fuzzification symbolique est appliquée indépendamment sur les deux variables d'entrée. Elle délivre alors deux parties floues linguistiques respectivement notées  $E_x$  et  $E_y$ . Une façon simple de réaliser cette tâche consiste à utiliser la description floue (cf paragraphe II.1), soit :

$$E_x = \varphi_2(x_0) = D(x_0) \quad \text{et} \quad E_y = \varphi_2(y_0) = D(y_0). \quad (76)$$

L'inférence  $\rho_4$  détermine ensuite l'image de  $E_x \times E_y$  par la relation  $R$  construite à partir des règles. Ce calcul est mis en oeuvre par application directe de la règle compositionnelle d'inférence au niveau linguistique :

$$\forall C_k \in LZ \quad \mu_{F_z}(C_k) = \max_{A_i \in LX, B_j \in LY} \min(\mu_{E_x \times E_y}(A_i, B_j), \mu_R(A_i, B_j, C_k)). \quad (77)$$

Si l'on utilise la t-norme min pour le produit cartésien, l'équation (77) devient :

$$\begin{aligned} \forall C_k \in LZ \quad \mu_{F_z}(C_k) &= \max_{A_i \in LX, B_j \in LY} \min(\mu_{E_x}(A_i), \mu_{E_y}(B_j), \alpha_{ijk}) \\ &= \max_{A_i \in LX, B_j \in LY} \min(\mu_{D(x_0)}(A_i), \mu_{D(y_0)}(B_j), \alpha_{ijk}) \\ &= \max_{A_i \in LX, B_j \in LY} \min(\mu_{A_i}(x_0), \mu_{B_j}(y_0), \alpha_{ijk}). \end{aligned} \quad (78)$$

De manière plus générale, l'inférence  $\rho_4$  peut s'exprimer sous la forme :

$$\forall C_k \in LZ \quad \mu_{F_z}(C_k) = \perp_{A_i \in LX, B_j \in LY} \top_1(\top_2(\mu_{A_i}(x_0), \mu_{B_j}(y_0)), \alpha_{ijk}), \quad (79)$$

où  $\perp$  représente une t-conorme et  $\top_1, \top_2$  des t-normes implantant respectivement les opérations de combinaison et de produit cartésien. Si l'on utilise le formalisme associé aux équations relationnelles floues, l'équation (79) se réécrit sous la forme compacte :  $F_z = (E_x \times E_y) \bullet R$ , où  $\bullet$  représente un opérateur de composition  $\perp / \top_1$ . Il est à noter que si les premières approches relationnelles utilisées en commande ([Tong78]) sont basées sur des relations floues définies sur des univers numériques discrétisés, des travaux plus récents ([Pedrycz96]) exploitent des relations linguistiques tel que préconisé ici.

Enfin, la défuzzification calcule une valeur numérique  $z$  à partir du sous-ensemble flou inféré. Diverses méthodes peuvent être utilisées, il est notamment possible de réaliser une défuzzification symbolique en transformant d'abord  $F_z$  en une partie floue numérique de façon à ensuite appliquer une méthode de défuzzification numérique classique. Dans toute la suite, la mise en oeuvre de  $\delta_2$  sera faite par la méthode des hauteurs dénotée  $H$ , c'est-à-dire :

$$z = \delta_2(F_z) = H(F_z) = \frac{\sum_{k=1}^K \mu_{F_z}(C_k) \cdot z_k}{\sum_{k=1}^K \mu_{F_z}(C_k)}, \quad (80)$$

avec  $z_k$  valeur modale associée au symbole  $C_k$ .

En résumé, l'implantation SF4 consiste à coder l'information numérique sous forme symbolique, puis à réaliser le traitement dans ce mode de représentation, et enfin à décoder le résultat, c'est-à-dire à le retranscrire en nombre. La construction des interfaces NS et SN est primordiale dans la mesure où ce sont elles qui fixent la sémantique du système mais aussi induisent une éventuelle perte d'informations liée au transcodage. Dans ce contexte, [deOliveira95] a introduit le concept d'optimalité des interfaces. Mathématiquement, une interface numérique/symbolique  $\varphi_2$  est dite optimale si elle est inversible, c'est-à-dire s'il existe une interface symbolique/numérique  $\bar{\delta}_2$  garantissant l'égalité :

$$\varphi_2 \circ \bar{\delta}_2 = \text{Identité}. \quad (81)$$

La contrainte (81) traduit le fait que l'opération de codage linguistique est réversible, autrement dit qu'il existe un décodeur permettant de reconstruire la donnée initiale sans perte d'informa-

tions. L'optimalité d'une interface symbolique/numérique  $\delta_2$  est définie de manière similaire et requiert donc l'existence d'une interface  $\bar{\varphi}_2$  telle que :

$$\bar{\varphi}_2 \circ \delta_2 = \text{Identité.} \quad (82)$$

La figure 37 illustre la notion d'interfaces optimales. Reste à souligner que les interfaces inverses  $\delta_2$  et  $\bar{\varphi}_2$ , introduites dans les équations (81) et (82), ne sont pas exploitées dans le système flou implanté, seule leur existence est requise. Les fonctions  $\varphi_2$  et  $\delta_2$  utilisées à la figure 36 ne peuvent en aucun cas être inverses puisqu'elles ne concernent pas les mêmes référentiels. Par contre, leurs implantations sur un même univers de discours via la description ( $D$ ) et la méthode des hauteurs ( $H$ ) sont inverses sous l'hypothèse d'un partitionnement triangulaire strict. Ce résultat, démontré dans [deOliveira99], sera exploité ultérieurement. Il permet en outre de valider les choix d'interfaces et de partitionnement utilisés dans toute la suite de ce chapitre.

Les mécanismes de calcul associés aux systèmes SF4 étant introduits, il est maintenant important de les positionner par rapport aux outils flous classiquement exploités. Il semble notamment intéressant d'étudier les deux points suivants :

*Est-il toujours possible de remplacer une inférence numérique  $\rho_1$  par une inférence symbolique  $\rho_4$ , plus facile à mettre en oeuvre ?*

*Un contrôleur de Sugeno est-il implantable en tant que SF4 ?*

Les paragraphes qui suivent tentent d'apporter des éléments de réponse à ces deux interrogations.

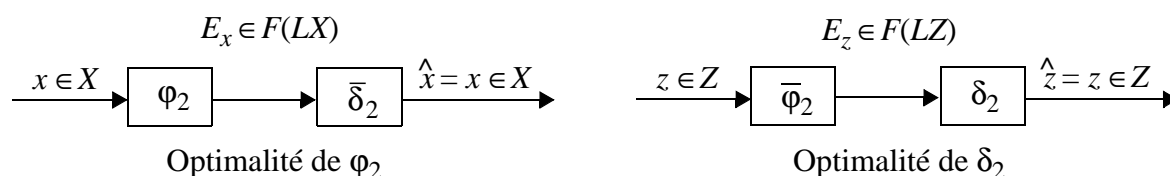


Fig. 37 : Optimalité des interfaces numérique/symbolique et symbolique/numérique.

### III Traitement d'entrées imprécises

La première question concernant l'implantation d'une inférence numérique par le biais d'une inférence symbolique, prend toute sa dimension lorsque des entrées numériques quelconques, et donc éventuellement imprécises, sont considérées. Sans perte de généralité, elle peut donc être reformulée de la manière suivante :

*Comment réaliser une inférence symbolique lorsque l'entrée disponible est imprécise ?*

Si cette interrogation revêt un caractère théorique intéressant, elle n'en est pas moins primordiale d'un point de vue pratique. En effet, l'inférence est généralement implantée selon le principe de combinaison/projection et nécessite donc l'évaluation d'un "sup" sur l'univers de discours d'entrée. Lorsque celui-ci est continu, cas fréquemment rencontré pour un univers numérique associé à un problème de commande, il est nécessaire de le discrétiser de façon à se ramener au calcul d'un "max". Il est alors clair que l'obtention d'un résultat approché satisfaisant requiert l'utilisation d'un pas de discrétisation suffisamment faible et conduit donc à un temps de calcul élevé. Une implantation symbolique de l'inférence permettrait de réduire significativement ce dernier dans la mesure où le "sup" est naturellement restreint à un "max" sur l'ensemble des termes linguistiques de cardinalité généralement faible.

L'étude réalisée, objet d'une collaboration avec D. Dubois et H. Prade de l'IRIT (Toulouse), a été abordée dans le cadre du raisonnement approximatif où l'on retrouve une problématique similaire. Dans ce contexte, la connaissance est en effet classiquement représentée par une distribution de possibilité, l'inférence étant alors appréhendée au niveau numérique. Peut-on prétendre obtenir des résultats équivalents en raisonnant au niveau symbolique par le biais d'une inférence  $\rho_4$  ? La figure 38 illustre l'objectif des travaux développés, à savoir l'élaboration du chemin liant  $E$  à  $F'$  et la comparaison de  $F'$  à  $F$ .

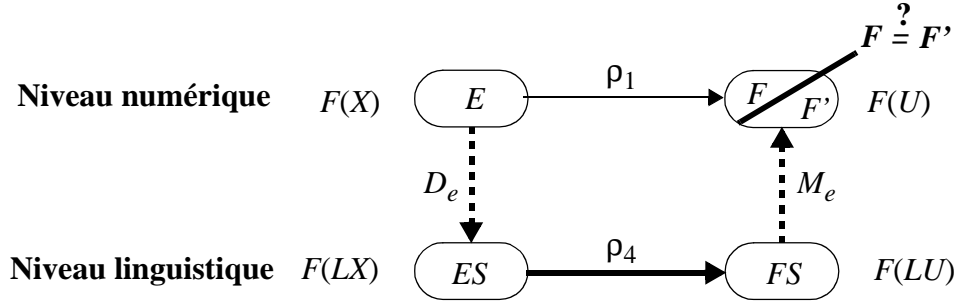


Fig. 38 : Raisonnement au niveau symbolique.

La première étape réside dans l'obtention d'une représentation linguistique de l'entrée numérique imprécise. Autrement dit, avant de pouvoir mener à bien l'inférence symbolique, il est nécessaire de décrire l'entrée  $E$  à l'aide des mots de  $LX$ , c'est-à-dire de construire une partie floue  $ES$  de  $F(LX)$ . Il est clair que la description floue  $D$  introduite dans le paragraphe II.1 (cf équation (71)) ne s'applique qu'aux entrées numériques précises et doit donc être généralisée pour pouvoir être utilisée dans le cas présent. Une solution immédiate consiste à calculer l'image de  $E$  par la relation  $RX$  liant les nombres  $x$  de  $X$  aux mots  $L$  de  $LX$  (cf équation (71)). Par application du principe de combinaison/projection, on obtient alors la description étendue  $D_e$  définie par :

$$\forall L \in LX \quad \mu_{D_e(E)}(L) = \sup_{x \in X} \min(\mu_E(x), \mu_{RX}(x, L)) = \sup_{x \in X} \min(\mu_E(x), \mu_{M(L)}(x)). \quad (83)$$

L'utilisation de la description étendue  $D_e$  permet de construire une partie floue symbolique  $ES$  approximant linguistiquement l'entrée numérique floue  $E$ . Cette dernière alimente le processus de raisonnement symbolique implanté via une inférence  $\rho_4$  de façon à générer une partie floue linguistique  $FS$ , élément de  $F(LU)$ . Le retour au monde numérique nécessite maintenant d'implanter une application permettant de passer de  $F(LU)$  à  $F(U)$ , ainsi qu'illustré à la figure 38. D'un point de vue conceptuel, cette opération consiste à reconstruire la signification de la partie floue symbolique  $FS$ . Une solution à ce problème réside à nouveau dans le calcul de l'image d'une partie floue ( $FS$ ) par une relation ( $RU$  liant cette fois les nombres  $u$  de  $U$  aux mots  $L'$  de  $LU$ ). On obtient ainsi une extension de la signification d'un mot que l'on représentera par une application  $M_e$ , appelée signification étendue, et implantée selon l'équation suivante :

$$\mu_{M_e(FS)}(u) = \max_{j=1, N} \min(\mu_{FS}(L'_j), \mu_{RU}(u, L'_j)) = \max_{j=1, N} \min(\mu_{FS}(L'_j), \mu_{M(L'_j)}(u)) \quad (84)$$

Nous avons montré dans [Dubois97] que lorsque l'inférence numérique  $\rho_1$  exploite des règles conjonctives basées sur les opérateurs min/max, elle peut être remplacée par une inférence symbolique  $\rho_4$  à laquelle sont associées les interfaces  $D_e$  et  $M_e$ , autrement dit :

$$F = \rho_1(E) = D_e \circ \rho_4 \circ M_e(E) = M_e(\rho_4(D_e(E))) = F'. \quad (85)$$

Par contre, lorsque la comparaison est effectuée au niveau symbolique entre  $FS$  et  $FS'$  (cf figure 39), il apparaît que l'approche linguistique est plus précise au sens de l'inclusion que son analogue numérique, c'est-à-dire :

$$FS \subseteq FS' \quad \text{ou encore} \quad \rho_4(D_e^{LX}(E)) \subseteq D_e^{LU}(\rho_1(E)). \quad (86)$$

Les “exposants”  $LX$  et  $LU$ , couplés à la notation  $D_e$ , représentent l’ensemble des mots utilisés pour faire la description et permettent donc de différencier les descriptions associées aux univers de discours  $X$  et  $U$ .

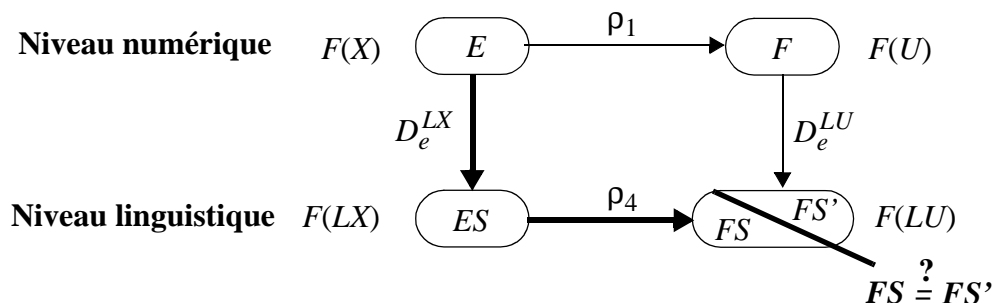


Fig. 39 : Comparaison des résultats au niveau symbolique.

La perte de l’égalité entre  $FS$  et  $FS'$  révèle que les interfaces  $D_e^{LU}$  et  $M_e^U$ , définies respectivement sur  $U$  et  $LU$ , ne sont pas inverses (cf équation (81)) et soulève le problème du chaînage de règles ainsi qu’illustré à la figure 40. En effet, lorsque plusieurs niveaux de raisonnement sont considérés, nous avons montré dans [Dubois99] que les résultats obtenus au niveau numérique sont, dans la plupart des cas, différents. En d’autres termes, l’égalité (85) n’est pas préservée, ce qui se traduit par le fait qu’en général  $G \neq M_e^Z(GS)$ , les notations  $G$  et  $GS$  étant définies à la figure 40. Par contre, le fait qu’une implantation symbolique conduise à des résultats plus précis qu’une implantation numérique (cf inégalité (86)) reste valide lorsqu’un chaînage de règles est mis en oeuvre, c’est-à-dire  $GS \subseteq D_e^{LZ}(G)$ . Si ces constatations peuvent s’avérer gênantes dans le cadre du raisonnement approximatif, elles sont peu préoccupantes en commande où une interface finale de défuzzification est de toute manière amenée à produire un résultat précis. Dans ce contexte, une réflexion plus globale reste cependant nécessaire pour pouvoir réellement se prononcer sur l’utilité d’une propagation de l’incertitude.

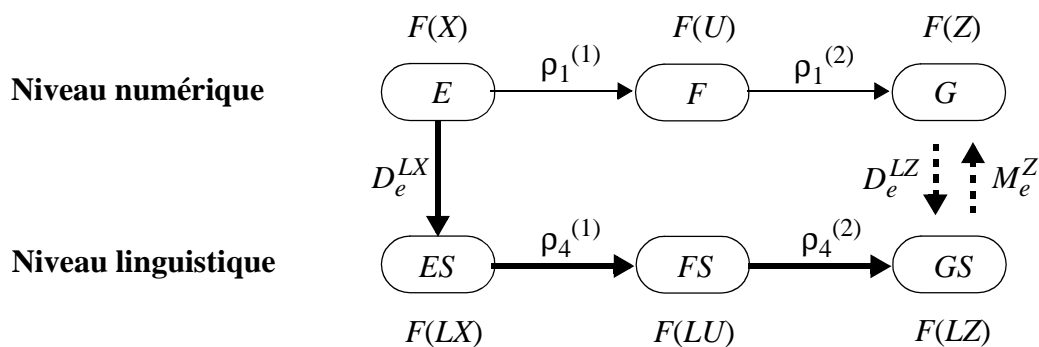


Fig. 40 : Chaînage de règles.

D’un point de vue pratique, il est important de noter que le problème lié à l’évaluation du “sup” lors de l’inférence numérique est effectivement éliminé de l’inférence symbolique. La difficulté n’a pas pour autant disparu puisqu’elle a en fait été déportée vers la description étendue (83). L’intérêt d’une telle démarche réside dans la possibilité de distribuer les traitements dans divers organes ainsi que préconisé dans un paragraphe à venir, lié à l’exploitation de capteurs flous dans une structure de commande (cf paragraphe VI.2).

L’ensemble des résultats énoncés jusqu’à présent est limité aux relations numériques floues conjonctives. Si cette restriction est usuelle en contrôle, il n’en demeure pas moins que l’approche implicative est couramment exploitée dans les disciplines liées au raisonnement. Nous avons mené divers travaux sur le sujet ([Dubois97], [Dubois99]) mettant en évidence la

difficulté d'une implantation symbolique. Une première piste consiste à exploiter des interfaces mieux adaptées au cas implicatif, notamment en ce qui concerne la description. La description étendue  $D_e$  définie précédemment (cf équation (84)) n'est autre que la possibilité de l'événement flou  $M(L)$  connaissant la distribution de possibilité  $E$ . On peut alors imaginer une approche similaire basée sur le concept de nécessité et conduisant à la définition de la description étendue  $D_e^-$  suivante :

$$D_e^- = N(M(L)) = 1 - \Pi(\overline{M(L)}), \quad (87)$$

où la barre horizontale correspond à spécifier l'évènement contraire.

D'un point de vue des degrés d'appartenance, l'équation (87) se traduit par :

$$\forall L \in LX \quad \mu_{D_e^-(E)}(L) = 1 - \sup_{x \in X} \min(\mu_E(x), 1 - \mu_{M(L)}(x)) = \inf_{x \in X} \max(1 - \mu_E(x), \mu_{M(L)}(x)). \quad (88)$$

Quand  $E$  est une partie floue numérique normalisée, l'inégalité suivante est garantie :

$$\forall L \in LX \quad \mu_{D_e^-(E)}(L) \leq \mu_{D_e(E)}(L) = \mu_{D_e^+(E)}(L), \quad (89)$$

d'où l'introduction des notations “-” et “+” permettant de différencier les deux types de description. D'un point de vue ensembliste, on peut également noter que les descriptions  $D_e^-(E)$  et  $D_e^+(E)$  correspondent respectivement aux bornes supérieure et inférieure de la famille  $(D(x))_{x \in E}$  ([Foulloy95a]).

Si l'entrée  $E$  est précise ( $E = \{x_0\}$ ), les descriptions étendues supérieure et inférieure coïncident avec la description du nombre considéré, c'est-à-dire :  $D_e^-(\{x_0\}) = D(x_0) = D_e^+(\{x_0\})$ . Cette constatation conduit à la conclusion qu'il est vain d'espérer obtenir une implantation symbolique implicative par un simple remplacement de la description  $D_e^+$  par la description  $D_e^-$ .

En fait, la difficulté de l'implantation implicative réside dans le fait que l'inférence ne peut pas être menée règle par règle. Cet obstacle nous a amenés à proposer une seconde piste visant cette fois à transformer la base de règles. A cet effet, nous proposons dans [Dubois99] d'exploiter la forme normale disjonctive des règles originales de façon à les convertir en règles conjonctives (cf figure 41). Si cette stratégie permet de se ramener au cas traité précédemment, c'est au prix d'une augmentation importante du nombre de règles et d'une redéfinition des vocabulaires utilisés. D'un point de vue pratique, le calcul de la description supérieure de l'entrée  $E$  sur le vocabulaire étendu nécessite d'évaluer  $2^n$  mesures de possibilité, soit  $2^n$  suprema,  $n$  représentant le cardinal de  $LX$ . Nous avons montré qu'il est cependant possible d'en obtenir des approximations supérieure et inférieure à partir des descriptions supérieure et inférieure de  $E$  sur le vocabulaire initial. On peut ainsi déterminer un intervalle d'appartenance du résultat final en calculant uniquement  $D_e^+(E)$  et  $D_e^-(E)$ , ce qui ramène à  $2n$  le nombre de suprema à évaluer.

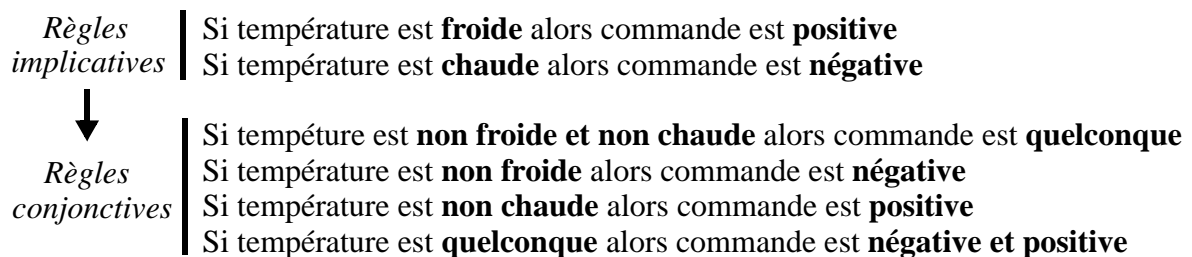


Fig. 41 : Transformation de règles implicatives en règles conjonctives.

Si l'on peut imaginer une exploitation future de ces résultats en commande, notamment de l'approche implicative, et espérer apporter des réponses pertinentes quant à la gestion des incertitu-

des, l'inférence symbolique  $\rho_4$  semble être à l'opposé des travaux actuellement développés en contrôle flou. Le paragraphe qui suit tente de briser cette impression dans la mesure où il met en évidence la possibilité d'implanter un système flou de Sugeno à conclusions constantes via une inférence linguistique. Cette approche représente un premier pas vers une unification des stratégies numériques, majoritairement exploitées dans les travaux récents en contrôle flou, et des stratégies expertes à l'origine des premiers succès industriels. Le lien établi entre les aspects numérique et linguistique de la commande floue représente également une clé de lecture de ce manuscrit puisqu'il rend possible une utilisation conjointe des outils et méthodes présentées dans les deux chapitres de ce document.

## IV Lien avec l'approche numérique TSK

### IV.1 TSK $\leftrightarrow$ SF4 avec règles pondérées

L'objectif de ce paragraphe est d'établir une méthodologie de conversion d'un système TSK à conclusion constante, tel qu'utilisé dans le premier chapitre, en un système SF4. La première solution consiste tout simplement à associer un symbole à chaque valeur numérique possible des conclusions de règles. L'utilisation d'une fonction d'appartenance de type singleton pour chacun des termes ainsi introduits permet d'obtenir une équivalence exacte entre les deux implantations. Cette façon de faire présente néanmoins l'inconvénient majeur de nécessiter la définition d'un grand nombre de symboles dans la mesure où chaque règle peut conduire à la création d'un terme spécifique. Hormis des cas particuliers, elle n'a donc aucun intérêt en ce qui concerne l'amélioration de la lisibilité du système original.

Une seconde solution, préférable dans l'optique d'une réelle interprétation linguistique des règles, est proposée ci-dessous. Cette dernière exploite la possibilité d'affecter des poids aux différentes règles d'un système SF4. Ainsi, l'équivalence entre un système flou de type TSK à conclusions constantes et un SF4 linguistique exploitant des règles pondérées est énoncée dans la proposition suivante :

#### **Proposition 1**

Sous l'hypothèse d'une partition triangulaire stricte des univers d'entrée et de sortie, le système flou de Sugeno décrit par les règles :

$$\text{Si } x_1 \text{ est } A_1^{i_1} \text{ et } x_2 \text{ est } A_2^{i_2} \text{ et } \dots \text{ et } x_n \text{ est } A_n^{i_n} \text{ alors } u = c_{i_1, i_2, \dots, i_n}, \quad (90)$$

est strictement équivalent au système SF4 implanté avec la t-norme "produit" et la t-conorme "somme bornée" dont les règles s'écrivent :

$$\text{Si } x_1 \text{ est } A_1^{i_1} \text{ et } \dots \text{ et } x_n \text{ est } A_n^{i_n} \text{ alors } u \text{ est } D(c_{i_1, i_2, \dots, i_n}) = \sum_{k=1}^K \alpha_{i_1, i_2, \dots, i_n, k} / C_k. \quad (91)$$

#### **Démonstration 1**

Selon l'équation (79), la sortie linguistique  $F$  engendrée par le système SF4 décrit par les règles (91) s'exprime sous la forme :

$$\forall C_k \in LU \quad \mu_F(C_k) = \bigwedge_{(i_1, i_2, \dots, i_n) \in I} \top (\mu_{A_1^{i_1}}(x_1^0), \dots, \mu_{A_n^{i_n}}(x_n^0), \alpha_{i_1, i_2, \dots, i_n, k}) \quad (92)$$

où  $E = (x_1^0, x_2^0, \dots, x_n^0)$  représente le vecteur d'entrée considéré et  $I$  l'ensemble des  $n$ -uplets indiquant les règles. Pour une partition stricte des univers d'entrée, le choix de la t-norme "produit" et la t-conorme "somme bornée" conduit à :

$$\forall C_k \in LU \quad \mu_F(C_k) = \sum_{(i_1, i_2, \dots, i_n) \in I} \omega_{i_1, i_2, \dots, i_n} \alpha_{i_1, i_2, \dots, i_n, k} \quad (93)$$

où  $\omega_{i_1, i_2, \dots, i_n} = \prod_{j=1, \dots, n} \mu_{A_j^{i_j}}(x_j^0)$  représente le degré d'activation de la règle indiquée  $(i_1, i_2, \dots, i_n)$ .

La défuzzification de  $F$  par la méthode des hauteurs  $H$  (cf équation (80)) aboutit à l'obtention de la sortie numérique  $u_{SF4}$  donnée par :

$$u_{SF4} = H(F) = \frac{\sum_{k=1, \dots, K} \sum_{(i_1, i_2, \dots, i_n) \in I} \omega_{i_1, i_2, \dots, i_n} \alpha_{i_1, i_2, \dots, i_n, k} u_k}{\sum_{k=1, \dots, K} \sum_{(i_1, i_2, \dots, i_n) \in I} \omega_{i_1, i_2, \dots, i_n} \alpha_{i_1, i_2, \dots, i_n, k}} \quad (94)$$

Le partitionnement strict des univers d'entrée et de sortie permet d'assurer que le dénominateur est égal à 1. De plus, sous l'hypothèse d'un partitionnement triangulaire de l'univers de sortie  $U$ , la description  $D$  sur  $LU$  est l'inverse de la méthode des hauteurs  $H$  (cf section II.3). L'application de cette propriété pour les valeurs  $c_{i_1, i_2, \dots, i_n}$  de  $U$  est illustrée à la figure 42. Elle permet de déduire que  $c_{i_1, i_2, \dots, i_n} = \sum_{k=1, \dots, K} \alpha_{i_1, i_2, \dots, i_n, k} u_k$ . Par substitution dans (94), on obtient alors :

$$u_{SF4} = \sum_{(i_1, i_2, \dots, i_n) \in I} \omega_{i_1, i_2, \dots, i_n} c_{i_1, i_2, \dots, i_n} \quad (95)$$

ce qui correspond à la sortie du système de Sugeno initial.

$$c_{i_1, i_2, \dots, i_n} \xrightarrow{D} \sum_{k=1}^K \alpha_{i_1, i_2, \dots, i_n, k} / C_k \xrightarrow{H} \sum_{k=1}^K \alpha_{i_1, i_2, \dots, i_n, k} \cdot u_k = c_{i_1, i_2, \dots, i_n}$$

Fig. 42 : Relation entre  $D$  et  $H$ .

### Exemple

La méthodologie de conversion est ici appliquée au contrôleur flou développé par application du principe des équivalences modales pour la régulation de niveau flottant dans une capacité tampon en raffinerie (cf paragraphe VI.1 du premier chapitre, page 39). La partition de l'univers de sortie a été réalisée selon l'opinion des experts. Elle est basée sur la définition de neuf symboles, notés NTG, NG, NM, NP, Z, PP, PM, PG et PTG. Les fonctions d'appartenance associées à ces termes sont triangulaires, leur recouvrement assurant une partition stricte. Les valeurs modales sont respectivement fixées à -10, -2, -1, -0.25, 0, 0.25, 1, 2, 10. La valeur 10 correspond à la valeur de saturation définie sur la variation de commande (cf équation (41)). La base de règles du Tableau 3 (cf page 41) est ainsi traduite en la base de règles pondérées du Tableau 5. Notons toutefois que celui-ci ne reprend que les règles correspondant à une variation de niveau positive, les autres pouvant être déduites par symétrie.

		→ signe( $\delta L$ ) * $\delta L^2$				
		$B_0$	$B_1$	$B_2$	$B_3$	
$L$	$L_{min}$	$A_0$	$1 / Z$	$0.89 / PP + 0.11 / PM$	$0.44 / PP + 0.56 / PM$	$1 / PM$
	↓	$A_1$	$1 / Z$	$0.74 / PP + 0.26 / PM$	$0.15 / PP + 0.85 / PM$	$0.67 / PM + 0.33 / PG$
	↓	$A_2$	$1 / Z$	$0.44 / PP + 0.56 / PM$	$0.67 / PM + 0.33 / PG$	$1 / PG$
	↓	$A_3$	$1 / Z$	$0.67 / PM + 0.33 / PG$	$0.92 / PG + 0.08 / PTG$	$0.75 / PG + 0.25 / PTG$
	$L_{max}$	$A_4$	$1 / Z$	$1 / PTG$	$1 / PTG$	$1 / PTG$

Tableau 5 : Base de règles du contrôleur SF4.

Le contrôleur SF4 engendré permet de préserver la précision numérique d'origine tout en autorisant une lecture à plusieurs niveaux des règles. En effet, si l'on oublie l'aspect numérique de la commande pour se focaliser sur l'aspect compréhension du mécanisme sous-jacent, la règle de prémisse " $A_0$  et  $B_1$ " (case grisée) permet de conclure que la variation de commande doit être entre PP et PM. Un résumé encore plus condensé peut être obtenu en choisissant le sym-

bole de sortie majoritairement présent dans la conclusion pondérée de chaque règle (termes mis en gras dans le Tableau 5).

D'autres approches de conversion, relativement proches de celle proposée ici, peuvent être trouvées dans la littérature. Ainsi, [Nozaki97] utilise deux bases de règles distinctes pour traduire linguistiquement un système TSK à conclusions constantes. La première correspond à l'utilisation des sorties linguistiques prépondérantes alors que la deuxième exploite les symboles secondaires, c'est-à-dire décrivant la sortie numérique désirée à un degré moindre. Le mécanisme de calcul associé combine les deux sorties produites selon les degrés de certitude associés à chacune d'entre elles. Si le résultat final est strictement identique à celui obtenu à partir d'un système SF4, la présentation à partir de deux bases de règles distinctes a conduit [Cordon00] à en proposer une version modifiée permettant de réduire le nombre de règles exploitées. L'auteur préconise l'exploitation d'une base de règles unique regroupant des règles de Mamdani classique et des règles dites à conclusion double. Ces dernières sont exprimées sous la forme générique "Si ... alors  $u$  est entre  $C_i$  et  $C_{i+1}$ " où  $u$  représente la variable de sortie et  $C_i, C_{i+1}$  deux symboles définis sur l'univers numérique correspondant. Elles sont traitées comme deux règles distinctes dans le processus d'inférence et produisent donc un résultat similaire à la règle pondérée "Si ... alors  $u$  est  $0.5 / C_i + 0.5 / C_{i+1}$ ". Finalement, la base de règles engendrée n'est autre qu'un cas particulier de base de règles pondérées, les poids étant restreints aux valeurs 0, 0.5 et 1. La réduction annoncée du nombre de règles correspond en fait à ne comptabiliser qu'une seule fois les règles dont le poids vaut 1 et non plus 2 fois comme c'est le cas lorsque deux bases de règles distinctes sont considérées.

Les travaux développés dans [Cordon00] sont présentés dans le cadre de l'apprentissage d'un jeu de règles à partir d'exemples numériques. A ce titre, l'auteur souligne que la méthode proposée a l'avantage de ne pas nécessiter une décomposition du problème global en deux étapes, la première consistant à construire un jeu de règles TSK à conclusion constante, la seconde correspondant à sa traduction linguistique. Reste à noter que la méthode d'apprentissage proposée dans le premier chapitre de ce document ne nécessite aucunement le passage explicite par des règles de Sugeno et que seul l'aspect pédagogique de la présentation nous a conduit à une telle démarche. Preuve en est qu'à l'origine, les travaux développés en apprentissage ont été directement introduits dans un contexte linguistique ([Nakoula97a], [Nakoula97b]).

Si la proposition 1 permet de convertir tout système TSK à conclusion constante en système SF4, elle ne permet un traitement de l'opération inverse que dans le cas particulier où les conclusions de règles du système SF4 résultent d'une description floue (cf équation (91)). Or, dans le cas général, les conclusions de règles ne sont pas toujours exprimables en tant que description floue d'une valeur numérique. Par exemple, la partie floue linguistique  $0.1/Z + 0.4/P + 0.5/M$  ne peut provenir d'aucune description floue basée sur une partition normalisée triangulaire stricte. La conversion d'un système SF4 en un système TSK à conclusion constante est donc appréhendée par le biais d'une nouvelle proposition énoncée ci-dessous. Cette dernière reste cependant restreinte aux systèmes SF4 dont les conclusions de règles sont des parties floues linguistiques  $PFL_{i_1, i_2, \dots, i_n}$  s'exprimant sous la forme  $\sum_{k=1, \dots, K} \alpha_{i_1, i_2, \dots, i_n, k} / C_k$  où la somme des  $\alpha_{i_1, i_2, \dots, i_n, k}$  vaut 1. De telles règles seront appelées règles unitaires.

### **Proposition 2**

Sous l'hypothèse d'une partition triangulaire stricte des univers d'entrée et de sortie, le système SF4 implanté avec la t-norme "produit" et la t-conorme "somme bornée" et décrit par les règles unitaires :

$$\text{Si } x_1 \text{ est } A_1^{i_1} \text{ et } \dots \text{ et } x_n \text{ est } A_n^{i_n} \text{ alors } u \text{ est } PFL_{i_1, i_2, \dots, i_n} = \sum_{k=1}^K \alpha_{i_1, i_2, \dots, i_n, k} / C_k, \quad (96)$$



est strictement équivalent au système TSK dont les règles s'écrivent :

$$\text{Si } x_1 \text{ est } A_1^{i_1} \text{ et } x_2 \text{ est } A_2^{i_2} \text{ et } \dots \text{ et } x_n \text{ est } A_n^{i_n} \text{ alors } u = c_{i_1, i_2, \dots, i_n}, \quad (97)$$

$$\text{avec } c_{i_1, i_2, \dots, i_n} = H(\text{PFL}_{i_1, i_2, \dots, i_n}) = \sum_{k=1}^K \alpha_{i_1, i_2, \dots, i_n, k} \cdot u_k.$$

### Démonstration 2

En exprimant la sortie du contrôleur flou TSK décrit par les règles (97), on retrouve immédiatement la sortie du système SF4 d'origine (cf équation (94)).

La figure 43 présente un résumé des mécanismes de conversion introduits via les propositions 1 et 2. Elle illustre notamment le fait que chaque règle peut être traitée indépendamment des autres et que l'utilisation successive des propositions 2 puis 1 permet de réduire toute règle unitaire SF4 en une règle simplifiée du type "Si ... alors u est  $\alpha_k/C_k + (1-\alpha_k)/C_{k+1}$ " où  $C_k$  et  $C_{k+1}$  représentent deux symboles consécutifs.

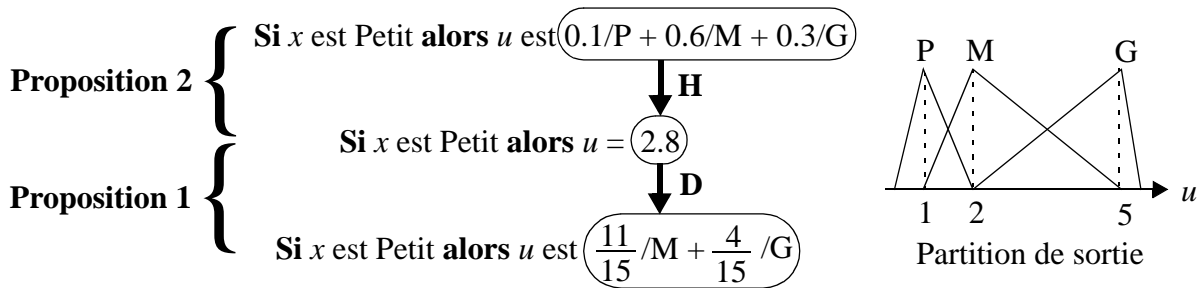


Fig. 43 : Mécanismes de conversion.

## IV.2 Extension du mécanisme de calcul TSK aux règles à conclusions linguistiques floues

Ce paragraphe est dédié au développement d'une nouvelle implantation d'un système SF4 exploitant le mécanisme de calcul des systèmes TSK, c'est-à-dire réalisant une moyenne pondérée des conclusions linguistiques floues des règles activées. La construction des opérateurs nécessaires à cette implantation est abordée selon un principe similaire à celui proposé dans [Herrera00]. La méthode est cependant adaptée de façon à prendre en compte la différence de représentation des informations linguistiques manipulées puisque [Herrera00] préconise de décrire un nombre  $u$  par un couple  $(C_k, d)$  où  $C_k$  correspond au symbole qui décrit le mieux  $u$  et  $d = u - u_k$  à la différence entre  $u$  et la valeur modale de  $C_k$ . L'extension du mécanisme de calcul TSK aux règles à conclusions linguistiques floues est introduite pas à pas et son équivalence avec la formulation initiale des systèmes SF4 est prouvée.

L'expression générique (91) d'une règle unitaire d'un système SF4 peut être réécrite sous la forme :

$$\text{Si } x_1 \text{ est } A_1^{i_1} \text{ et } \dots \text{ et } x_n \text{ est } A_n^{i_n} \text{ alors } u = \text{PFL}_{i_1, i_2, \dots, i_n}, \quad (98)$$

où  $\text{PFL}_{i_1, i_2, \dots, i_n}$  représente une partie floue linguistique, élément de  $F(LU)$ , associée à la règle indiquée  $(i_1, i_2, \dots, i_n)$ . Si l'on fait abstraction de la nature linguistique des conclusions de règles, il est possible d'exploiter le mécanisme de calcul attaché aux systèmes TSK pour déterminer la sortie correspondant au vecteur d'entrée  $(x_1^0, x_2^0, \dots, x_n^0)$  soit, dans le cas d'une partition stricte des univers d'entrée :

$$F = \text{SUG}(x_1^0, x_2^0, \dots, x_n^0) = \bigoplus_{(i_1, i_2, \dots, i_n)} w_{i_1, i_2, \dots, i_n} \odot \text{PFL}_{i_1, i_2, \dots, i_n}, \quad (99)$$

où  $w_{i_1, i_2, \dots, i_n} = \prod_{k=1, \dots, n} \mu_{A_k}^{i_k}(x_k^0)$  représente le degré d'activation de la règle  $(i_1, i_2, \dots, i_n)$ .

Les opérations d'addition et de produit par un réel mises en oeuvre dans (99) concernent des parties floues linguistiques. Les opérateurs correspondants ont donc été encerclés de façon à pouvoir les distinguer de leurs équivalents sur des nombres réels. Il est évident que l'exploitation de (99) nécessite avant tout de définir  $\oplus$  et  $\odot$  pour pouvoir évaluer le résultat  $F$ . Celui-ci n'est autre que la somme pondérée d'un ensemble de parties floues linguistiques. A ce titre, il est également un sous-ensemble flou linguistique et doit donc être défuzzifié pour produire une valeur numérique de commande. Pour garder une cohérence avec les systèmes SF4 présentés précédemment, cette tâche est réalisée via la méthode des hauteurs H (cf équation (80)). L'objectif visé pour la définition des opérateurs  $\oplus$  et  $\odot$  est de maintenir la cohérence avec l'implantation SF4 ce qui se traduit par le souhait que la partie floue  $F$  produite par le mécanisme de calcul (99) soit identique à celle délivrée par le système linguistique SF4 (cf équation (93)).

L'addition de deux parties floues linguistiques engendre une nouvelle partie floue linguistique. L'opérateur  $\oplus$  doit donc être implanté par le biais d'une fonction de  $F(LU) \times F(LU)$  vers  $F(LU)$ . De même, la multiplication d'une partie floue linguistique par un réel (degré  $\in [0, 1]$ ) produit une partie floue linguistique, ce qui signifie que l'opérateur  $\odot$  correspond à une application de  $[0, 1] \times F(LU)$  vers  $F(LU)$ . Dans la mesure où les opérations d'addition et de multiplication ne sont pas définies sur les ensembles de mots, il est impossible d'avoir recours au principe d'extension pour construire  $\oplus$  et  $\odot$ . Nous proposons donc une méthodologie basée sur l'utilisation des interfaces SN et NS que sont la méthode des hauteurs H et la description floue D. L'idée est alors de définir les opérations  $\oplus$  et  $\odot$  en trois étapes : la première convertit les informations linguistiques floues en nombres, la seconde réalise l'opération désirée sur les nombres et la troisième retranscrit le résultat en partie floue linguistique. Cette approche se traduit alors par les équations suivantes :

$$\forall \text{PFL}_1, \text{PFL}_2 \in F(LU) \quad \text{PFL}_1 \oplus \text{PFL}_2 = D( H(\text{PFL}_1) + H(\text{PFL}_2) ), \quad (100)$$

$$\forall \text{PFL} \in F(LU), \forall w \in [0, 1], \quad w \odot \text{PFL} = D( w \cdot H(\text{PFL}) ). \quad (101)$$

Illustrons la combinaison des deux opérations, telle que réalisée dans l'équation (99), pour un système monovarié à deux règles :

$$\begin{aligned} F = \text{SUG}(x^0) &= \left( \sum_{i=1}^2 w_i \odot \text{PFL}_i \right) \oplus (w_2 \odot \text{PFL}_2) \\ &= D( H( D(w_1 \cdot H(\text{PFL}_1))) + H( D(w_2 \cdot H(\text{PFL}_2))) ), \end{aligned} \quad (102)$$

qui, grâce à la relation entre les interfaces H et D, se simplifie en :

$$F = \text{SUG}(x^0) = D( w_1 \cdot H(\text{PFL}_1) + w_2 \cdot H(\text{PFL}_2) ) = D( \sum_{i=1}^2 w_i \cdot H(\text{PFL}_i) ). \quad (103)$$

En généralisant et en notant  $c_{i_1, i_2, \dots, i_n} = H(\text{PFL}_{i_1, i_2, \dots, i_n})$ , l'équation (99) se réécrit donc :

$$F = \text{SUG}(x_1^0, x_2^0, \dots, x_n^0) = D( \sum_{(i_1, i_2, \dots, i_n)} w_{i_1, i_2, \dots, i_n} \cdot c_{i_1, i_2, \dots, i_n} ). \quad (104)$$

La défuzzification de la partie floue  $F$  par la méthode des hauteurs H, inverse de la description D, fournit le résultat qu'aurait produit le contrôleur flou TSK à conclusions constantes  $c_{i_1, i_2, \dots, i_n}$ . Par application de la proposition 2 du paragraphe précédent, on peut finalement déduire que l'utilisation du mécanisme de calcul TSK, étendu au cas des règles à conclusions linguistiques floues, est équivalente à l'exploitation du système SF4 dont les règles sont données par :

$$\text{Si } x_1 \text{ est } A_1^{i_1} \text{ et } \dots \text{ et } x_n \text{ est } A_n^{i_n} \text{ alors } u \text{ est } D( H(\text{PFL}_{i_1, i_2, \dots, i_n}) ). \quad (105)$$

Bien que la relation  $D( H(\text{PFL}_{i_1, i_2, \dots, i_n}) ) = \text{PFL}_{i_1, i_2, \dots, i_n}$  ne soit pas vérifiée, l'équivalence avec le système SF4 d'origine, défini par les règles (98), est garantie par application successive des

propositions 2 et 1 (cf figure 43).

Si l'équivalence exacte est prouvée d'un point de vue théorique, il n'en demeure pas moins que l'utilisation directe des opérateurs  $\oplus$  et  $\ominus$  (cf équations (100) et (101)) nécessite que tout nombre, intervenant dans une phase intermédiaire de calcul, puisse être décrit par les symboles définis. Ceci requiert notamment que la partition de l'univers de sortie permette de décrire le nombre 0. Cette contrainte est d'ailleurs posée comme une propriété sémantique importante des partitions dans [deOliveira95].

En conclusion, l'implantation TSK d'un système SF4, introduite ici comme extension naturelle de nos travaux, peut répondre, par un choix raisonné des opérateurs  $\oplus$  et  $\ominus$ , à des problèmes de fusion d'informations linguistiques pour lesquelles il existe une représentation numérique sous-jacente.

### IV.3 Approximation linguistique ou réduction d'une base de règles

Les deux propositions énoncées dans le paragraphe IV.1 permettent de transformer un système TSK à conclusions constantes en un système linguistique SF4. Une suite logique à ces résultats consiste maintenant à étendre la méthodologie proposée à des systèmes flous quelconques. Dans la mesure où la contrainte de transformation exacte est relaxée, on aboutit à une problématique d'approximation linguistique, ou encore de réduction, d'une base de règles.

D'un point de vue conceptuel, la réduction d'une base de règles consiste tout simplement à diminuer le nombre de règles qu'elle contient sans trop dégrader le transfert entrée/sortie qu'elle représente. Le principe général est alors d'éliminer les règles inutiles (redondantes ou sans importance) pour ne garder que les règles pertinentes. D'un point de vue pratique, la suppression de règles s'accompagne souvent d'une modification des règles conservées de façon à ce qu'elles puissent intégrer correctement l'information éliminée. Cette éventuelle transformation des règles initiales peut aller d'un simple ajustement de paramètres jusqu'à un changement profond dans la nature des règles considérées.

Avant d'exploiter une quelconque méthode de réduction de base de règles, il est essentiel de clairement définir l'objectif final recherché. On peut ainsi distinguer deux buts distincts, voire antagonistes. Le premier réside dans une amélioration de l'implantation finale du système flou (temps de calcul, mémoire). Quant au second, il est avant tout d'augmenter la lisibilité de la base de règles. Dans ce cas, la réduction d'une base de règles peut être vue comme une approximation linguistique de cette dernière ([Sugeno93]). C'est l'objectif premier de la méthodologie proposée ici ([Galichet97]). Elle repose en fait sur l'utilisation de la méthode d'apprentissage présentée dans le premier chapitre (cf paragraphe VII.2) qui produit une base de règles réduite sous la forme d'un ensemble de règles TSK à conclusions constantes. L'exploitation de partitions triangulaires strictes permet ensuite une conversion systématique en un système SF4 (cf paragraphe IV.1). Le résultat final est alors une base de règles compréhensibles, à pouvoir explicatif élevé dans la mesure où le nombre de règles est limité.

La stratégie que nous avons développée est ainsi basée sur un scénario à trois étapes tel qu'illustré à la Fig. 44. Dans un premier temps, le comportement numérique externe du système flou initial que l'on cherche à réduire est résumé par un ensemble de données entrées/sortie. Les points ainsi générés alimentent le processus de construction d'une nouvelle base de règles. De par sa nature incrémentale, l'apprentissage mis en oeuvre a l'avantage de déterminer le nombre de règles nécessaires à l'obtention d'une erreur d'approximation fixée et produit donc un jeu de règles de taille minimale. La transformation en règles linguistiques est finalement réalisée par application de la proposition 1 du paragraphe IV.1.

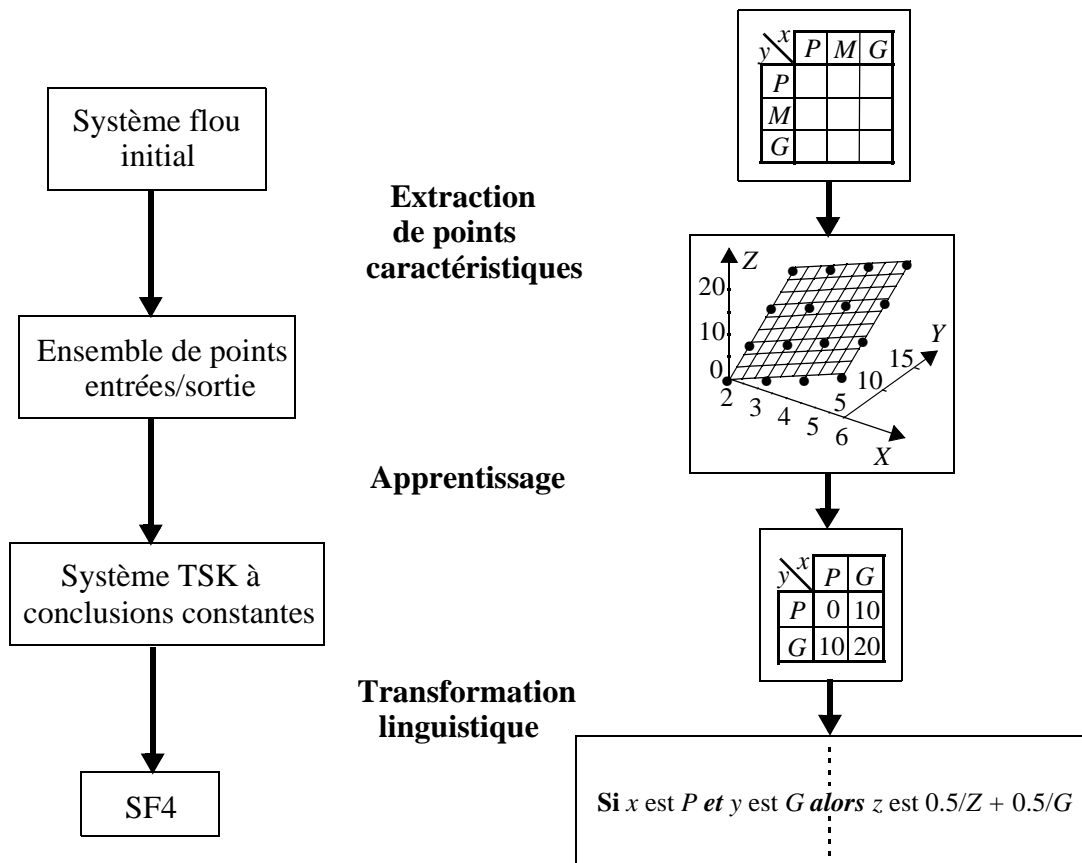


Fig. 44 : Stratégie de réduction d'une base de règles.

L'aspect linguistique de cette approche la différencie de celles basées sur l'exploitation d'une transformation orthogonale tel que préconisé dans [Yam99] et [Yen99] qui présentent deux utilisations possibles d'une décomposition en valeurs singulières (DVS). Ainsi, [Yam99] effectue la DVS de la matrice formée des conclusions de règles et élimine les règles correspondant à des valeurs singulières faibles. Pour maintenir une approximation correcte, les fonctions d'appartenance associées aux entrées sont modifiées sans souci de la lisibilité du système final. Dans [Yen99], la DVS est appliquée à la matrice formée des degrés d'activation des différentes règles pour un ensemble de points d'apprentissage donné. L'élimination des règles correspondant à des valeurs singulières faibles est répercutée par un ajustement des conclusions de règles dont la lisibilité n'est aucunement améliorée.

L'approche que nous proposons se distingue également des méthodologies basées sur une analyse de la similarité entre ensembles flous ou entre règles ([Chao96], [Babuska96]). En effet, la combinaison d'objets jugés similaires réduit le nombre de règles mais n'en améliore aucunement la lisibilité.

Comme premier exemple illustratif, nous considérons des règles initiales purement symboliques afin de mettre en évidence la réduction de leur nombre sans dégradation de leur lisibilité. La figure 45 présente la base de règles initiales ainsi que la forme réduite obtenue après apprentissage à partir des points modaux. En effet, ces derniers représentent de manière univoque le comportement numérique du système et suffisent donc à la construction de l'ensemble des points d'apprentissage. Les règles sont ici présentées sous une forme TSK à conclusion constante de façon à pouvoir rapidement vérifier que la réduction n'induit aucune erreur d'approximation (cf paragraphe VIII.2). Notons que les méthodes évoquées précédemment (DVS ou approche par similarité) sont mises en échec sur cet exemple simple où l'aspect lin-

guistique n'est pourtant pas primordial puisqu'il est déjà présent dans la base de règles initiale et nécessite simplement d'être préservé ([Galichet98]). La raison de cet échec peut être trouvée dans le type de redondance présente dans les règles initiales. Celle-ci est en effet de nature interpolative et s'avère difficilement appréhendable par les approches citées. La méthodologie proposée permet d'exploiter ce type de redondance ainsi que préconisé dans [Koczy97] qui réalise l'épuration de la base de règles initiale puis utilise un mécanisme d'inférence spécifique aux bases de règles incomplètes.

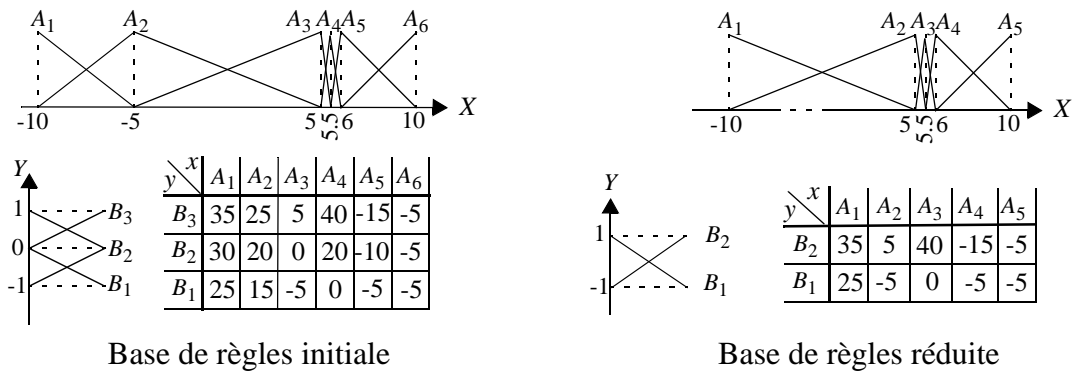


Fig. 45 : Réduction d'une base de règles.

Le deuxième exemple, extrait de [Chao96], concerne la réduction d'une base composée de 40 règles de la forme "Si  $x$  est  $A_i$  et  $y$  est  $B_j$  alors  $z = c_i$ ". Les symboles  $A_i$  et  $B_j$  ont des fonctions d'appartenance gaussiennes ainsi qu'illustré à la figure 46 sur les termes associés aux huit premières règles. Il est clair que le recouvrement des différentes significations floues induit une perte totale de la lisibilité des règles. L'objectif de la simplification de règles est donc d'améliorer l'aspect linguistique du système flou. La réduction du nombre de règles est ici accessoire dans la mesure où un partitionnement strict des univers d'entrée permettra déjà de diminuer le temps de calcul nécessaire à l'obtention de la sortie et ce, quel que soit le nombre de règles final. En effet, seules les quatre règles activées devront être évaluées, ce qui représente un gain de temps important par rapport à l'exploitation des quarante règles initiales.

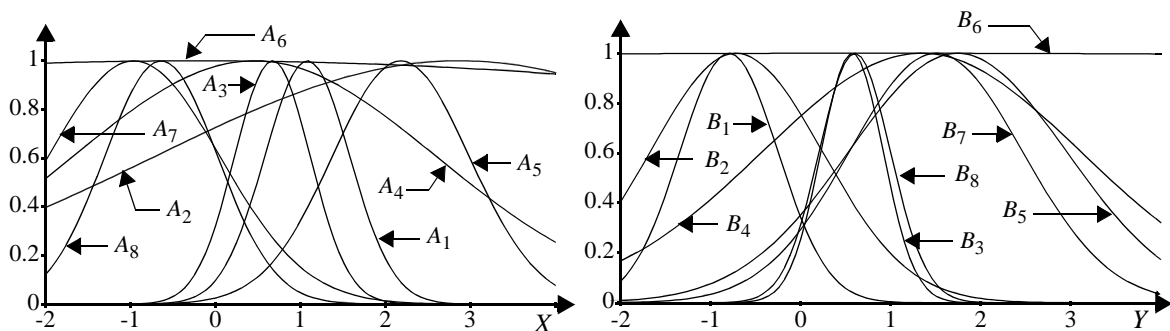


Fig. 46 : Quelques fonctions d'appartenance associées aux entrées proposées dans [Chao96].

La méthode proposée permet d'obtenir une approximation numérique satisfaisante à partir de 8x8 règles ainsi qu'en atteste l'évolution de l'erreur quadratique moyenne en fonction de la structure de la base de règles (cf figure 47). Si la lisibilité de la base de règles correspondante est jugée insuffisante, il est possible de renoncer à la qualité numérique de l'approximation pour favoriser l'aspect "résumé linguistique" procuré par la base composée de 25 règles. La figure 48 illustre la surface entrées/sortie du système flou initial ainsi que celles correspondant aux bases de règles reconstruites. On peut remarquer que la base de 25 règles permet déjà une reproduction réaliste de la surface d'origine. Signalons que la génération des données

d'apprentissage, exploitées pour la synthèse du nouveau système flou, est réalisée de façon à aider au bon déroulement de la méthode de reconstruction des règles. Ainsi, des données d'entrée régulièrement réparties sont utilisées pour chacune des variables, soit  $21 \times 21 = 441$  points d'apprentissage répartis sur un maillage rectangulaire. Cette façon de faire évite la phase de remplissage nécessaire lorsqu'un nouveau point modal est introduit et que la sortie correspondante est inconnue (cf paragraphe VII.2).

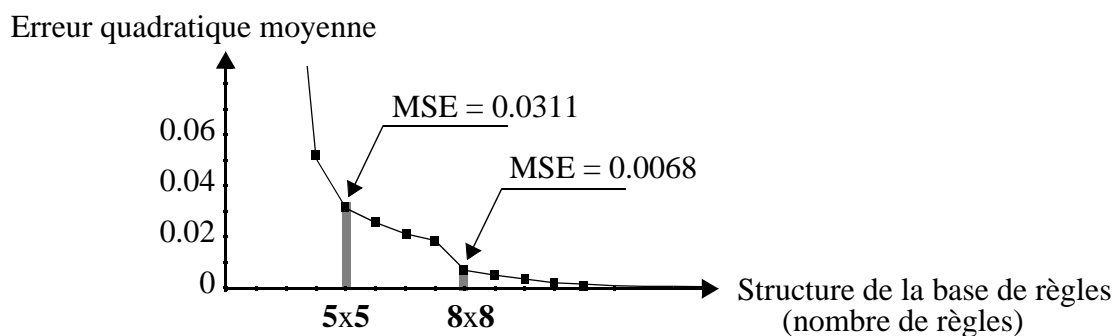


Fig. 47 : Evolution de l'erreur quadratique moyenne en fonction du nombre de règles reconstruites.

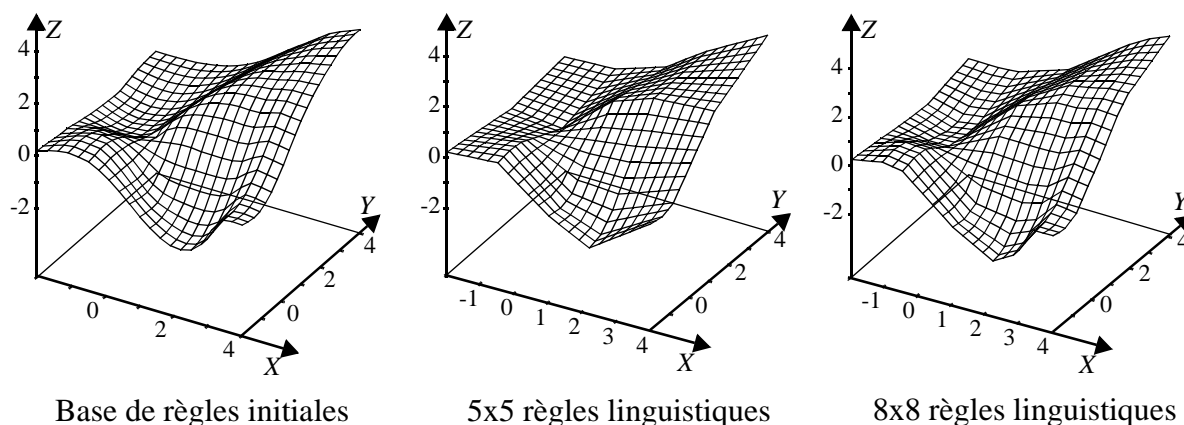


Fig. 48 : Surfaces entrées / sortie.

Dans ce chapitre, nous avons tout d'abord introduit les systèmes flous à règles linguistiques (systèmes SF4), puis établi les liens existant avec le raisonnement approximatif et les approches numériques développées en première partie de ce document. Les outils étant définis et positionnés, nous pouvons maintenant étudier comment les intégrer dans une structure de commande et plus particulièrement comment les exploiter pour coder l'expertise disponible sur le procédé ou sur la façon de le piloter. C'est ce dernier point qui guide la fin de ce mémoire dans laquelle sont présentées diverses approches à l'intégration de la connaissance experte.

## V Expertise en commande floue

L'intérêt majeur des systèmes SF4 réside dans leur lisibilité. Si cette propriété est particulièrement intéressante dans un contexte d'analyse qualitative d'une structure de commande, elle l'est également dans une perspective de synthèse. En effet, si un expert est capable d'interpréter les règles d'un système SF4, il est aussi capable de formuler sa connaissance sous la forme de règles directement intégrables dans un système SF4. Cette double compétence de l'expert conduit à

envisager deux façons différentes de travailler avec lui. La première exploite sa capacité à produire des règles de commande. D'un point de vue pratique, elle correspond à construire le contrôleur flou directement à partir des règles fournies par l'expert. La seconde tire profit de sa faculté d'analyse et d'interprétation d'un système SF4. La démarche envisagée est alors de lui soumettre une version initiale de contrôleur qu'il pourra analyser puis modifier de façon à améliorer les performances de commande. Dans ce contexte, le contrôleur flou initial peut être généré par copie d'un contrôleur numérique existant.

Cette dichotomie dans la façon d'appréhender l'introduction de l'expertise dans un contrôleur flou nous a conduits à structurer le présent paragraphe en deux parties. La première intitulée "Expertise et contrôleurs symboliques" présente les différentes étapes nécessaires à une construction directe de la base de règles et du système SF4 correspondant. La seconde partie est appelée "Expertise et contrôleurs numériques" dans la mesure où le système flou est élaboré à partir d'un contrôleur numérique existant. La traduction floue de celui-ci sert de base à l'introduction de l'expertise.

Dans les deux cas, l'approche experte est abordée à partir d'applications réelles, issues du monde industriel. La présentation qui en est faite s'attache à détailler les diverses étapes de la conception du contrôleur flou de façon à pouvoir en extraire les lignes directrices. Un dernier paragraphe dresse le bilan des différentes techniques évoquées et propose une méthodologie générale de développement.

## **V.1 Expertise et contrôleurs symboliques**

La synthèse directe d'un contrôleur flou expert est appréhendée à partir de travaux développés dans le cadre de la thèse CIFRE de M. Dussud ([Dussud99]). Ils sont le fruit d'une collaboration entre le laboratoire et la société SERT (Société d'Etudes et de Réalisations Techniques), entreprise domiciliée à Lyon et spécialiste de régulation de niveau en lingotière de coulée continue. Ce partenariat a permis d'aboutir à une implantation industrielle d'un contrôleur SF4 dont la synthèse est réalisée à partir d'une expertise sur le procédé. Aucune phase de modélisation n'est introduite et la base de règles est construite par "copie" du savoir-faire des opérateurs. Après une rapide introduction à la régulation de niveau en lingotière de coulée continue, la méthodologie de développement du contrôleur SF4 est détaillée étape par étape.

La figure 49 donne un aperçu du procédé de fabrication de l'acier en coulée continue. Le métal liquide est fourni dans une poche de grande capacité qui alimente le répartiteur servant de réservoir-tampon. Par l'intermédiaire de la busette, l'acier en fusion s'écoule vers la lingotière, équipée d'un système de circulation d'eau qui assure le refroidissement nécessaire à la solidification du métal. Les rouleaux extracteurs entraînent alors la barre de métal puis la ramènent à l'horizontale afin de pouvoir procéder à sa coupe. Le niveau de métal en lingotière est maintenu constant par une boucle de commande comprenant un capteur de niveau, un régulateur et un actionneur agissant sur la position de la quenouille par le biais du mécanisme de quenouille.

Il est communément admis que la qualité du produit final est fortement corrélée à la performance de la régulation de niveau en lingotière ([Jenkins94]). Cette constatation conduit d'ailleurs les sidérurgistes à "déclasser" de manière systématique les barres d'acier obtenues lors de variations de niveau importantes. De nos jours, la plupart des installations industrielles sont équipées de PID qui s'avèrent performants en fonctionnement normal mais inefficaces lorsqu'apparaissent des perturbations brutales telles que celles induites par des incidents de bouchage/débouchage de la busette (cf figure50).

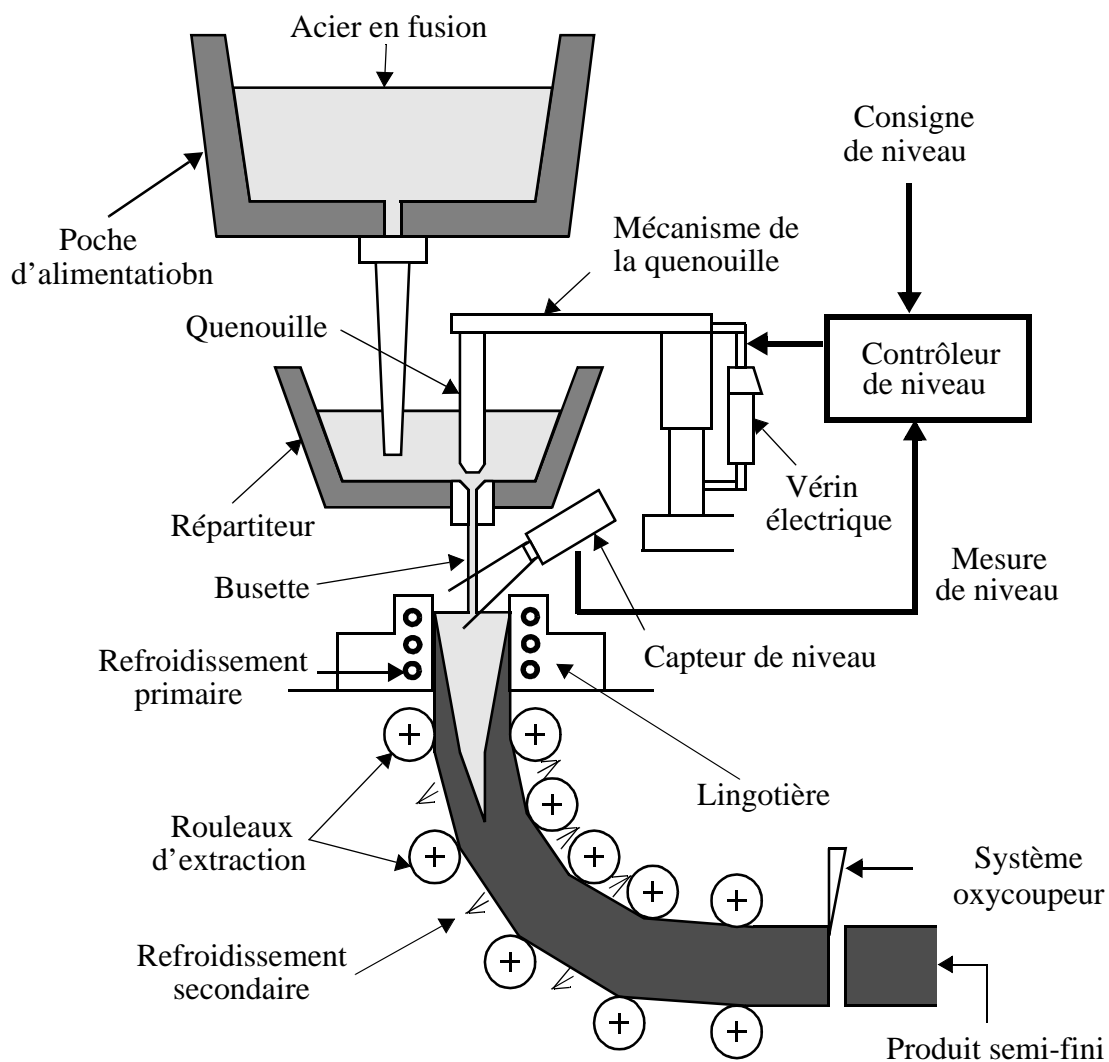


Fig. 49 : Vue schématique d'une machine de coulée continue.

Le débouchage de la busette entraîne une brusque augmentation de niveau pouvant aller jusqu'à un débordement du métal en fusion si l'opérateur ne reprend pas la main sur le pilotage de la quenouille.

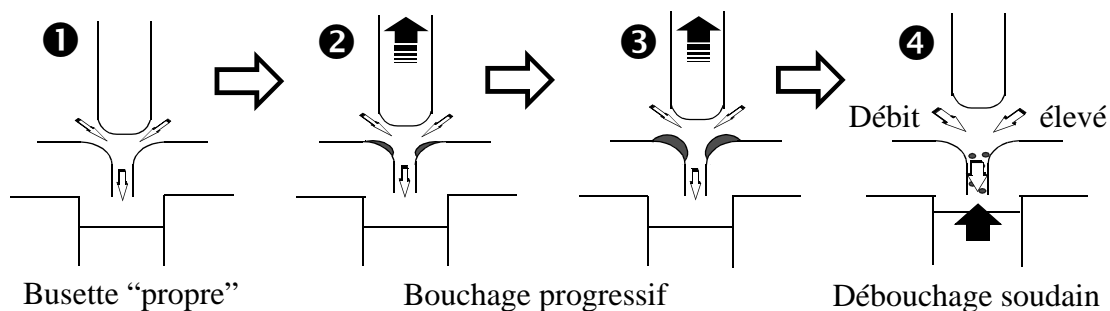


Fig. 50 : Cycle de bouchage/débouchage de la busette.



L'objectif de la structure de commande proposée (cf figure 51) est d'éviter au maximum les reprises manuelles. Le principe mis en oeuvre ([Dussud98]) repose sur les points suivants :

- En mode normal, le contrôleur PID est utilisé.
- En mode perturbé, le contrôleur fluo est substitué au contrôleur PID.
- La commutation est réalisée sur la valeur de commande produite par le contrôleur fluo.

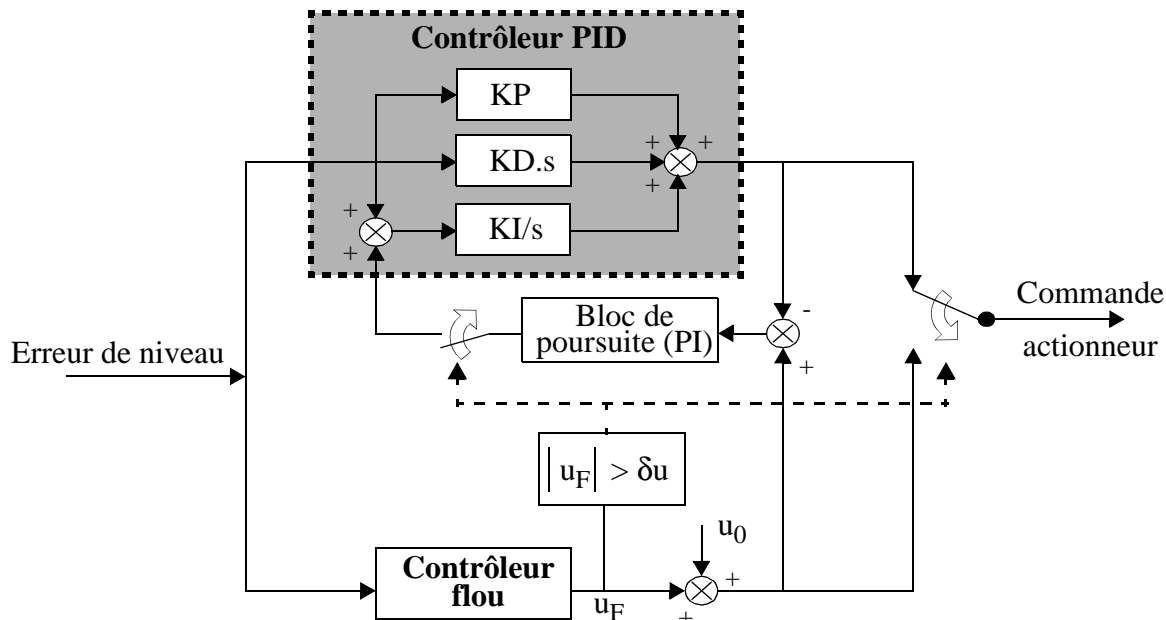


Fig. 51 : Architecture globale du contrôleur.

Le passage d'un régulateur à l'autre est implanté de façon à éviter les "à-coups" tel que cela est illustré à la figure 52. Le contrôleur fluo est activé lorsque  $|u_F| > \delta u$ . Dans ce cas, la commande effective est obtenue en additionnant la contribution fluo  $u_F$  à la dernière sortie du PID  $u_0$ . Le seuil  $\delta u$  est fixé suffisamment bas pour rendre négligeable l'incrément de commande. Durant toute la période de contrôle fluo, le PID fonctionne en mode "suiveur" par forçage de sa sortie à la valeur de commande effective grâce à un bloc de poursuite de type PI. Il est ainsi garanti que la commutation inverse n'engendre pas d'à-coup important.

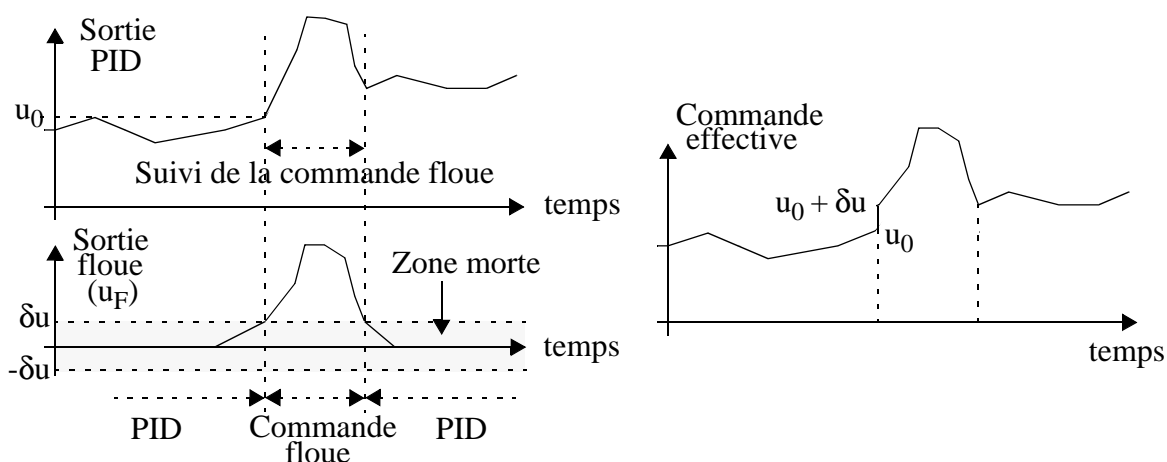


Fig. 52 : Commutation sans "à-coup".

La synthèse du contrôleur flou est délicate dans la mesure où elle fait appel à l'appréciation humaine pour traduire la connaissance experte. La démarche présentée ici tente d'apporter une certaine rigueur en décomposant la construction en différentes phases. En effet, on établit tout d'abord les règles générales de fonctionnement du contrôleur flou sur l'intégralité de l'univers de discours. Les règles ainsi générées sont alors modifiées pour gérer la bande morte nécessaire à la commutation avec le PID. Le traitement des débouchages est ensuite introduit par ajout de nouvelles règles créant une action anticipative. Enfin, le réglage fin du contrôleur est effectué par pondération des règles.

### V.1.1 Règles générales

Il s'agit ici d'établir le comportement général du contrôleur flou d'après la formulation de l'expert. Celui-ci distingue trois situations selon que le niveau d'acier est "trop haut", "correct" ou "trop bas" relativement à une consigne fixée. Cette classification est aisément retranscrite en terme d'erreur "**P**ositive", "**Z**éro" et "**N**égative" en définissant l'erreur comme étant la différence entre la mesure et la consigne. Quoique non conventionnelle, cette façon de faire est intuitive dans la mesure où elle permet une équivalence entre les termes "trop haut", "correct" et "trop bas" et les mots "**P**ositive", "**Z**éro" et "**N**égative", ainsi qu'illustré à la figure 53. L'opérateur est également sensible au sens de variation du niveau et considère trois cas selon que le niveau augmente, est stable ou diminue. La variation d'erreur correspondante est alors qualifiée de "**P**ositive", "**Z**éro" ou "**N**égative". Enfin, d'après sa perception de la situation, l'expert détermine l'action à adopter concernant la quenouille : "**M**aintenir", "**O**uvrir" ou "**F**ermer". Lorsqu'il hésite entre deux actions possibles, les deux solutions sont introduites dans la base de règles (cf figure 53).

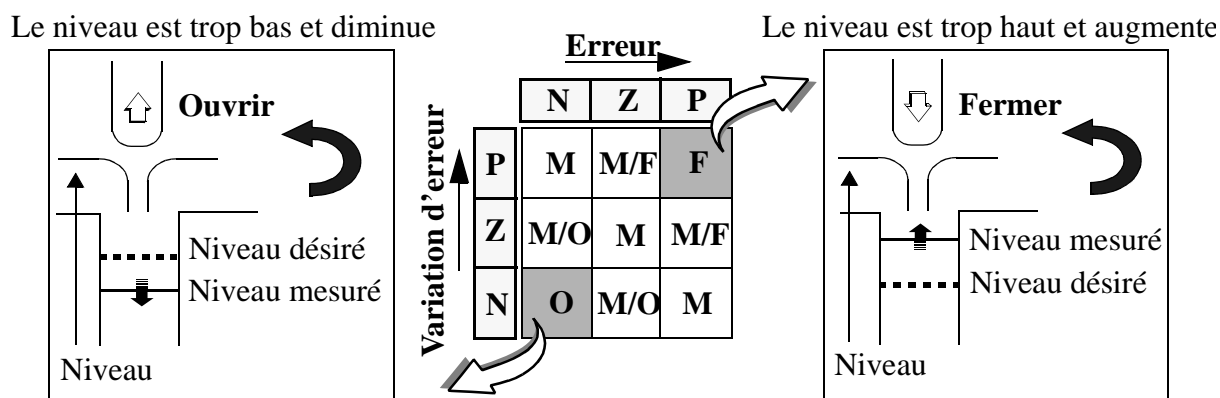


Fig. 53 : Règles générales.

### V.1.2 Gestion de la commutation

La base de règles est maintenant modifiée pour satisfaire aux exigences de commutation de la stratégie de contrôle élaborée. Ainsi, le contrôleur flou doit produire une action nulle (**M**) lorsque le procédé est en fonctionnement normal. Une telle situation correspond à une zone rectangulaire autour du point (0, 0) dans le plan de phase et coïncide donc au cas décrit par la proposition "Erreur est **Z** et Variation d'erreur est **Z**". L'examen de la base de règles permet de conclure que l'action floue générée est bien **M**. En pratique, la variation d'erreur étant particulièrement bruitée, son utilisation pour décider d'une commutation risque d'induire de nombreux passages d'un régulateur à l'autre. La zone de fonctionnement du PID est donc étendue à une bande verticale dans le plan de phase ainsi qu'en témoigne la base de règles modifiée de la figure 54.

		<b>Erreur</b> →		
		N	Z	P
<b>Variation d'erreur</b> ↑	P	M	M	F
	Z	M/O	M	M/F
	N	O	M	M

Fig. 54 : Stratégie de commutation.

### V.1.3 Règles anticipatives

Reste maintenant à introduire de nouvelles règles dans la base afin de rejeter les perturbations brutales de type débouchage. Les experts ont proposé d'utiliser une action anticipative pour réagir préventivement lorsque des variations de niveau soudaines et importantes surviennent. Ce principe consiste en fait à fermer la quenouille lorsque le niveau monte même s'il est en dessous de la consigne. De même, afin de faciliter le retour au fonctionnement normal, la quenouille est ouverte même si le niveau est au dessus de la référence. L'introduction des règles supplémentaires nécessite la définition de deux nouveaux termes linguistiques pour décrire l'erreur (**N**égative**P**etite et **P**ositive**P**etite). Ils représentent des sous-classes des classes **N** et **P** préalablement définies ainsi qu'illustré à la figure 55.

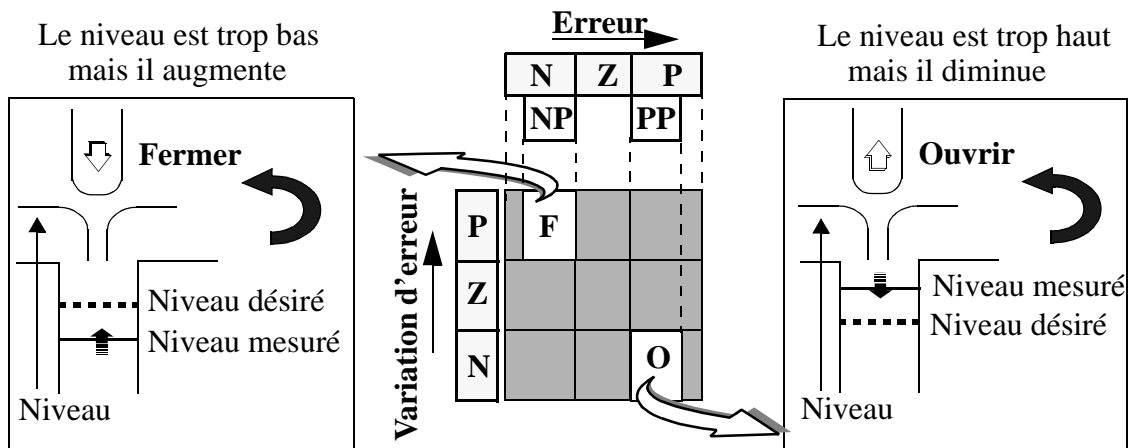


Fig. 55 : Introduction des règles anticipatives.

### V.1.4 Ajustement du contrôleur flou

La quatrième et dernière étape consiste à régler le contrôleur flou. En effet, les règles étant maintenant construites, il reste à établir les interfaces SN et NS pour pouvoir implanter la loi de commande floue. Il est également nécessaire de calibrer l'importance respective des actions anticipatrice et naturelle puisqu'elles peuvent être activées simultanément. Enfin, les conclusions multiples sont pondérées selon l'opinion de l'expert. Les règles finales sont données à la figure 56 sous la forme d'un système SF4. Remarquons que malgré la présence de cellules vides, la base de règles est complète puisqu'il est possible de calculer la sortie du système flou pour n'importe quel couple d'entrées. Ceci est dû au fait que le support des fonctions d'appar-

tenance associées aux termes linguistiques **NP** et **PP** est également couvert par la signification floue des symboles **N** et **P**.

		Erreur				
		N	NS	Z	PS	P
Variation d'erreur	P	0.3/M	0.6/F	0.5/M		1/F
	Z	0.4/M+0.4/O		1/M		0.4/F+0.4/M
	N	1/O		0.5/M	0.6/O	0.3/M

Fig. 56 : Base de règles finale.

La structure de commande globale est implantée sur une installation industrielle depuis Janvier 96. Elle a permis d'améliorer notablement la régulation durant les phases perturbées qui suivent les débouchages de busette. La figure 57 présente l'évolution de niveau pendant un débouchage réel. Les performances obtenues sont satisfaisantes dans la mesure où le débouchage, d'amplitude importante puisqu'il peut être assimilé à une perturbation additive de 10.4 mm sur la position de la quenouille, induit un dépassement de niveau acceptable et que le temps de retour à la normale est réduit à 7s. Ces résultats valident dans un cadre applicatif la stratégie de synthèse d'un contrôleur flou par exploitation directe de règles expertes. Il est clair qu'il serait abusif de conclure à la généralité de l'approche. Toutefois, dans la mesure où un procédé est pilotable manuellement, il est tout à fait imaginable de reproduire le comportement de l'opérateur humain par un système SF4. L'efficacité du contrôleur ainsi généré n'est pas garantie, pas plus d'ailleurs que ne l'est celle du "contrôleur" humain ayant servi de modèle. Ceci dit, l'obtention de performances optimales n'est envisageable qu'au prix d'une phase de modélisation pouvant être longue et coûteuse.

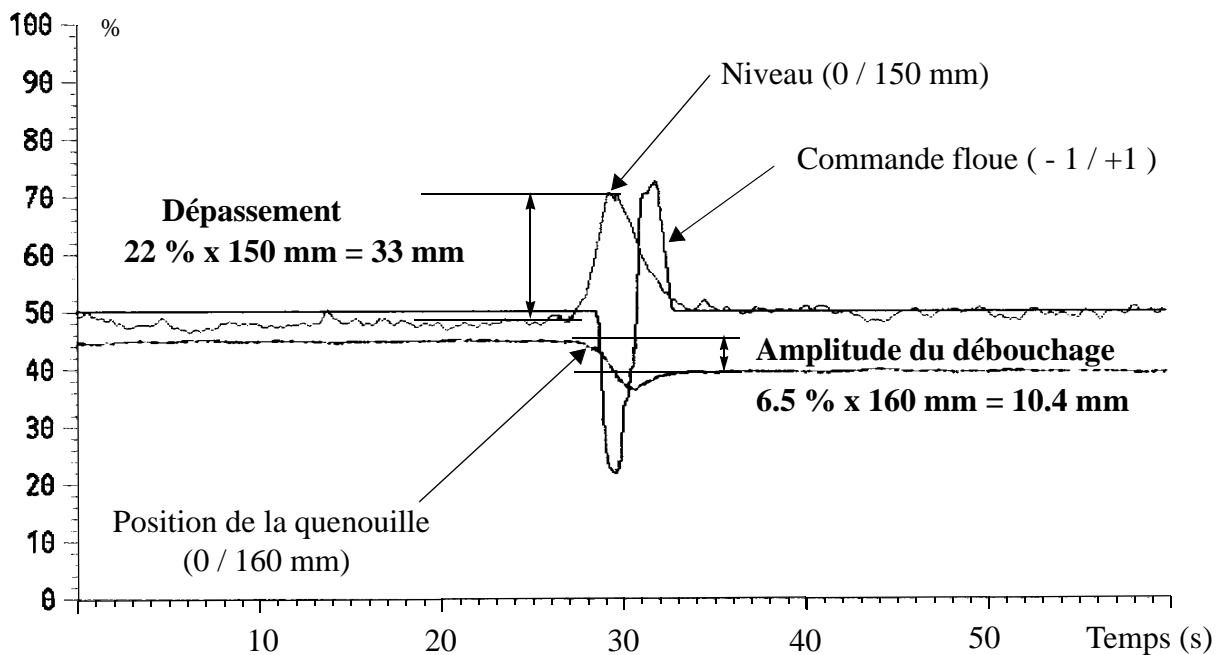


Fig. 57 : Débouchage réel d'une busette.

Revenons quelques instants à la problématique de la coulée continue de façon à discuter, non plus de la synthèse du contrôleur flou, mais du choix d'une architecture de contrôle hybride. La décision d'une implantation à deux contrôleurs (PID + flou) repose en fait sur des considérations pratiques. La raison majeure en est que l'architecture proposée est en tout point similaire à celle préalablement implantée sur l'installation industrielle. Ainsi, si l'on considère le schéma de la figure 51 et que l'on remplace le contrôleur flou par un mécanisme de reprise manuelle par un opérateur, on retrouve la configuration existant à l'origine. Le coût d'implantation du système flou est alors réduit à un temps de développement logiciel, le procédé étant déjà connecté à un ordinateur PC chargé d'enregistrer les données de fabrication. Outre son prix minime, la solution choisie présente l'avantage d'assurer une bonne fiabilité puisque la régulation de base est assurée par des composants sécurisés (automates ou régulateurs analogiques). Enfin, d'un point de vue psychologique, la démarche proposée est rassurante pour l'industriel puisqu'elle permet à tout instant de revenir à la configuration antérieure par une simple déconnexion du contrôleur flou.

Pour des installations où ces arguments ne sont pas valides, il est intéressant de réduire l'architecture de contrôle proposée à un simple contrôleur flou qui inclurait directement le PID par le biais de règles équivalentes (cf paragraphe IV.5). C'est le choix effectué dans le paragraphe suivant où la phase d'introduction de l'expertise est effectuée après avoir intégré le contrôleur numérique existant dans un système SF4.

## **V.2 Expertise et contrôleurs numériques**

La synthèse de la loi de commande floue est ici réalisée à partir d'un contrôleur numérique dont on améliore les performances en élargissant le contexte d'utilisation par ajout d'une connaissance experte. Par rapport aux travaux précédents, la différence se fait alors dans la façon dont cet ajout est géré. Le principe développé ici n'est plus d'élaborer une structure multi-contrôleurs comme cela a été pratiqué pour la régulation de niveau en coulée continue, mais d'intégrer l'ensemble des connaissances disponibles sur le procédé dans un contrôleur flou unique. Ainsi que cela a été évoqué précédemment, ce choix de conception est lié à des aspects pratiques, spécifiques à chaque installation.

En guise d'illustration de la méthodologie proposée, deux applications industrielles sont ici présentées. Dans les deux cas, des systèmes SF4 sont utilisés et leur synthèse est réalisée selon une démarche identique. Celle-ci est décomposée en deux étapes, la première consistant à construire l'équivalent flou d'un contrôleur numérique cible, la seconde visant à l'améliorer par modification des règles initiales selon l'expertise disponible.

### *V.2.1 Régulation de niveau dans un procédé de fabrication semi-continue de barres de laiton*

Cette application ([Dussud96]) concerne le remplissage de moules pour la fabrication semi-continue de barres de laiton, procédé illustré à la figure 58. Bien que le principe de fonctionnement global soit très proche de celui décrit précédemment pour la production de brames d'acier, les équipements utilisés sont suffisamment différents pour modifier la problématique de la régulation de niveau. En effet, le débit de métal liquide dans la busette est ici automatiquement ajusté par un système à base de flotteurs sur lequel aucune action n'est envisageable. La stabilité du niveau de métal dans les moules dépend alors de celle du niveau de métal dans le répartiteur. L'objectif de régulation poursuivi ici est donc de maintenir constant le niveau de laiton dans le répartiteur. Le débit d'alimentation de ce dernier est contrôlable par l'intermédiaire d'un vérin hydraulique permettant d'agir sur l'inclinaison du réservoir.

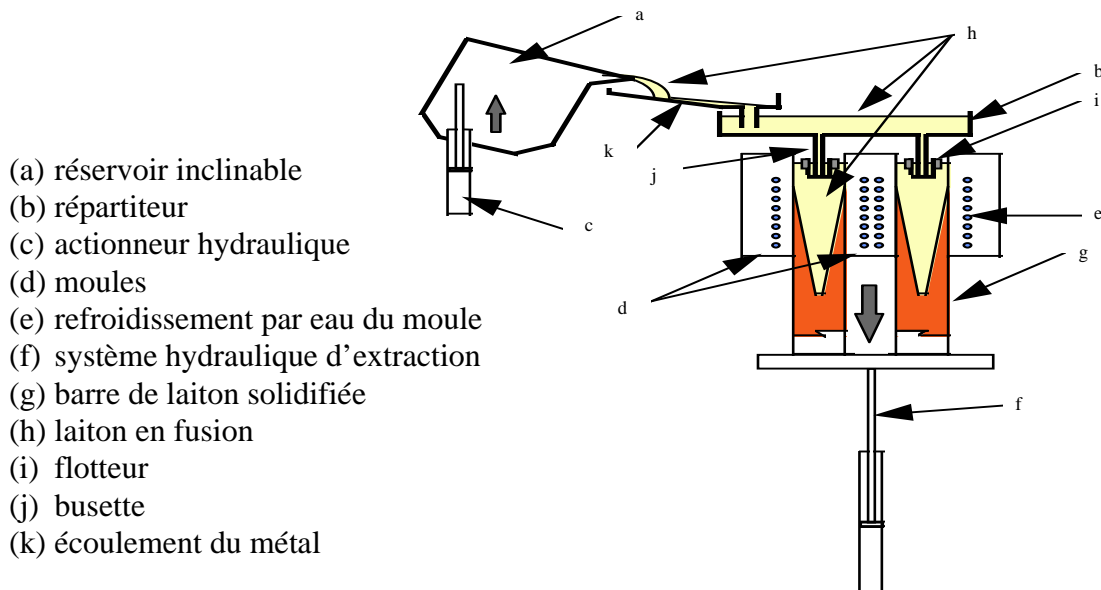


Fig. 58 : Fabrication semi-continue de barres de laiton.

Une modélisation mathématique simplifiée du procédé a permis de calibrer en simulation un contrôleur de type PD puis de synthétiser un contrôleur flou équivalent. Celui-ci est ensuite modifié de façon à contrer le retard induit par l'écoulement du métal dans la rigole. Des actions anticipatives sont ainsi introduites dans la base de règles selon l'expertise des opérateurs. A ce stade du développement, la démarche mise en oeuvre est similaire à celle présentée dans le paragraphe précédent. Le contrôleur flou final implanté est de type PD non linéaire. La surface de contrôle correspondante est illustrée à la figure 59.

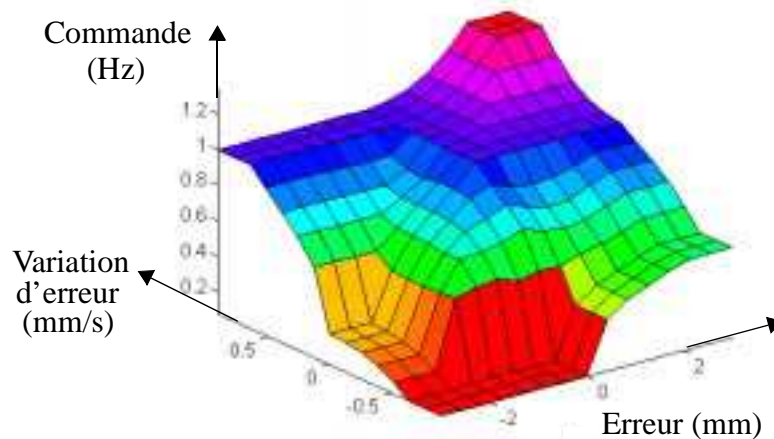


Fig. 59 : Surface de contrôle du régulateur flou implanté.

Le choix d'une implantation de la commande sous la forme d'un système flou unique a été guidé par le fait que l'installation considérée n'était équipée d'aucun régulateur spécifique puisque c'est un opérateur humain qui la pilotait. La démarche de conception a permis de réduire le temps de développement nécessaire à l'obtention de résultats satisfaisants ainsi qu'en atteste la figure 60. Reste à signaler que la phase de remplissage du moule est toujours manuelle. Lorsque celle-ci est terminée, l'opérateur enclenche le contrôleur flou qui prend en charge la régulation de niveau pendant toute la durée de la coulée (20 minutes à 1 heure). La figure 60 illustre l'amélioration de qualité apportée par la régulation floue qui s'avère plus régulière dans ses actions que l'opérateur humain ayant servi de "modèle".

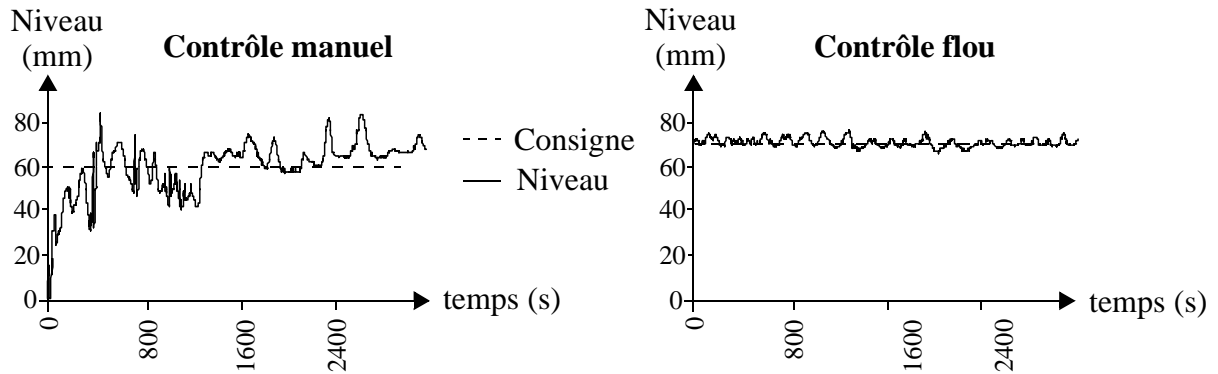


Fig. 60 : Résultats réels.

### V.2.2 Régulation de niveau en raffinerie

Rappelons que cette application, utilisée à titre d'illustration tout au long de ce document, a été développée en collaboration avec le centre de recherche d'ELF Solaize. Dans un premier temps, un contrôleur algorithmique existant a permis de construire un contrôleur flou initial par application du principe des équivalences modales (cf paragraphe VI.1 du premier chapitre). Celui-ci, de type TSK à conclusions constantes, a ensuite été converti en un système SF4 (cf paragraphe IV.1) dont la version condensée de la figure 61 sert maintenant de base de travail à l'intégration de la connaissance experte.

→ signe( $\delta L$ )\* $\delta L^2$

		$B_{-3}$	$B_{-2}$	$B_{-1}$	$B_0$	$B_1$	$B_2$	$B_3$
$L_{min}$	$A_0$	NTG	NTG	NTG	Z	PP	PM	PM
	$A_1$	NG	NG	NM	Z	PP	PM	PM
$L$	$A_2$	NG	NM	NM	Z	PM	PM	PG
	$A_3$	NM	NM	NP	Z	PM	PG	PG
$L_{max}$	$A_4$	NM	NM	NP	Z	PTG	PTG	PTG

Fig. 61 : Résumé linguistique des règles floues initiales.

La première remarque de l'expert concerne le fait qu'il est important d'être dans une situation favorable lorsqu'une perturbation apparaît puisque les marges de manoeuvre sont alors plus larges. Ceci se concrétise par le souhait que le niveau ne se stabilise qu'autour d'une valeur centrale et conduit donc à modifier les conclusions de règles associées à la colonne  $B_0$ . De même, quelle que soit l'évolution du niveau, l'expert préconise un traitement "choc" dans le cas où il est proche des limites  $L_{min}$  et  $L_{max}$ . La prise en compte de ces deux considérations aboutit à la base de règles modifiée de la Fig. 62.

L'amélioration majeure est obtenue grâce à l'intégration d'une nouvelle variable dans les prémisses. Celle-ci, construite à partir de données mesurées sur le procédé en amont de la capacité tampon, reflète l'évolution future du procédé. En fait, elle représente l'amplitude prédite des perturbations à venir et permet donc d'anticiper la démarche à suivre pour faciliter leur rejet si elles se produisent effectivement. Ainsi, si une perturbation positive importante est prévue (forte montée de niveau), il est souhaitable que le niveau soit bas pour pouvoir la traiter efficacement quand elle survient. Cette constatation amène à réagir de façon préventive par une commande positive permettant de faire baisser le niveau.

		$\longrightarrow$ signe( $\delta L$ )* $\delta L^2$						
		$B_{-3}$	$B_{-2}$	$B_{-1}$	$B_0$	$B_1$	$B_2$	$B_3$
$L_{min}$	$A_0$	NTG	NTG	NTG	NTG	NTG	NTG	NTG
	$A_1$	NG	NG	NM	NP	PP	PM	PM
	$A_2$	NG	NM	NM	Z	PM	PM	PG
	$A_3$	NM	NM	NP	PP	PM	PG	PG
$L_{max}$	$A_4$	PTG	PTG	PTG	PTG	PTG	PTG	PTG

Fig. 62 : Base de règles modifiée par expertise.

L'ajout d'une nouvelle variable dans les prémisses se traduit par des règles de la forme "Si le niveau est élevé et augmente fortement et qu'une perturbation positive importante est prédite alors la commande est positive grande". D'un point de vue pratique, sept termes linguistiques sont utilisés pour caractériser la perturbation prédite. La base de règles de la figure 62 est maintenue et correspond au cas où aucune perturbation n'est prévue. Six nouvelles bases de règles de structure identique sont introduites pour gérer les autres classes de perturbations prévues. Elles sont écrites avec l'aide de l'expert selon les considérations énoncées précédemment. Le régulateur flou ainsi généré, industrialisé sous le nom FLOUNIV par Elf, a été installé dans une première raffinerie en 94. Depuis, une dizaine de sites ont été équipés de manière similaire.

Les études décrites dans ce paragraphe ont en commun le fait qu'elles reposent sur un régulateur flou linguistique conçu pour améliorer un contrôleur numérique existant. La phase d'initialisation du système flou est effectuée de manière systématique par application du principe des équivalences modales. Cette façon de faire permet d'aborder plus facilement l'introduction du savoir-faire dans la mesure où la base de règles est déjà construite et n'a donc plus qu'à être adaptée.

### V.3 Aspect méthodologique

Il nous paraît maintenant important de conclure ce paragraphe en énonçant quelques lignes directrices à suivre pour concevoir un contrôleur flou expert. La méthodologie proposée s'appuie sur les applications présentées précédemment mais aussi sur l'expérience que nous avons pu acquérir au cours de ces dix dernières années. Elle se présente sous la forme d'une "gamme de conception", composée de différentes tâches à réaliser séquentiellement :

❶ *Répertorier l'existant.*

La première démarche à entreprendre face à un problème réel est un inventaire des ressources existantes. Celui-ci doit être établi en termes de connaissances (modèle, loi de commande, expertise, données enregistrées) mais aussi en termes d'installations matérielles (régulateur, capteur, actionneur).

❷ *Déterminer la tâche que doit accomplir le contrôleur flou.*

Au vu de l'existant, il est nécessaire de définir le rôle exact que devra assurer le contrôleur flou. Cette étape correspond en fait à fixer l'architecture globale de la structure de commande dans laquelle le contrôleur flou s'insérera.

❸ *S'assurer que l'expertise disponible est en cohérence avec la tâche définie.*

La connaissance experte est rarement disponible sous la forme de règles écrites, mais le



fait qu'un opérateur humain soit capable de résoudre le problème posé garantit l'existence de la connaissance. A ce stade, il nous paraît important de souligner que le contrôle flou n'est pas la panacée universelle et que, pas plus que d'autres approches, il n'est capable de répondre à des problèmes qui ne sont aujourd'hui résolus ni par la science ni par l'homme.

**④ Déterminer les grandeurs d'influence et s'assurer qu'elles sont accessibles ou peuvent être reconstruites**

Toutes les quantités intervenant dans le raisonnement humain que l'on cherche à reproduire via le contrôleur flou doivent être disponibles (mesurées ou reconstruites). Ce point traduit la nécessité d'équiper le système automatisé des organes de perception appropriés. Le succès de bon nombre d'applications commerciales intégrant des techniques floues est d'ailleurs étroitement lié à l'augmentation du nombre de grandeurs d'influence considérées. Citons à titre d'exemple l'intégration d'un capteur de turbidité dans les machines à laver dites intelligentes.

**⑤ Construire la base de règles.**

Quelle que soit la technique utilisée (synthèse directe ou initialisation à partir d'un contrôleur numérique), la construction des règles ne peut être faite qu'en étroite collaboration avec l'expert. La base de règles représente en fait la "fonction de transfert" implantée dans le contrôleur flou. Elle doit impérativement rester lisible de façon à ce que l'expert puisse en faire une validation finale.

**⑥ Construire les interfaces numérique/symbolique et symbolique/numérique.**

Cette étape correspond à attribuer une signification numérique à chacun des mots présents dans la base de règles. D'un point de vue pratique, elle est réalisée par un partitionnement des univers de discours numériques. D'un point de vue conceptuel, elle peut être assimilée à une identification des paramètres de la "fonction de transfert" que représente la base de règles.

**⑦ Affiner le réglage par l'intermédiaire des poids de règles.**

Cette dernière phase consiste à établir un calage final du contrôleur flou. Elle est souvent réalisée sur le procédé lui-même dans la mesure où un outil de simulation n'est pas disponible.

Quelle que soit la démarche choisie pour réaliser la construction de la base de règles (synthèse directe du système flou ou modification à partir d'une configuration initiale), l'intégration d'expertise dans le contrôleur n'est envisageable que si les règles de commande sont manipulables par un être humain. Cela se traduit par le fait que les approches utilisées jusqu'à présent ne sont applicables que pour des systèmes flous ayant un nombre réduit d'entrées. Pour des procédés complexes ou ayant un ordre élevé, il est nécessaire de réfléchir à une structuration de l'expertise de façon à pouvoir l'introduire dans différents composants, répartis dans une architecture de commande distribuée.

Cette façon de faire nous permettra également d'aborder la reconstruction des grandeurs d'influence, ainsi qu'évoqué au point ④. Dans ce contexte, l'exploitation d'informations issues de différents capteurs sera facilitée par la possibilité de concevoir des composants chargés de réaliser la fusion des informations élémentaires. Les techniques floues offrent un cadre naturel à l'intégration de l'expertise dans ces derniers ainsi que l'illustrent les paragraphes suivants préconisant l'usage de capteurs flous dans les structures de commande. Ceux-ci, très brièvement décrits ici, devraient faire l'objet d'une prochaine habilitation à diriger des recher-

ches au sein de l'équipe.

## VI Distribution de l'expertise dans la structure de commande

### VI.1 Introduction

De manière classique, les systèmes flous à plusieurs entrées exploitent des règles à prémisses composées du type :

$$\text{Si } x_1 \text{ est } A_1^{i_1} \text{ et } x_2 \text{ est } A_2^{i_2} \text{ et } \dots \text{ et } x_n \text{ est } A_n^{i_n} \text{ alors } u \text{ est } C_k \quad (106)$$

Cette façon de faire entraîne une croissance exponentielle du nombre de règles avec le nombre d'entrées. Ainsi, pour un système à plus de trois variables, il devient illusoire de préserver la lisibilité de la base de règles et la démarche experte introduite précédemment devient inapplicable. Dans ce contexte, il est nécessaire de réfléchir à une distribution de l'expertise dans la structure de commande.

Pour poser les bases de la distribution, il est souhaitable d'avoir une compréhension en profondeur des systèmes à entrées multiples décrits par les règles (106) et notamment de revenir à la différence fondamentale existant entre les implantations SF1 et SF4 (cf paragraphe II.2). Quel que soit le type de représentation choisi, l'inférence est implantée selon le principe de combinaison/projection qui se décline en :

$$\forall u \in U$$

$$\mu_{F_1}(u) = \sup_{(x_1, \dots, x_n) \in X_1 \times \dots \times X_n} T_1 \left( T_2 \left( \mu_{\varphi_1(x_j^0)}(x_j) \right), \mu_R(x_1, x_2, \dots, x_n, u) \right) \quad (107)$$

pour une implantation purement numérique ( $\rho_1$  dans la typologie) et en :

$$\forall C_k \in LU$$

$$\mu_{F_2}(C_k) = \max_{(A_1^{i_1}, \dots, A_n^{i_n}) \in LX_1 \times \dots \times LX_n} T_1 \left( T_2 \left( \mu_{\varphi_2(x_j^0)}(A_j^{i_j}) \right), \mu_R(A_1^{i_1}, A_2^{i_2}, \dots, A_n^{i_n}, C_k) \right) \quad (108)$$

dans le cas d'une implantation symbolique ( $\rho_4$  dans la typologie).

Les équations (107) et (108) mettent en évidence le fait que le mécanisme de calcul combine, dans un formalisme unique, l'inférence à proprement parler (opérateurs sup/max et  $T_1$ ) et l'agrégation des variables d'entrée (opérateur  $T_2$ ). En fait, les entrées numériques  $x_j^0$  sont fuzzifiées indépendamment puis les sous-ensembles flous produits ( $\varphi_1(x_j^0)$  ou  $\varphi_2(x_j^0)$  suivant le choix d'implantation) sont agrégés de façon à former un sous-ensemble flou multidimensionnel qui représente la véritable entrée d'inférence. Au premier abord, il semble donc relativement simple de distinguer l'étape de fusion d'informations de celle associée au raisonnement. Un examen plus avancé des équations (107) et (108) révèle cependant que ces deux tâches sont beaucoup plus imbriquées qu'il n'y paraît. En effet, la relation  $R$  code de façon implicite une phase d'agrégation, réalisée cette fois sur la connaissance exprimée et non plus sur les grandeurs mesurées. Dans les systèmes SF4, l'agrégation de connaissances est clairement implantée au niveau symbolique puisqu'aucune information numérique n'est nécessaire à l'établissement de  $R$ . Le cas numérique porte beaucoup plus à confusion puisque la construction de  $R$  repose généralement sur l'utilisation d'opérateurs flous agissant sur la signification floue des termes utilisés dans les règles. L'exploitation de la relation engendrée induit alors une agrégation des grandeurs mesurées, le "et" utilisé dans les prémisses de règles se traduisant par une conjonction des  $\mu_{A_j^{i_j}}(x_j^0)$ . En d'autres termes, l'implantation numérique rend délicate la distinction des phases d'agrégation des grandeurs mesurées et de raisonnement. Au contraire l'implantation SF4 préconisée ici permet de dissocier clairement ces deux étapes. Notons que

cette constatation appuie le discours préalablement tenu sur la différence des implantations SF1 et SF4 (cf Fig. 35 ou encore début du paragraphe III concernant le traitement d'entrées imprécises).

Cette réflexion peut paraître anecdotique mais elle prend toute son importance lorsqu'une distribution des traitements est envisagée. En effet, l'implantation SF4 permet d'exploiter des entités différentes pour réaliser les trois étapes que représentent la fuzzification symbolique, l'inférence symbolique et la défuzzification symbolique. Dans ce contexte, le composant assurant la fuzzification peut prendre en charge l'agrégation de mesures élémentaires pour construire la description symbolique d'un vecteur. Il est ainsi possible de décrire des grandeurs subjectives telles que la couleur, le confort, ou encore le style de conduite d'une personne par agrégation des mesures numériques sous-jacentes. Lorsque la description floue d'une grandeur est construite dans l'organe effectuant les mesures, celui-ci prend l'appellation de capteur flou ([Mauris92], [Benoit93], [Mauris94], [Foulloy94a]). Dans la suite, nous attribuerons cette dénomination à toute entité logicielle produisant la description symbolique du vecteur de mesures qui lui est fourni.

## VI.2 Expertise et capteurs flous

La fuzzification symbolique étant déportée vers le capteur flou, il est possible d'introduire l'expertise dès la phase de conception de ce dernier. Cette façon de faire présente l'énorme avantage de pouvoir déterminer le type d'agrégation le mieux adapté à la situation et ce, en fonction des connaissances expertes disponibles. En ce sens, le fait de déplacer l'agrégation des grandeurs élémentaires aboutit à une liberté beaucoup plus importante quant à la manière de réaliser celle-ci.

Deux types distincts d'expertise sont ici considérés. Le premier concerne le cas où l'expert n'est pas capable de formuler explicitement le lien existant entre les mesures et la grandeur à caractériser. Par contre, il sait donner des exemples correspondant aux catégories qu'il distingue. Une telle situation se trouve dans la vie de tous les jours, où l'homme est en mesure de montrer différents objets de couleur jaune alors qu'il est incapable d'exprimer la relation entre le mot jaune et les intensités R(ouge), V(ert), B(leu) classiquement utilisées en imagerie numérique. Le second type d'expertise envisagé correspond au cas où cette dernière est disponible sous la forme de règles linguistiques reliant les grandeurs mesurées à la notion que l'on cherche à décrire. Ainsi, suivant la nature de la connaissance disponible, le capteur flou implanté mettra en oeuvre une agrégation par interpolation ou encore une agrégation par règles ([Mauris96]), ainsi qu'illustré dans les exemples présentés ci-dessous.

### VI.2.1 Agrégation par interpolation

L'implantation de ce type de capteurs nécessite que soit établie la liste des termes linguistiques qui seront utilisés pour la description d'un vecteur quelconque. Pour chaque terme, il est demandé à l'expert de fournir un ensemble de points caractéristiques. La méthode consiste ensuite à construire la signification floue associée aux différents symboles par une technique d'interpolation basée sur une triangulation de Delaunay de l'espace d'entrée multidimensionnel ([Benoit93]). Les points caractéristiques donnés par l'expert sont considérés comme étant des points modaux de la signification floue du terme linguistique concerné. Un partitionnement strict de l'univers de discours multidimensionnel est imposé.

La figure 63 illustre l'implantation d'un capteur d'état dans le plan de phase ([Foulloy94b]). Trois états distincts sont considérés, qualifiés respectivement de "au dessous de la droite de glissement", "sur la droite de glissement" et "au dessus de la droite de glissement" et abrégés symboliquement par  $\swarrow DG$ ,  $DG$ ,  $\nearrow DG$ . Trois points caractéristiques sont associés à chacun de

ces états. La connaissance introduite par le biais des exemples étiquetés est interpolée de façon à construire la signification floue des trois états linguistiques considérés. La fuzzification linguistique d'un vecteur  $(\epsilon, \delta\epsilon)$  quelconque est ensuite obtenue par exploitation directe de la relation liant la signification floue d'un symbole et la description floue d'un vecteur d'entrée, soit :

$$\forall L \in \{\neg DG, DG, \neg\neg DG\} \quad \mu_{\varphi_2(\epsilon, \delta\epsilon)}(L) = \mu_{D(\epsilon, \delta\epsilon)}(L) = \mu_L(\epsilon, \delta\epsilon). \quad (109)$$

Le capteur flou d'état peut être intégré de façon naturelle dans une structure de commande. La fuzzification étant déportée dans le capteur, seules les tâches d'inférence et de défuzzification sont implantées dans le contrôleur. Si l'expert est capable d'énoncer des règles liant l'état du procédé à la commande, leur introduction dans le processus d'inférence est immédiate. Ainsi, dans le cas du capteur d'état de la figure 63, la base de règles associée au contrôleur se résume aux trois règles élémentaires suivantes :

- Si** état est  $\neg DG$  **alors** commande est Négative
- Si** état est  $DG$  **alors** commande est Zéro
- Si** état est  $\neg\neg DG$  **alors** commande est Positive.

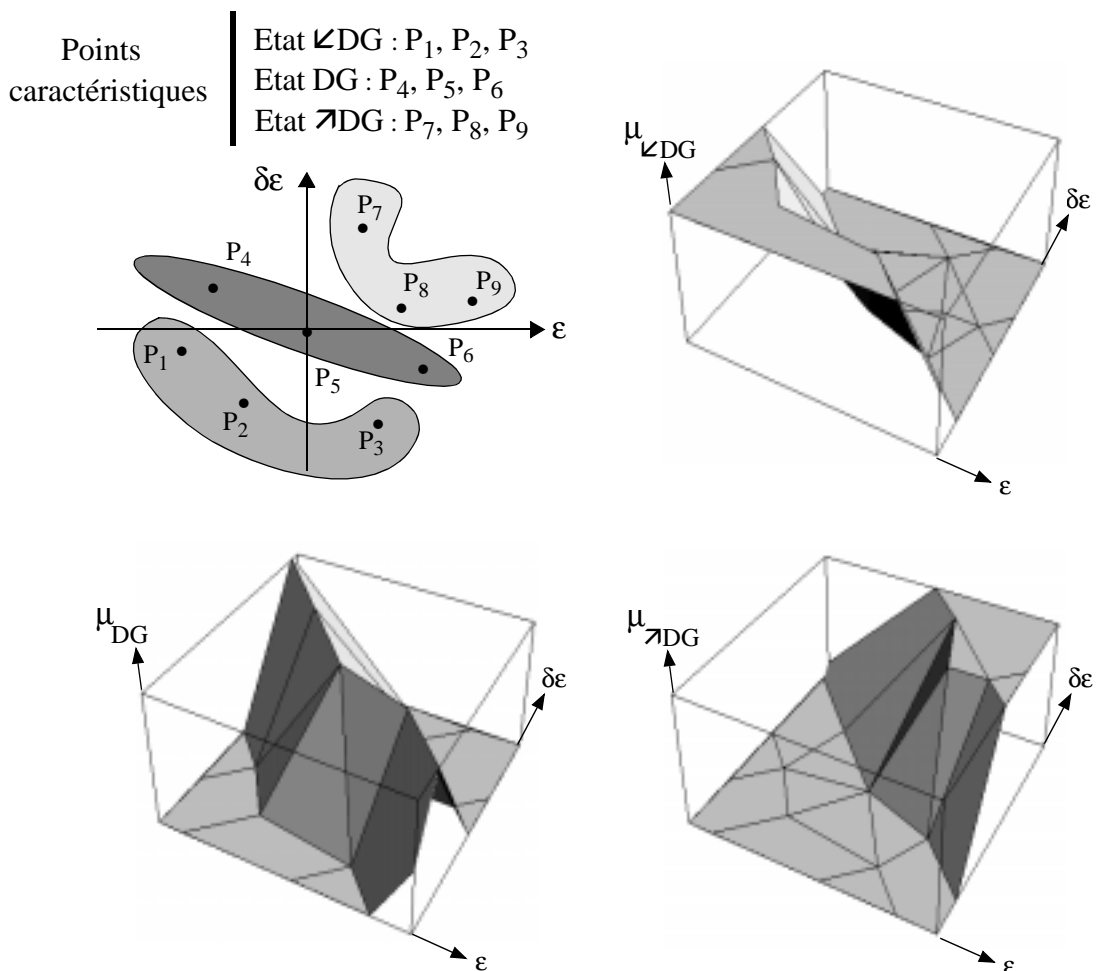


Fig. 63 : Capteur flou d'état.

La mise en place de la défuzzification nécessite d'attribuer une signification floue aux termes linguistiques utilisés pour décrire la commande. La figure 64 présente la surface de contrôle obtenue lorsque les valeurs modales -1, 0 et 1 sont associées aux mots Négative, Zéro et Positive et que la défuzzification est implantée par la méthode des hauteurs. Elle met en évidence que la

structure de commande globale, composée du capteur flou d'état, de l'inférence et de la défuzzification, correspond à un PD non linéaire.

Les points appris par le capteur d'état ( $P_1$  à  $P_9$  dans la figure 63) n'étant pas répartis sur un maillage rectangulaire du plan de phase, une implantation classique de cette loi de commande aurait nécessité la définition de neuf symboles sur l'erreur et sa variation et donc de 81 règles à prémisses composées. L'implantation proposée permet de réduire de façon importante le nombre des règles de commande. Cette simplification de la base de règles est rendue possible par le fait que les significations floues des termes associés à l'état ont un support non rectangulaire. La méthode interpolative mise en oeuvre dans le capteur permet en fait de s'affranchir de la contrainte de décomposabilité en mailles rectangulaires imposée par les règles à prémisses conjonctives. Il est également important de noter que la lisibilité du système est préservée dans la mesure où c'est en fait l'expert qui définit, via des points caractéristiques, les différentes situations qu'il distingue. La caractérisation linguistique des états ainsi identifiés permet d'aboutir à des règles de contrôle d'autant plus interprétables qu'elles traduisent de manière immédiate l'expertise disponible en terme de commande.

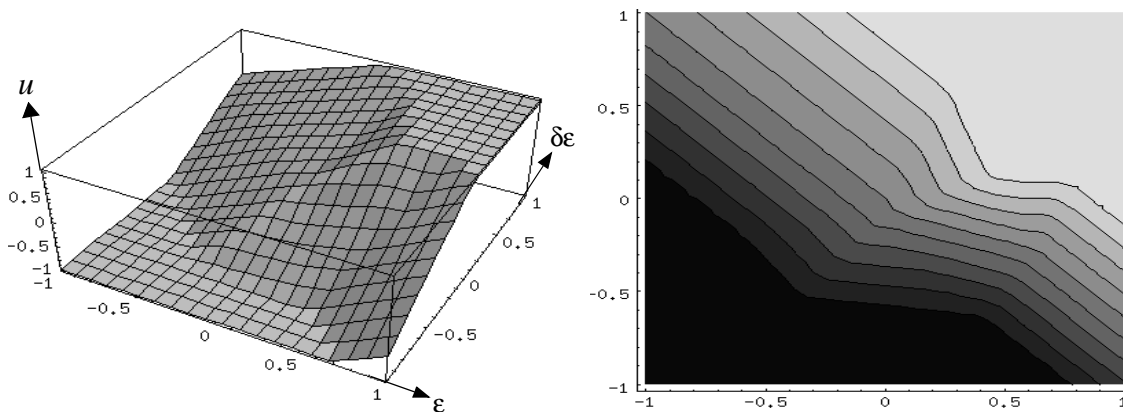


Fig. 64 : Surface de contrôle et courbes de niveaux associées.

La figure 65 illustre une autre application possible de la structuration proposée sur un problème de commande en vitesse d'un mini-robot, l'objectif de ce dernier étant de rallier le point de coordonnées  $(0, 0)$ . A tout instant, la position  $(x, y)$  du robot est connue. Dans sa partie inférieure, le graphique 65 présente la connaissance utilisée pour l'implantation de la structure de commande. Les flèches en pointillés matérialisent la manière dont l'expertise est intégrée dans les différents composants flous.

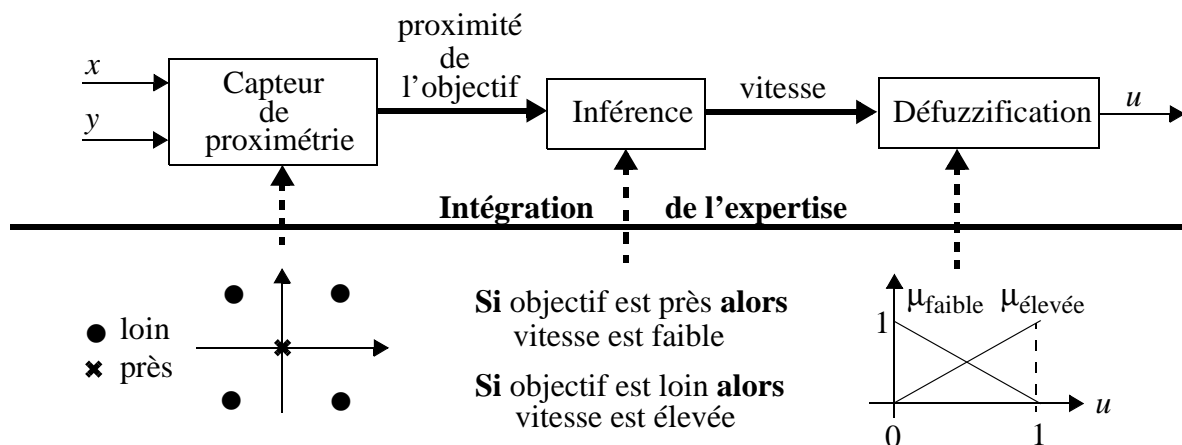


Fig. 65 : Contrôle en vitesse d'un mini-robot.

D'un point de vue conceptuel, la technique d'apprentissage implantée dans le capteur flou est très proche de celle développée en première partie de ce document (cf paragraphe VII.2) puisqu'elle repose sur l'apprentissage exact d'un certain nombre de points. Un affinement du processus peut ainsi être envisagé selon une approche incrémentale ainsi que préconisée précédemment. Cette stratégie nécessite d'avoir à disposition un ensemble de points correspondant au transfert entrées/sortie désiré de la structure de commande globale. Dans le cas du mini-robot, ces données sont donc des triplets  $(x, y, u)$ . La structure de commande établie à partir de l'expertise permet de calculer pour tout vecteur  $(x, y)$  la sortie  $u$  correspondante et donc de déterminer le point d'apprentissage engendrant l'erreur maximale. Le capteur flou est alors modifié de façon à apprendre exactement ce point, noté ici  $(x_p, y_p, u_p)$ . Cette étape est réalisée par l'introduction du couple  $(x_p, y_p)$  dans le maillage de Delaunay. La déformation des significations floues nécessite de déterminer quel doit être le degré d'appartenance de  $(x_p, y_p)$  aux différentes classes. L'information est en fait reconstruite à partir d'une base de règles inverse, obtenue par simple permutation des prémisses et conclusions dans les règles de commande initiales. La fuzzification de  $u_p$  puis l'inférence selon la base de règles inverse génère les degrés d'appartenance recherchés. Il est à noter que ce mécanisme n'est applicable que lorsque les règles de commande ne comportent pas de prémisses composées (cf paragraphe VIII.3 concernant l'inversion d'un système flou).

A titre d'illustration, la figure 66 présente la signification des symboles près et loin obtenue après affinement incrémental des significations initiales construites par expertise. Les points d'apprentissage utilisés ici ont été générés à partir de l'équation  $u = x^2 + y^2$ .

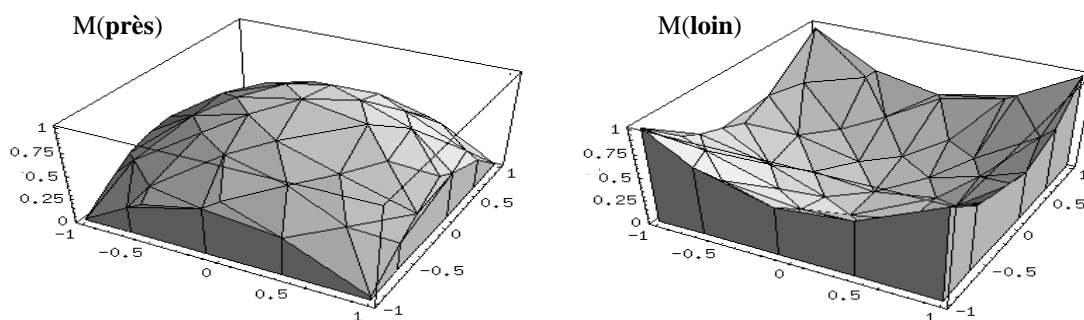


Fig. 66 : Système flou à entrées non décomposées.

La figure 67 présente le maillage de Delaunay de l'espace d'entrée. Chaque sommet des triangles correspond à un point appris, c'est-à-dire un point pour lequel le système flou engendre exactement la sortie désirée. Dans le cas présent, pour un ensemble d'apprentissage de 5000 points, l'erreur moyenne d'approximation est égale à 0.035. L'obtention des mêmes performances avec un système flou à entrées décomposées nécessite un partitionnement triangulaire en 12 symboles sur chaque entrée et la définition de 144 règles. On peut également noter que la déformation des significations n'entraîne aucune perte de lisibilité de la structure de commande globale.

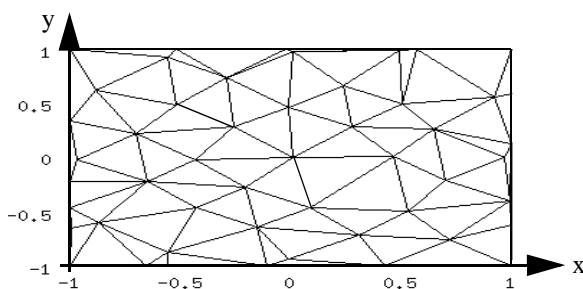


Fig. 67 Maillage de Delaunay de l'espace d'entrée.

La conception des capteurs flous a ici été abordée selon une approche interpolative dans un univers numérique multidimensionnel. La connaissance intégrée dans le capteur est une simple collection d'exemples numériques que l'on considère représentatifs de situations qualifiables linguistiquement. Le capteur agit en tant qu'interface numérique/symbolique. A ce titre, il s'intègre aisément dans une structure de contrôle qui peut être construite par expertise. Lorsque des données numériques entrées/sortie sont disponibles, le capteur peut être "recalibré" de façon à ce que la connaissance experte "colle" à la réalité numérique.

## VI.2.2 Agrégation par jeu de règles

La conception d'un capteur flou peut être abordée directement à partir d'une connaissance experte. En effet, d'un point de vue entrées/sortie, un capteur flou n'est rien d'autre qu'un système SF4 privé de son interface de défuzzification. A ce titre, il peut donc être directement développé à partir d'une base de règles.

Les capteurs présentés ici sont relatifs à la problématique du contrôle de niveau en lingotière de coulée continue. Le procédé est soumis à de nombreuses perturbations qu'il est intéressant de pouvoir détecter. Ainsi, deux capteurs spécifiques sont proposés, le premier visant à détecter les perturbations soudaines que représentent les débouchages de busette, le second étant dédié à la détection de phénomènes oscillatoires.

### VI.2.2.1 Capteur flou de débouchage

Il s'agit ici de concevoir un capteur capable de détecter l'occurrence d'un débouchage soudain de busette (cf paragraphe V.1). La méthode proposée repose sur l'observation des trajectoires de débouchage dans le plan de phase "erreur de niveau / variation d'erreur". La figure 68 illustre une telle trajectoire dans un cas réel de débouchage, l'erreur étant définie comme la différence entre le niveau mesuré et la consigne.

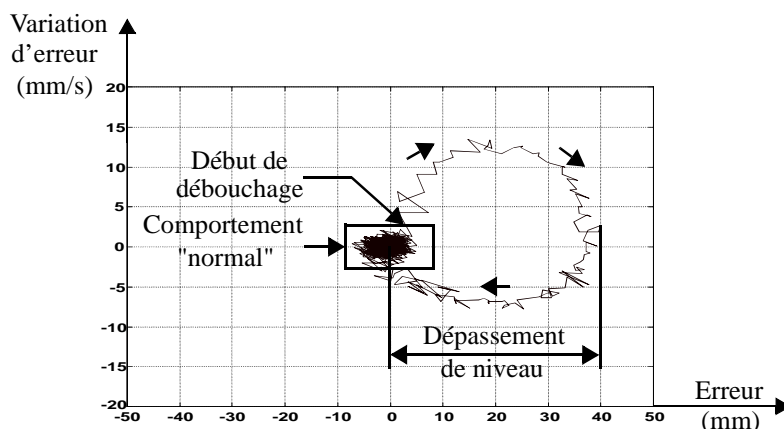


Fig. 68 : Trajectoire d'un débouchage.

Si le bruit de mesure induit des phénomènes instantanés aberrants, trois zones peuvent néanmoins être distinguées dans le plan de phase. La première, centrée sur l'origine, correspond à une zone non perturbée autour du point de fonctionnement. Les écarts par rapport à l'origine sont alors dus au bruit de mesure ou à de petites perturbations. La seconde zone est constituée du demi-plan gauche, hormis la zone centrale précédente. Un débouchage induit une augmentation importante de niveau et par conséquent une erreur positive, il ne peut donc se produire dans cette deuxième zone. Enfin, la troisième zone délimite la région possible pour une trajectoire de débouchage. Elle est complémentaire aux précédentes et comprend donc le demi-plan droit hormis la zone centrale non perturbée. Le capteur développé repose sur la définition de ces trois zones. A partir de l'erreur et de sa variation, il fournit une description symbolique floue de

la variable “Présence d’un débouchage”. Les partitions des univers de discours associés aux entrées et la base de règles utilisée pour l’inférence sont présentées à la figure 69. Il est à noter que le résultat délivré se présente sous la forme d’une partie floue de symboles du type  $0.2/Oui + 0.8/Non$ .

### VI.2.2.2 Capteur flou d’oscillations

Il s’agit maintenant de détecter la présence d’oscillations du niveau moyen d’acier en lingotière dont l’origine est liée à différents facteurs tels que, par exemple, des défauts de géométrie des rouleaux extracteurs ou encore des jeux et frottements dans le mécanisme de la quenouille. Pour un procédé et des conditions d’exploitation donnés, il s’avère que la période des oscillations pouvant apparaître est relativement bien connue. Si l’on note  $T_0$  cette dernière, le capteur doit fournir une information concernant l’existence d’oscillations de période  $T_0$ . Cette contrainte sur la période des oscillations permet de dissocier les véritables oscillations du niveau moyen, des vagues de surface pouvant induire des oscillations du niveau mesuré si un capteur ponctuel est utilisé.

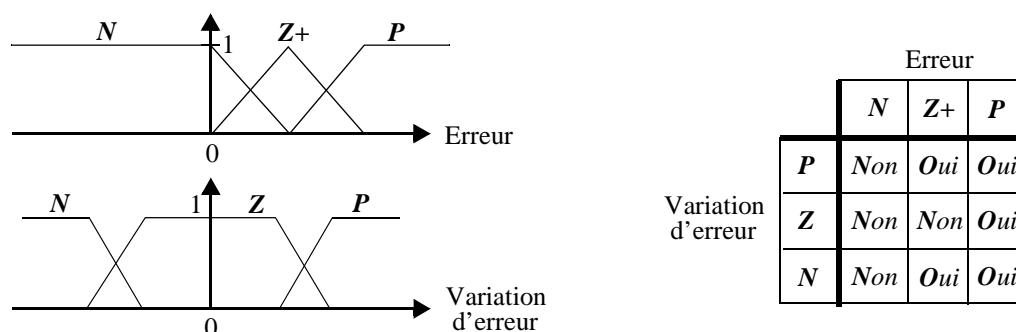


Fig. 69 : Implantation du capteur de débouchage.

Le phénomène recherché étant périodique, il est nécessaire d’étudier le signal d’erreur sur une fenêtre de temps glissante. Le traitement proposé se décompose alors en deux étapes. La première consiste à analyser le signal d’erreur et à produire amplitude ( $A_d$ ) et période ( $T_d$ ) des éventuelles oscillations détectées. Si aucun phénomène périodique n’est isolé sur la fenêtre temporelle considérée, les valeurs délivrées sont mises à zéro. La deuxième étape consiste à construire l’information symbolique “Présence d’oscillations de période  $T_0$ ”. Elle nécessite bien évidemment une comparaison de la période des oscillations détectées avec  $T_0$ . Quant à l’information d’amplitude  $A_d$ , elle est utilisée pour filtrer les oscillations de faible amplitude pouvant être dues à la présence de bruit sur le signal de niveau. Le mécanisme mis en oeuvre est donné à la figure 70 La base de règles illustre le fait que la décision finale de “Présence d’oscillations de période  $T_0$ ” n’est validée que si les oscillations détectées satisfont à la fois l’adéquation en période et le critère d’amplitude suffisante.

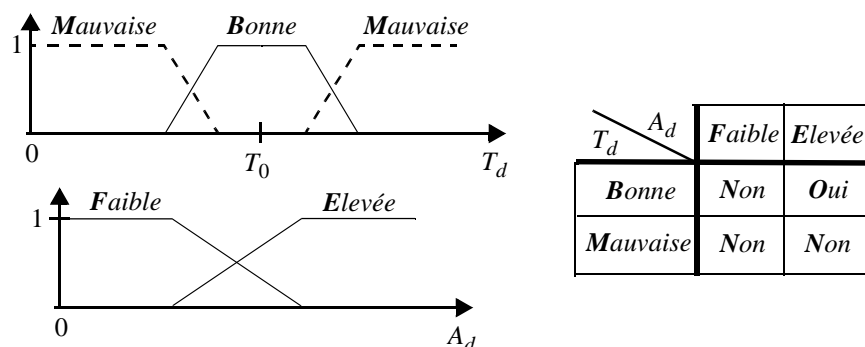


Fig. 70 Implantation du capteur d’oscillations.



## VI.3 Expertise et hiérarchisation

### VI.3.1 Supervision de lois de commande

Face à un procédé complexe, il est souvent délicat, voire impossible, de synthétiser une loi de commande unique prenant en compte toutes les situations pouvant être rencontrées. Par contre, moyennant certaines hypothèses ou simplifications sur le procédé, il est généralement possible de concevoir différents régulateurs, chacun d'entre eux répondant à un objectif de commande spécifique. L'idée développée ici est alors d'exploiter ces régulateurs "dédiés" pour traiter le problème global de commande. En d'autres termes, d'un point de vue conceptuel, la tâche est décomposée en sous-tâches élémentaires que l'on sait résoudre selon une stratégie de "diviser pour mieux régner". Le but de ce paragraphe n'est pas de traiter le problème de la décomposition en général (voir [Lacrose97] pour une analyse bibliographique sur le sujet) mais plutôt d'illustrer celle-ci sur un exemple.

Suivant le type de procédé étudié, différentes décompositions peuvent être envisagées. Dans le cas d'un système non linéaire, le point de fonctionnement peut fournir un critère de décomposition. On est alors amené à établir une famille de modèles linéarisés tangents, chacun d'entre eux permettant de synthétiser un régulateur dont le domaine d'utilisation est restreint au voisinage du point de fonctionnement considéré. Une autre approche de la décomposition repose sur une caractérisation des éventuelles perturbations affectant le procédé. Un régulateur spécifique peut alors être associé à chaque type de perturbation détectée, permettant ainsi de la rejeter au mieux.

L'étape de décomposition du problème global repose sur les deux compétences complémentaires que sont celle du "spécialiste procédé" et celle de l'automaticien. Le premier est en effet capable de déterminer les différents modes de fonctionnement du système ainsi que les perturbations pouvant l'affecter. Quant à l'automaticien, il est en mesure de déterminer et de régler un régulateur adapté à des contextes et à des objectifs particuliers.

Dans le cas de la coulée continue, le "spécialiste procédé" a isolé trois comportements différents, nécessitant un traitement adapté. Le premier correspond aux incidents de débouchage soudain de la busette, le deuxième à l'apparition d'oscillations périodiques du niveau moyen d'acier en lingotière, enfin le troisième à un mode de fonctionnement normal, c'est-à-dire sans perturbation notable du niveau de métal.

Le rôle de l'automaticien est d'élaborer, pour chaque comportement recensé, un régulateur permettant de traiter au mieux le problème correspondant. Dans le cas des débouchages ou oscillations, l'objectif du contrôle est de ramener le système le plus rapidement possible dans un mode de fonctionnement normal (rejet de perturbations), situation dans laquelle l'objectif devient alors un objectif de régulation. Dans la suite, on appellera :

*RD, un régulateur permettant de "rejeter" les débouchages de busette,*  
*RO, un régulateur permettant de limiter les oscillations du niveau d'acier, et*  
*RN, un régulateur permettant d'agir dans un contexte non perturbé.*

Diverses techniques de synthèse de ces contrôleurs sont détaillées dans [Dussud99].

L'objectif de la supervision étant de commuter entre les différents régulateurs disponibles, il est nécessaire d'élaborer des indicateurs permettant de qualifier le contexte courant. Dans notre cas, les capteurs de perturbations développés précédemment délivrent l'information d'existence des phénomènes considérés. Le choix du régulateur est ensuite effectué via un mécanisme décisionnel à base de règles. Enfin, la commande appliquée au procédé est calculée à partir des sorties délivrées par les régulateurs "bas niveau". La structure globale de la commande supervisée est résumée à la figure 71.

La stratégie de commutation entre régulateurs est élaborée à partir des règles suivantes, découlant de façon naturelle de la décomposition initiale :

Si débouchage = *Oui* alors régulateur = *RD*

Si oscillations = *Oui* alors régulateur = *RO*

Si débouchage = *Non* et oscillations = *Non* alors régulateur = *RN*

Dans les deux premières règles, les présences de débouchage ou d'oscillations sont considérées indépendamment l'une de l'autre, ce qui induit un conflit dans le choix du régulateur lorsque les deux phénomènes se produisent simultanément. Pour résoudre ce problème, il est tout simplement choisi de traiter en priorité la perturbation prépondérante.

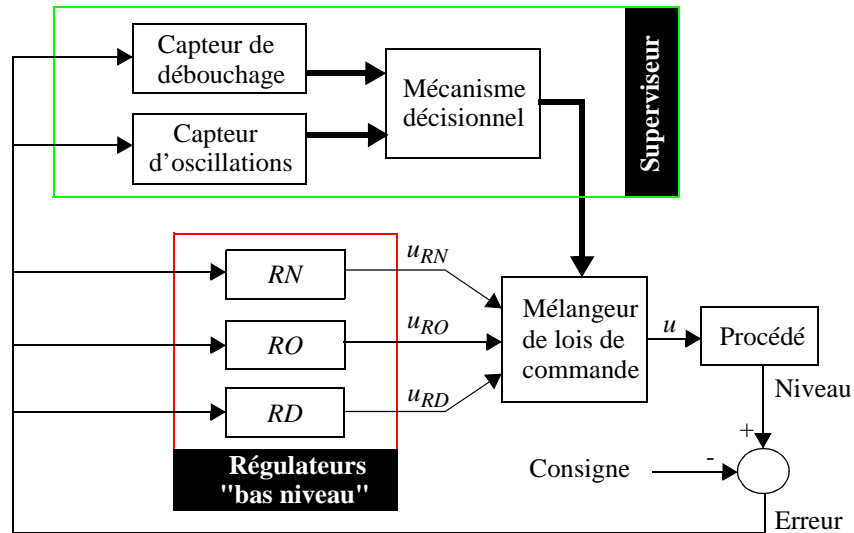


Fig. 71 : Structure supervisée pour la régulation de niveau en coulée continue.

L'amplitude des oscillations étant généralement bien inférieure aux variations de niveau occasionnées par un débouchage, le régulateur *RD* est choisi dès qu'un débouchage est détecté. La base de règles finale est alors donnée à la figure 72.

		Oscillations	
		<i>Oui</i>	<i>Non</i>
Débouchage	<i>Oui</i>	<i>RD</i>	<i>RD</i>
	<i>Non</i>	<i>RO</i>	<i>RN</i>

Fig. 72 : Stratégie de commutation.

Le résultat délivré par le mécanisme décisionnel est une partie floue de symboles du type  $0.2/RN + 0.1/RO + 0.7/RD$ . Reste maintenant à défuzzifier celui-ci de façon à déterminer la commande numérique à appliquer au procédé. Cette étape est réalisée via le mélangeur de lois de commande par une moyenne des sorties de chacun des régulateurs, pondérées par le degré de validité associé. Si l'on note respectivement  $u_{RN}$ ,  $u_{RO}$  et  $u_{RD}$  les sorties des régulateurs et  $\alpha_{RN}$ ,  $\alpha_{RO}$  et  $\alpha_{RD}$  les degrés de validité correspondants, la commande à appliquer est donnée par :

$$u = \frac{\alpha_{RN} \cdot u_{RN} + \alpha_{RO} \cdot u_{RO} + \alpha_{RD} \cdot u_{RD}}{\alpha_{RN} + \alpha_{RO} + \alpha_{RD}} \quad (110)$$

Dans la mesure où l'on utilise des partitions floues strictes et que les inférences mises en oeuvre

utilisent les opérateurs produit et somme bornée à 1, la somme des degrés de validité est toujours égale à 1. L'équation 110 se résume alors à :

$$u = \alpha_{RN} \cdot u_{RN} + \alpha_{RO} \cdot u_{RO} + \alpha_{RD} \cdot u_{RD}. \quad (111)$$

Le mélangeur de lois de commande peut être vu comme un défuzzificateur symbolique utilisant, pour partitionner l'univers de discours associé à  $u$ , trois singletons positionnés respectivement en  $u_{RN}$ ,  $u_{RO}$  et  $u_{RD}$ .

La structure supervisée de la figure 71 répond à l'attente initiale de la société SERT dans la mesure où elle permet de capitaliser le savoir-faire acquis au fil des années. Elle offre en outre un cadre d'accueil suffisamment flexible pour autoriser la prise en compte de régulateurs quelconques et faciliter l'ajout de nouvelles situations. A ce titre, l'aspect générique et modulaire, introduit par la supervision, est particulièrement intéressant pour l'entreprise. Reste à souligner que la démarche développée n'est spécifique, ni à la coulée continue, ni à la sidérurgie, ni même à la supervision de boucles de commande. Elle est en fait applicable à toute problématique dans laquelle la connaissance experte peut être répartie dans différents composants. Le chaînage de blocs symboliques flous permet de véhiculer des informations linguistiques tout en préservant le caractère graduel de l'information. Cette caractéristique est à la base de la distribution de l'expertise dans différents blocs. Elle est d'ailleurs au centre de la hiérarchisation de la connaissance experte telle qu'abordée dans le paragraphe suivant.

### VI.3.2 Hiérarchisation d'une connaissance experte

La hiérarchisation d'une connaissance experte permet d'appréhender des problèmes relativement complexes par une analyse descendante. La méthodologie mise en oeuvre consiste simplement à structurer la connaissance globale en termes de connaissances relatives à des domaines spécifiques, elles mêmes décomposées par restrictions successives des domaines concernés. Au niveau le plus bas, la connaissance est généralement exprimable à partir d'un nombre restreint de variables numériques. Bien que cette approche hiérarchisante ait un caractère générique et puisse donc être exploitée pour différents problèmes, elle est difficilement abordable en dehors de tout contexte. La démarche est ici illustrée sur une application spécifique, développée dans le cadre d'un contrat avec Renault Recherche. La problématique étudiée s'inscrit dans un projet d'aide à la conduite, et plus particulièrement d'aide à la gestion d'une boîte de vitesse ([Bonnetain81], [Ueda84]).

Sur des véhicules équipés de boîte de vitesses manuelle, une utilisation efficace des rapports permet de réduire la consommation moyenne sans dégrader les performances globales. Pour aider le conducteur dans son choix de rapport, il est nécessaire d'estimer l'efficacité d'un rapport. De manière instantanée, un rapport peut être qualifié d'efficace s'il permet un fonctionnement du moteur dans sa plage de meilleur rendement. Cependant, la proposition d'un rapport jugé instantanément efficace peut s'avérer totalement inappropriée si la décision est prise sans aucune considération du délai induit par le changement de rapport. En effet, un rapport jugé efficace à l'instant courant (instant  $t$ ) peut devenir totalement inadapté au moment où il est effectivement enclenché (instant  $t+TE$ ). Le retard  $TE$  engendré par l'actionneur que représente l'opérateur humain couplé avec le système mécanique est donc un paramètre essentiel qui doit être pris en compte dans la stratégie de changement de rapport. Dans ce contexte, un mécanisme de prédiction du temps d'exécution d'un changement de rapport ( $TE$ ) est proposé. Celui-ci est défini comme étant le temps écoulé entre le début de l'action du conducteur et l'instant où le rapport est effectivement enclenché. S'il peut être déterminé a posteriori à partir de signaux mesurés en ligne (régime en sortie de boîte, régime moteur, ...), sa prédiction reste un problème complexe, ignoré dans [Ueda84] et traité via des valeurs pré-déterminées dans [Bonnetain81]. Dans la présente étude, la prédiction de  $TE$  est abordée à partir de la connais-

sance experte du domaine.

Dans une première étape, le temps d'exécution que l'on cherche à prédire est décomposé en deux contributions d'origine différente, appelées respectivement temps minimal et temps additif. D'un point de vue interprétation, le temps minimal correspond à un délai incompressible dû aux contraintes mécaniques de la boîte de vitesse. Il est donc indépendant du contexte de conduite et du chauffeur. Quant au temps additif, il reflète essentiellement la façon dont le conducteur gère le changement de rapport. Il est bien évidemment lié au conducteur mais aussi à la situation de conduite dans laquelle se produit le changement de rapport.

Cette décomposition additive du temps d'exécution permet de restreindre la prédiction à celle du temps additif, le temps minimal pouvant être déterminé a priori par une caractérisation de la boîte de vitesse. La structure globale proposée est illustrée à la figure 73. Elle comporte deux blocs indépendants chargés respectivement de prédire le temps additif et de déterminer le temps minimal. La prédiction du temps additif est basée sur un ensemble de variables contextuelles. Par contre, seul le changement de rapport envisagé est nécessaire à la détermination du temps minimal.

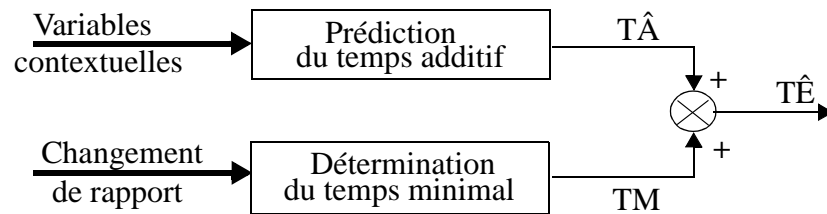


Fig. 73 : Architecture globale du système de prédiction.

Le mécanisme de prédiction du temps additif a été développé à partir d'un recueil d'expertise, réalisé sur site industriel par un intervenant extérieur à l'entreprise (étudiant post DEA). Outre le fait qu'elle repose sur des systèmes SF4, l'architecture proposée présente les caractéristiques suivantes.

- Le système est implanté selon une structure modulaire et hiérarchisée à deux étages. Les blocs du premier niveau ont pour objectif de construire, à partir de données numériques, une représentation symbolique floue de différents concepts caractérisant la situation. A ce titre, ils sont assimilables à des capteurs flous. Au niveau haut, on trouve un bloc décisionnel délivrant la prédiction du temps additif.
- Les blocs du premier étage contiennent un module d'inférence symbolique ( $\rho_4$  dans la typologie des systèmes flous) dont les entrées et sortie sont des sous-ensembles flous symboliques. Les sorties de ces blocs étant les entrées du bloc de décision, le transfert d'informations est direct. **Le chaînage des bases de règles est alors immédiat et ne nécessite aucun retour à un univers numérique factice.**
- Une interface NS est intégrée dans les blocs de bas niveau afin de convertir les données numériques en sous-ensembles flous symboliques avant de les transmettre au module de raisonnement. A l'inverse, le bloc décisionnel réalise une conversion SN de façon à produire une prédiction numérique du temps additif.

D'un point de vue pratique et selon la figure 74, le contexte du changement de rapport est décrit selon trois facettes distinctes. La première concerne le profil de la mission en cours (PM). Il est déterminé à partir de la vitesse instantanée du véhicule (VV) et du nombre de changements de

rappports effectués dans un intervalle de temps fixé précédant la prédiction (NBRAP). La seconde facette est plus particulièrement associée au chauffeur puisqu'il s'agit de caractériser son dynamisme de conduite (DC). Celui-ci est évalué à partir de la dérivée de la course de la pédale d'accélération (DA) et du temps additif moyen réalisé sur les derniers changements de rapport effectués (TAM). L'accès à cette grandeur nécessite de réaliser en différé le calcul du temps d'exécution d'un changement de rapport et de lui soustraire le temps minimal associé. On obtient ainsi le temps additif réel. Une moyenne glissante peut alors être réalisée sur les derniers changements de rapports effectués. Enfin, le contexte d'un changement de rapport peut également être caractérisé d'un point de vue de la "charge" du véhicule (CH). La notion de "charge" est définie ici comme l'effet combiné de la masse du véhicule (M) et de l'effort résistant qui lui est appliqué (EFRES). La décision finale concernant le temps additif associé à un changement de rapport est prise au regard des trois grandeurs caractéristiques évaluées au niveau bas de la structure.

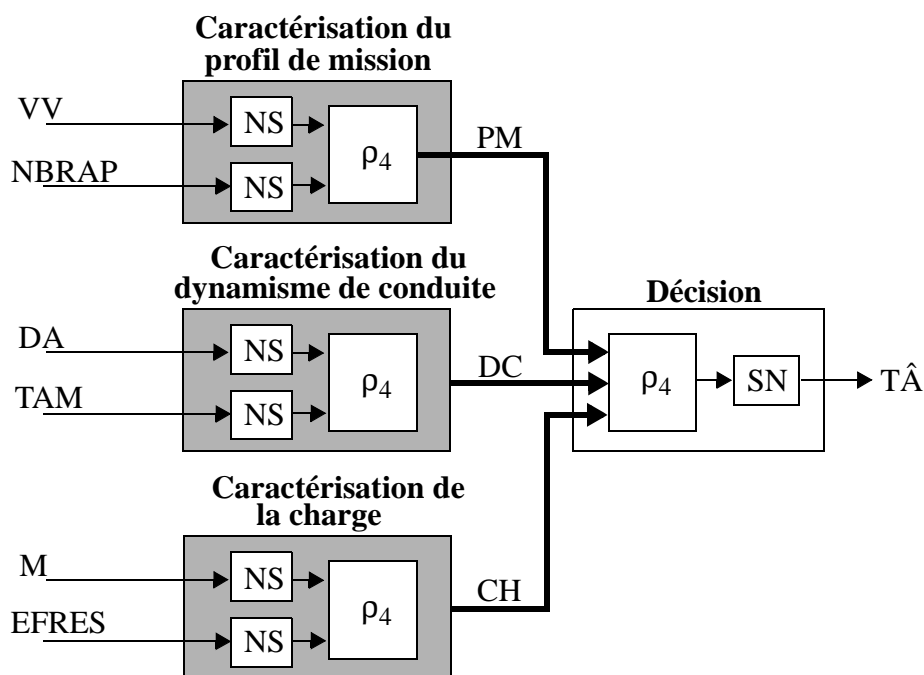


Fig. 74 : Prédiction du temps additif.

A titre d'exemple, la figure 75 décrit la mise en oeuvre du bloc de caractérisation du profil de mission PM, qualifié à l'aide des termes NATionale, AUToroute et TRANSitoire.

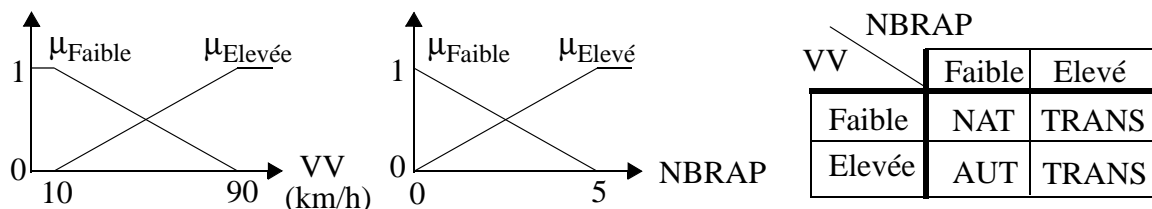


Fig. 75 : Caractérisation du profil de mission

Le temps additif étant prédit, il ne reste plus qu'à lui ajouter le temps minimal pour obtenir la prédiction finale du temps d'exécution du passage de rapport (TÊ) (cf figure 73). La détermination du temps minimal TM dépend du changement de rapport considéré. Celui-ci est spécifié

sous la forme d'un couple regroupant rapport initial et rapport final, noté  $(N_i, N_f)$ . La caractérisation de la boîte de vitesse utilisée consiste alors à déterminer pour chaque couple  $(N_i, N_f)$  le temps de passage minimal associé. Elle aboutit à l'obtention d'une matrice de valeurs prédéterminées estimées à partir d'essais réalisés, soit sur banc, soit sur circuit.

Les travaux présentés ont fait l'objet d'une demande de dépôt de brevet intitulé "Module de prédiction des temps de passage de rapports d'un véhicule industriel" auprès de la direction de la recherche de Renault.

Ce paragraphe nous a permis d'illustrer, à partir d'applications réelles, différentes possibilités de distribution de l'expertise au sein d'une architecture globale. Si la majorité des travaux est clairement associée à une problématique de commande, il n'en demeure pas moins que la démarche envisagée n'est pas dédiée au contrôle, ainsi qu'en atteste d'ailleurs la dernière étude présentée.

Il est maintenant important d'extraire quelques éléments clés permettant de positionner nos travaux dans un contexte général. La première remarque que nous pouvons faire concerne l'aspect générique de l'intégration d'expertise au sein d'un composant flou. En effet, quelle que soit la fonction de celui-ci (capteur, contrôleur, organe de décision, ...), l'introduction de la connaissance experte relève du même mécanisme qui peut prendre deux formes distinctes, à savoir :

- écriture directe des règles,
- utilisation d'exemples.

Notons que la deuxième situation regroupe à la fois l'approche interpolative mise en oeuvre dans les capteurs (cf paragraphe VI.2.1) et l'exploitation du principe des équivalences modales préconisé pour la construction des contrôleurs flous (cf chapitre I et paragraphe V.2). Dans les deux cas, il est notamment possible d'aider l'expert en utilisant une technique d'apprentissage incrémentale pour sélectionner les exemples les plus pertinents au vue de l'interpolation produite. L'unique différence réside en fait dans la nature des exemples considérés qui doivent être répartis sur un maillage rectangulaire lorsque l'on cherche à générer des règles.

Cette première constatation nous conduit tout naturellement à revenir à la méthodologie de conception élaborée pour les contrôleurs flous (cf paragraphe V.3) et à l'appliquer dans le contexte plus général de développement d'une architecture distribuée. Chaque composant flou peut être traité de manière similaire en ce qui concerne l'intégration d'expertise. Reste alors à introduire la notion de distribution, voire de hiérarchisation, pour aboutir à une méthodologie générale. A nos yeux, cette dernière peut être appréhendée en itérant les étapes ②, ③ et ④ selon une analyse descendante de façon à décomposer la tâche globale à exécuter en un ensemble de sous-tâches élémentaires que l'on sait résoudre.

## VII Conclusion

L'ensemble des travaux présentés dans ce chapitre repose sur une implantation SF4 des systèmes flous. Il nous semble donc tout naturel de revenir ici sur les propriétés fondamentales de ces derniers qui sont à la base des développements proposés.

D'un point de vue pratique, les systèmes SF4 se présentent sous la forme d'une collection de règles à la Mamdani, à la différence près qu'un poids est attribué à chacune d'entre elles. Cette caractéristique leur confère lisibilité et pouvoir explicatif, éléments essentiels à la mise en place d'une démarche experte. En effet, si un expert est à même de comprendre et d'analyser un ensemble de règles, on peut espérer qu'il soit également

capable de le produire et ainsi envisager un codage d'expertise.

Sous l'hypothèse d'un partitionnement triangulaire strict des univers de discours d'entrée, les systèmes SF4 permettent de représenter de manière exacte un système de Sugeno à conclusions constantes. Cette propriété permet d'établir un lien fort avec les travaux du chapitre I. Combinée à l'aspect linguistique évoqué ci-dessus, elle autorise en outre l'intégration dans un formalisme unique de connaissances de nature diverse (lois de commande numériques, données mesurées, expertise).

L'implantation SF4 permet un découplage complet des tâches de conversion et de raisonnement. Cette caractéristique permet d'aborder une distribution des traitements mais aussi de l'expertise. Elle ouvre ainsi la voie au développement d'architectures nouvelles dans lesquelles il devient possible de manipuler des grandeurs subjectives et de chaîner des blocs de nature différente (détection, prise de décision, supervision, filtrage, commande, ...).

Dans ce contexte, l'implantation SF4 des systèmes flous nous semblent répondre à des préoccupations actuelles de la communauté floue. Elle offre notamment la possibilité de "calculer avec les mots" tout en garantissant, via des interfaces de conversion, une cohérence avec le calcul numérique. Il n'en demeure pas moins que de nombreux efforts sont encore à fournir avant de pouvoir prétendre au développement d'une "théorie des systèmes linguistiques".

A nos yeux, une des faiblesses de l'approche SF4 actuelle réside dans la pauvreté d'expression des règles exploitées, due essentiellement à la difficulté d'implantation d'une sémantique particulière de règles. Pour améliorer le codage d'expertise, il est important d'étudier en profondeur la possibilité d'implanter, via une approche SF4, des règles conjonctives mais aussi des règles implicatives. Nous avons déjà entrepris une première réflexion sur les règles à certitude (cf paragraphe III), l'étude des règles graduelles est maintenant une de nos perspectives prioritaires. De manière plus générale, il nous semble aujourd'hui primordial de situer nos travaux dans le cadre du raisonnement. En effet, ce n'est sans doute qu'au prix d'une compréhension fine des mécanismes associés au raisonnement que nous pourrions envisager leur usage potentiel en commande.

Un autre type de perspective tout aussi attrayant concerne la caractérisation, ou tout au moins la détermination de propriétés élémentaires, d'un système SF4 dont on connaît la base de règles. Si l'on voit une base de règles comme une fonction de transfert ou plus exactement comme une classe de fonctions de transfert, il est en effet probablement possible d'en déduire certaines caractéristiques du système correspondant, sous l'hypothèse qu'un ordre soit établi sur les mots utilisés. Il est sans doute également possible d'établir des contraintes sur les interfaces de conversion pour le système possède une propriété donnée. On imagine aisément pouvoir ainsi aborder les notions de monotonie ou encore de linéarité à un niveau linguistique. Reste un pas à franchir pour appréhender le concept de stabilité.





## Conclusion générale

Ce manuscrit nous a permis d'aborder les aspects numérique mais aussi linguistique de la commande floue. Dans chacun de ces domaines, nous avons tenté d'apporter une contribution originale. Dans un cadre numérique, celle-ci prend la forme du principe des équivalences modales alors que d'un point de vue linguistique, elle se concrétise par une implantation purement symbolique des systèmes flous. Nous avons eu l'occasion de discuter en détails des différentes applications qui pouvaient découler d'une exploitation conjointe de ces deux approches. Des perspectives à nos travaux ont été évoquées en conclusion des deux chapitres de ce mémoire, aussi, ne nous paraît-il pas opportun de revenir ici sur les aspects scientifiques ou techniques des outils proposés, mais plutôt de réfléchir à la démarche que nous avons développée.

Le choix que nous avons fait pour présenter nos travaux a été de mettre en évidence l'intérêt des techniques floues pour la commande des processus en insistant sur la dualité entre représentation numérique et représentation symbolique de l'information. Il y a bientôt trente ans, Zadeh s'interrogeait déjà sur cette dualité et écrivait ([Zadeh72]) :

*“I also believe that, in order to provide a hospitable environment for the development of fuzzy algorithms, control theory must become less preoccupied with mathematical rigor and precision, and more concerned with the development of qualitative or approximate solutions to pressing real world problems. Such a theory may well turn out to be far richer and far more exciting than control theory is today.”*

Pour notre part, nous sommes convaincus qu'une réelle avancée en terme d'automatisation des systèmes industriels ne peut être obtenue qu'en optimisant le fonctionnement global d'une application et non plus chacun de ses constituants pris individuellement. Dans cette optique, il nous semble nécessaire d'évoluer vers une approche de la commande de haut niveau, regroupant les aspects liés à la fusion d'informations, le diagnostic, la décision et la supervision et permettant d'intégrer l'homme dans le schéma global d'automatisation. De notre point de vue, les techniques floues sont particulièrement prometteuses dans ce contexte puisqu'elles permettent d'intégrer, dans un formalisme unique, des informations numériques mais aussi des connaissances expertes.

Une autre clé de lecture de nos travaux pourrait aussi être la comparaison entre la conception de lois de commande floues à partir de modèles et celle reposant sur des heuristiques, dont la finalité serait alors de répondre à la question du choix de l'une ou l'autre de ces deux stratégies.

A travers les différentes applications que nous avons développées, nous avons pu montrer qu'elles permettraient toutes deux d'aboutir à des résultats satisfaisants. Cependant, cela ne signifie pas pour autant que les solutions retenues soient uniques. Ainsi, la commande d'axe de robot, utilisée dans ce document pour illustrer la démarche orientée modèle, aurait probable-

ment pu être abordée selon une approche experte puis calibrée à partir d'un protocole d'essais/erreurs. A l'inverse, la régulation de niveau d'acier en lingotière de coulée continue, traitée ici par un codage direct d'expertise, aurait sans doute pu être appréhendée à partir d'un modèle fin du procédé qu'il aurait alors fallu identifier dans une phase préliminaire.

En dehors de tout contexte, il nous paraît clair qu'aucune réponse systématique ne peut être avancée. Il nous semble cependant que nous pouvons tirer de nos travaux une orientation générale. Ainsi, lorsqu'un modèle du procédé a été identifié, la démarche naturelle consiste à l'exploiter pour synthétiser la loi de commande. Cette approche est d'ailleurs d'autant plus justifiée que l'objectif de commande est spécifié de manière quantitative et qu'aucune interaction homme/machine n'est envisagée. La majorité des publications vont actuellement dans ce sens, mais conduisent parfois à s'interroger sur la signification de la classification comme contrôleurs flous des structures de commande obtenues. Nous pensons en particulier aux approches de type PDC et leurs variantes qui permettent certes des preuves de stabilité mais pour lesquelles les "fonctions d'appartenance" rendent l'interprétation difficile, pour ne pas dire impossible.

D'un autre côté, en l'absence de modèle numérique et dans la mesure où une connaissance experte peut être formulée, l'approche symbolique permet de s'affranchir des coûts que représente l'étape préliminaire d'une modélisation. Cette solution se fait cependant au détriment d'une garantie de performances puisqu'il devient impossible d'assurer la stabilité ou la robustesse du système bouclé autrement que par une vérification expérimentale qui n'a évidemment pas valeur de preuve. Dans ces conditions, en prenant en compte le coût associé à l'obtention d'un modèle, **le choix d'une synthèse experte relève plus d'une stratégie d'entreprise que d'un réel problème scientifique**. Il nous semble donc que le débat sur les avantages ou les inconvénients de la commande floue ne devrait plus être placé à un niveau purement scientifique, comme cela a parfois été tenté de manière un peu polémique, en raison de son intérêt très limité tant qu'il ne sera pas possible d'intégrer des critères économiques, ayant parfois une dimension marketing, dans des comparaisons objectives.

## Références

- [Babuska95] R. Babuska, J. Sousa, H.B. Verbruggen, “Model-based design of fuzzy control systems”, Proc. of the 3rd European Congress on Intelligent Techniques and Soft Computing (EUFIT’95), Aachen, Allemagne, pp. 837-841, 95.
- [Babuska96] R. Babuska, M. Setnes, U. Kaymak, H.R. Van Nauta Lemke, “Rule base simplification with similarity measures”, Proc. of the 5th IEEE Int. Conf. on Fuzzy Systems (FUZZ’IEEE96), New Orleans, USA, pp. 1642-1647, Sept. 96.
- [Baranyi96] P. Baranyi, P. Korondi, H. Hashimoto, Z. Suto, “Learning fuzzy controller based on linguistic model inversion”, Proc. of the 7th Int. Power Electronics & Motion Control Conf. (PEMC’96), Budapest, Hongrie, pp. 467-471, 96.
- [Baranyi97] P. Baranyi, I. Bavelaar, L.T. Koczy, A. Titli, “Inverse rule base of various fuzzy interpolation techniques”, Proc. of the 7th IFSA World Congress, Prague, Tchéquie, pp. 121-126, 97.
- [Benoit93] E. Benoit, “Capteurs symboliques et capteurs flous : un nouveau pas vers l’intelligence”, Thèse de doctorat, Université de Grenoble, Janvier 93.
- [Bikdash99] M. Bikdash, “A highly interpretable form of Sugeno inference systems”, IEEE Trans. on Fuzzy Systems, vol. 7, no. 6, pp. 686-696, Dec. 99.
- [Bonnetain81] Y. Bonnetain, J.P. Sibeud, “Road transportation vehicle drive assist process and apparatus”, US Patent 4,463,427, Renault Véhicules Industriels, July 81.
- [Bouchon93] B. Bouchon-Meunier, *La logique floue*, Collection Que sais-je, no 2702, Presses Universitaires de France, 93.
- [Boukezzoula97] R. Boukezzoula, S. Galichet, L. Foulloy, “Fuzzy Input-Output Linearization of Nonlinear Systems”, Proc. of the 5th European Congress on Intelligent Techniques and Soft Computing (EUFIT’97), Aachen, Germany, Vol. 2, pp. 1316-1320, Sept. 97.
- [Boukezzoula98] R. Boukezzoula, S. Galichet, L. Foulloy, “Fuzzy Learning Control of Nonlinear Systems Using Input-Output Linearization”, Proc. of the 1998 IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC’98), San Diego, California, USA, pp. 2095-2100, Oct. 98.
- [Boukezzoula00] R. Boukezzoula, “Commande floue d’une classe de systèmes non linéaires : application au problème de suivi de trajectoire”, Thèse de doctorat, Université de Savoie, Mars 2000.

- [Boukezzoula01] R. Boukezzoula, S. Galichet, L. Foulloy, "Robust fuzzy control for a class of continuous nonlinear systems using input-output linearization: Real-time implementation for a robot wrist", Proc. of the 2001 IEEE Int. Conf. on Control Applications (CCA2001), Mexico, Mexique, Sept. 2001, pp 311-316.
- [Boukezzoula02] R. Boukezzoula, S. Galichet, L. Foulloy, "Fuzzy Nonlinear Adaptive Internal Model Control (FNAIMC)", European Journal of Control, à paraître.
- [Bouzlama92] F. Bouzlama, A. Ichikawa, "Fuzzy control rules and their natural control laws", Fuzzy Sets and Systems, vol. 48, pp. 65-86, 1992.
- [Buckley89] J.J Buckley and H. Ying, "Fuzzy Controller Theory : Limit Theorems for Linear Fuzzy Control Rules", Automatica, vol. 25, pp. 469-472, 1989.
- [Buckley92] J.J Buckley, "Universal fuzzy controllers", Automatica, vol. 28, no. 6, pp. 1245-1248, 1992.
- [Buckley93] J.J Buckley, "Sugeno type controllers are universal controllers", Fuzzy Sets and Systems, vol. 53, pp. 299-303, 1993.
- [Chao96] C.T. Chao, Y.J. Chen, C.C. Teng, "Simplification of fuzzy-neural systems using similarity analysis", IEEE Trans. on Systems, Man, and Cybernetics, Part B, vol. 26, no. 2, pp. 344-354, April 96.
- [Charbonnier00] S. Charbonnier, S. Galichet, G. Mauris, J.P. Siché, "Statistical and Fuzzy Models of Ambulatory Systolic Blood Pressure for Hypertension Diagnosis", IEEE Trans. on Instrumentation and Measurement, Vol. 49, N° 5, October 2000, pp. 998-1003.
- [Cordon99] O. Cordon, F. Herrera, "A two-stage evolutionary process for designing TSK fuzzy rule-based systems", IEEE Trans. on Systems, Man, and Cybernetics, Part B, vol. 29, no. 6, pp. 703-715, Dec. 99.
- [Cordon00] O. Cordon, F. Herrera, "A proposal for improving the accuracy of linguistic modeling", IEEE Trans. on Fuzzy Systems, vol. 8, no. 3, pp. 335-344, June 2000.
- [deOliveira95] J.V. de Oliveira, "Design methodology for fuzzy system interfaces", IEEE Trans. on Fuzzy Systems, vol. 3, no. 4, pp. 404-414, Nov. 95.
- [deOliveira99] J.V. de Oliveira, "Semantic constraints for membership function optimization", IEEE Trans. on Systems, Man, and Cybernetics, Part A, vol. 29, no. 1, pp. 128-138, Jan. 99.
- [Dickerson96] J.A. Dickerson, B. Kosko, "Fuzzy function approximation with ellipsoidal rules", IEEE Trans. on Systems, Man, and Cybernetics, Part B, vol. 26, no. 4, pp. 542-560, Aug. 96.

- [Ding00] Y. Ding, H. Ying, "Necessary conditions on minimal system configuration for general MISO Mamdani fuzzy systems as universal approximators", *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, vol. 30, no. 6, pp. 857-863, Dec. 2000.
- [Dubois87] D. Dubois, H. Prade, *Théorie des possibilités*, Masson, 2ième édition, 87.
- [Dubois97] D. Dubois, L. Foulloy, S. Galichet, H. Prade, "Two different views of approximate reasoning", *Proc. of the 7th Int. Fuzzy Systems Association World Congress (IFSA'97)*, Prague, Czech Republic, June 97, Vol. 1, pp. 238-242.
- [Dubois99] D. Dubois, L. Foulloy, S. Galichet, H. Prade, "Performing approximate reasoning with words?", in *Computing with Words in Information/Intelligent Systems 1 - Foundations*, (L.A. Zadeh, J. Kacprzyk Eds), Physica-Verlag, 1999, pp. 24-49.
- [Dussud96] M. Dussud, S. Galichet, L. Foulloy, "Design and implementation of a fuzzy-logic controller for brass casting control", *Proc. of the 4th European Congress on Intelligent Techniques and Soft Computing (EUFIT'96)*, Aachen, Germany, pp. 1230-1236, Sept. 96.
- [Dussud98] M. Dussud, S. Galichet, L. Foulloy, "Application of fuzzy logic control for continuous casting mold level control", *IEEE Trans. on Control Systems Technology*, vol. 6, no. 2, pp. 246-256, March 98.
- [Dussud99] M. Dussud, "Régulation de niveau en lingotière de coulée continue", Thèse de doctorat, Université de Savoie, Nov. 99.
- [El Hajjaji94] A. El Hajjaji, A. Rachid, "Explicit formulas for fuzzy controller", *Fuzzy Sets and Systems*, vol. 62, pp. 135-141, 1994.
- [El Hajjaji95] A. El Hajjaji, A. Rachid, "Analytic formulation of linguistic rules for fuzzy controller", *Fuzzy Sets and Systems*, vol. 73, pp. 219-225, 1995.
- [Evsukoff02] A. Evsukoff, A.C.S. Branco, S. Galichet, "Structure identification and parameter optimization for non-linear fuzzy modeling", *Fuzzy Sets and Systems*, accepté avec révision mineure, 2002.
- [Farag98] W.A. Farag, V.H. Quintana, G. Lambert-Torres, "A genetic-based neuro-fuzzy approach for modeling and control of dynamical systems", *IEEE Trans. on Neural Networks*, vol. 9, no. 5, pp. 756-767, Sept. 98.
- [Foulloy94a] L. Foulloy, S. Galichet, "Fuzzy Sensor for Fuzzy Control", *Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 2, no. 1, 1994, pp. 55-66.
- [Foulloy94b] Foulloy L., Galichet S., E. Benoit, "Fuzzy control with fuzzy state sensors", *Proc. of the 2nd European Congress on Fuzzy Intelligent*

- Technologies (EUFIT 94), Aachen, Germany, pp. 1156-1160, Sept. 94.
- [Foulloy95a] L. Foulloy, S. Galichet, "Typology of Fuzzy Controllers", in *Theoretical Aspects of Fuzzy Control* (H.T. Nguyen, M. Sugeno, R. Tong, and R. Yager Eds), John Wiley & Sons, 1995, pp. 65-90.
- [Foulloy95b] L. Foulloy, S. Galichet, "Representations of Mamdani's Controllers", *Proc. of the 6th Int. Fuzzy System Association World Congress (IFSA 95)*, Sao Paulo, Brasil, July 95, pp. 581-584.
- [Foulloy96] L. Foulloy, S. Galichet, "Evaluation des performances de contrôleurs flous à partir d'un benchmark", *RAIRO-APII-JESA*, Vol. 30, no. 5, pp. 585-605, 1996.
- [Foulloy98] L. Foulloy, S. Galichet, "Fuzzy and Linear Controllers", in *FUZZY SYSTEMS Modeling and Control* (H.T. Nguyen, M. Sugeno Eds), *The Handbooks of Fuzzy Sets Series*, Kluwer Academic Publishers, pp. 197-225, 1998.
- [Galichet92] S. Galichet, M. Dussud, L. Foulloy, "Contrôleurs flous : équivalences et études comparatives", *Actes des 2ièmes Journées Nationales sur les Applications des Ensembles Flous*, Nimes, France, pp. 229-236, Nov. 1992.
- [Galichet93] S. Galichet, L. Foulloy, "Fuzzy Equivalence of Classical Controllers", *Proc. of the 1st European Congress on Fuzzy Intelligent Technologies (EUFIT 93)*, Aachen, Germany, pp. 1567-1573, Septembre 1993.
- [Galichet94a] Galichet S., Foulloy L., Chebre M., Beauchene J., "Fuzzy Logic Control of a Floating Level in a Refinery Tank", *Proc. of the 3rd IEEE Conf. on Fuzzy Systems (FUZZ-IEEE '94)*, Orlando, USA, pp. 1538-1542, June 1994.
- [Galichet94b] Galichet S., Foulloy L., "State Feedback Fuzzy Controllers", *Proc. of the 2nd European Congress on Fuzzy Intelligent Technologies (EUFIT 94)*, Aachen, Germany, pp. 1161-1167, Sept. 94.
- [Galichet95] S. Galichet, L. Foulloy, "Fuzzy Controllers: Synthesis and Equivalences", *IEEE Trans. on Fuzzy Systems*, vol. 3, no. 2, pp. 140-148, 1995.
- [Galichet97] S. Galichet, D. Passaquay, L. Foulloy, "Rulebase structure identification or reduction in the framework of fuzzy learning", *Proc. of the 7th IFSA World Congress*, Prague, pp. 363-368, 1997.
- [Galichet98] S. Galichet, L. Foulloy, "Size reduction in fuzzy rulebases", *Proc. of the IEEE Int. Conf. on Systems, Man and Cybernetics (SMC'98)*, San Diego, USA, pp. 2107-2112, Oct. 98.
- [Galichet00] S. Galichet, S. Charbonnier, G. Mauris, J.P. Siché, "A Fuzzy Linguistic

Model for Ambulatory Systolic Blood Pressure Variation Prediction”, Proc. of the IEEE Int. Conf. on Fuzzy Systems (FUZZ’IEEE 2000), San Antonio (TX), USA, pp. 522-527, May 2000.

- [Galichet01] S. Galichet, “Expression analytique unique de la sortie d’un contrôleur flou de Mamdani”, Rapport interne LAMII 01-02, Université de Savoie, Octobre 2001.
- [Guély94] F. Guély, P. Siarry, “A centred formulation of Takagi-Sugeno rules for improved learning efficiency”, Fuzzy Sets and Systems, vol. 62, pp. 277-285, 1994.
- [Herrera00] F. Herrera, L. Martinez, “A 2-tuple fuzzy linguistic representation model for computing with words”, IEEE Trans. on Fuzzy Systems, vol. 8, no. 6, pp. 746-752, April 2000.
- [Higgins94] C.M. Higgins, R.M. Goodman, “Fuzzy rule-based networks for control”, IEEE Trans. on Fuzzy Systems, vol. 2, no. 1, pp. 82-88, Feb. 94.
- [Hu99] B. Hu, G.K.I. Mann, R.G. Gosine, “New methodology for analytical and optimal design of fuzzy PID controllers”, IEEE Trans. on Fuzzy Systems, vol. 7, no. 5, pp. 521-539, April 2000.
- [Ishibuchi94] H. Ishibuchi, K. Nozaki, H. Tanaka, Y. Hosaka, M. Matsuda, “Empirical study on learning in fuzzy systems by rice taste analysis”, Fuzzy Sets and Systems, vol. 64, pp. 129-144, 94.
- [Jang93] J-S. R. Jang, “ANFIS: Adaptive-network-based fuzzy inference system”, IEEE Trans. on Systems, Man, and Cybernetics, vol. 23, no. 3, pp. 665-684, May 93.
- [Jenkins94] M.S. Jenkins, B.G. Thomas, W.C. Chen, R.B. Mahapatra, “Investigation of strand surface defects using mold instrumentation and modeling”, Iron and Steel Society, Warrendale, PA, 94.
- [Jin00] Y. Jin, “Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement”, IEEE Trans. on Fuzzy Systems, vol. 8, no. 2, pp. 212-221, April 2000.
- [Kim97] E. Kim, M. Park, S. Ji, M. Park, “A new approach to fuzzy modeling”, IEEE Trans. on Fuzzy Systems, vol. 5, no. 3, pp. 328-337, Aug. 97.
- [Koczy97] L.T. Koczy, K. Hirota, “Size reduction by interpolation in fuzzy rule bases”, IEEE Trans. on Systems, Man, and Cybernetics, Part B, vol. 27, no. 1, pp. 14-25, Feb. 97.
- [Kosko95] B. Kosko, “Optimal fuzzy rules cover extrema”, Int. Journal of Intelligent Systems, vol. 10, pp. 249-255, 95.
- [Kueon95] Y. Kueon, J.S. Bedi, “Input-ouput fuzzy systems and the inverse pro-

- blem”, Proc of the SPIE, Applications of fuzzy logic technology II, Eds B. Bosacchi, J.C. Bezdek, vol. 2493, pp. 340-346, 95.
- [Lacrose97] V. Lacrose, “Réduction de la complexité des contrôleurs flous : application à la commande multivariable”, Thèse de doctorat, Institut National des Sciences Appliquées de Toulouse, Nov. 97.
- [Li95] H.X. Li, H.B. Gatland, “A new methodology for designing a fuzzy logic controller”, IEEE Trans. on Systems, Man, and Cybernetics, vol. 25, no. 3, pp. 505-512, March 95.
- [Li97] H.X. Li, H.B. Gatland, A.W. Green, “Fuzzy variable structure control”, IEEE Trans. on Systems, Man, and Cybernetics, Part B, vol. 27, no. 2, pp. 306-312, April 97.
- [Li99] H.X. Li, S.K. Tso, “Higher order fuzzy control structure for higher order or time-delay systems”, IEEE Trans. on Fuzzy Systems, vol. 7, no. 5, pp. 540-552, Oct. 99.
- [Lin95] Y. Lin, G.A. Cunningham III, “A new approach to fuzzy-neural system modeling”, IEEE Trans. on Fuzzy Systems, vol. 3, no. 2, pp. 190-198, May 95.
- [Lygeros97] J. Lygeros, “A formal approach to fuzzy modeling”, IEEE Trans. on Fuzzy Systems, vol. 5, no. 3, pp. 317-327, August 97.
- [MacVicar76] P.J. MacVicar-Whelan, “Fuzzy sets for man-machine interaction”, Int. J. Man-Machine Studies, vol. 8, pp. 687-697, 1976.
- [Mamdani75] E.H. Mamdani, S. Assilian, “An experiment in linguistic synthesis with a fuzzy logic controller”, Int. J. Man-Machine Studies, vol. 7, pp. 1-13, 1975.
- [Mann99] G.K.I. Mann, B.G. Hu, R.G. Gosine, “Analysis of direct action - Fuzzy PID Controller Structures”, IEEE Trans. on Systems, Man, and Cybernetics, Part B, vol. 29, no. 3, pp. 371-388, June 99.
- [Mauris92] G. Mauris, “Capteurs ultrasonores “intelligents” : application à la représentation symbolique de mesures de distance par codage flou”, Thèse de doctorat, Université de Savoie, Juin 92.
- [Mauris94] G. Mauris, E. Benoit, L. Foulloy, “Fuzzy symbolic sensors - From concepts to applications”, Measurement, vol. 12, pp. 357-384, 94.
- [Mauris96] G. Mauris, E. Benoit, L. Foulloy, “The aggregation of complementary information via fuzzy sensors”, Measurement, vol. 17, no. 4, pp. 235-249, 96.
- [Mizumoto93] M. Mizumoto, “Fuzzy control under product-sum gravity methods and new fuzzy control methods”, in Fuzzy Control Systems, A. Kandel and



G. Langholz, Eds. Boca Raton, FL: CRC Press, pp. 276-294, 1993.

- [Moon95] B.S. Moon, "Equivalence between fuzzy logic controllers and PI controllers for single input systems", *Fuzzy Sets and Systems*, vol. 69, pp. 105-113, 1995.
- [Mudi99] R.K. Mudi, N.R. Pal, "A robust self-tuning scheme for PI- and PD-type fuzzy controllers", *IEEE Trans. on Fuzzy Systems*, vol. 7, no. 1, pp. 2-16, Feb. 99.
- [Nakoula97a] Y. Nakoula, "Apprentissage des modèles linguistiques flous, par jeu de règles pondérées", Thèse de doctorat, Université de Savoie, Juillet 97.
- [Nakoula97b] Y. Nakoula, S. Galichet, L. Foulloy, "Identification of Linguistic Fuzzy Models Based on Learning", in *Fuzzy Model Identification - Selected Approaches*, H. Hellendoorn, D. Driankov (Eds.), Springer Verlag, pp. 281-319, 1997.
- [Nie96] J.H. Nie, T.H. Lee, "Rule-based modeling: fast construction and optimal manipulation", *IEEE Trans. on Systems, Man, and Cybernetics*, part A, vol. 26, no. 6, pp. 728-738, Nov. 96.
- [Nozaki97] K. Nozaki, H. Ishibuchi, H. Tanaka, "A simple but powerful heuristic method for generating fuzzy rules from numerical data", *Fuzzy Sets and Systems*, vol. 86, pp. 251-270, 1997.
- [Park00] S. Park, T. Han, "Iterative inversion of fuzzified neural networks", *IEEE Trans. on Fuzzy Systems*, vol. 8, no. 3, pp. 266-280, June 2000.
- [Pedrycz84] W. Pedrycz, "An identification algorithm in fuzzy relational systems", *Fuzzy Sets and Systems*, vol. 13, pp. 153-167, 1984.
- [Pedrycz96] W. Pedrycz, J.V. de Oliveira, "Optimization of fuzzy models", *IEEE Trans. on Systems, Man, and Cybernetics*, part B, vol. 26, no. 4, pp. 627-636, Aug. 96.
- [Pedrycz97] W. Pedrycz, M. Reformat, "Rule-based modeling of nonlinear relationships", *IEEE Trans. on Fuzzy Systems*, vol. 5, no. 2, pp. 256-269, May 97.
- [Rojas00] I. Rojas, H. Pomares, J. Ortega, A. Prieto, "Self-organized fuzzy system generation from training examples", *IEEE Trans. on Fuzzy Systems*, vol. 8, no. 1, pp. 23-36, Feb. 2000.
- [Siler89] W. Siler, H. Ying, "Fuzzy control theory: the linear case", *Fuzzy Sets and Systems*, vol. 33, pp. 275-290, 1989.
- [Slotine91] J.J. Slotine, W. Li, *Applied nonlinear control*, Prentice-Hall, 91.
- [Sudkamp94] T. Sudkamp, R.J. Hammel II, "Interpolation, completion and learning

- fuzzy rules”, IEEE Trans. on Systems, Man, and Cybernetics, vol. 24, no. 2, pp. 332-342, Feb. 94.
- [Sugeno91] M. Sugeno, K. Tanaka, “Successive identification of a fuzzy model and its applications to prediction of a complex system”, Fuzzy Sets and Systems, vol. 42, pp. 315-334, 1991.
- [Sugeno93] M. Sugeno, T. Yasukawa, “A fuzzy-logic-based approach to qualitative modeling”, IEEE Trans. on Fuzzy Systems, vol. 1, no. 1, pp. 7-31, Feb. 93.
- [Sugeno99] M. Sugeno, “On Stability of Fuzzy Systems Expressed by Fuzzy Rules with Singleton Consequents”, IEEE Trans. on Fuzzy Systems, vol. 7, no. 2, pp. 201-224, 1999.
- [Takagi85] T. Takagi, M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control”, IEEE Trans. on Systems, Man, and Cybernetics, vol. 15, no. 1, pp. 116-132, Feb. 85.
- [Tao00] C.W. Tao, J-S. Taur, “Flexible complexity reduced PID-like fuzzy controllers”, IEEE Trans. on Systems, Man, and Cybernetics, part B, vol. 30, 510-516, Aug. 00.
- [Tikk00] D. Tikk, P. Baranyi, “Comprehensive analysis of a new fuzzy rule interpolation method”, IEEE Trans. on Fuzzy Systems, vol. 8, no. 3, pp. 281-296, June 2000.
- [Tong78] R.M. Tong, “Analysis and control of fuzzy systems using finite discrete relations”, Int. J. Control, vol. 27, no. 3, pp. 431-440, 78.
- [Ueda84] M. Ueda, “Apparatus for indicating an optimum shift position for a manual transmission installed in a vehicle”, US Patent 4,550,596, Toyota, April 84.
- [Varkonyi99] A.R. Varkonyi-Koczy, A. Almos, T. Kovacs-hazy, “Genetic algorithms in fuzzy model inversion”, Proc. of Int. Fuzzy Systems Conference (FUZZ’IEEE99), Seoul, Korea, pp. 1421-1426, Aug. 99.
- [Wang92a] L.X. Wang, “Fuzzy systems are universal approximators”, Proc. IEEE Int. Conf. on Fuzzy Systems (FUZZ’IEEE 92), San Diego, pp. 1163-1170, Mar. 92.
- [Wang92b] L.X. Wang, J.M. Mendel, “Generating fuzzy rules by learning from examples”, IEEE Trans. on Systems, Man, and Cybernetics, vol. 22, no. 6, pp. 1414-1427, Nov. 92.
- [Wang96] L. Wang, R. Langari, “Complex systems modeling via fuzzy logic”, IEEE Trans. on Systems, Man, and Cybernetics, vol. 26, no. 1, pp. 100-106, Feb. 96.

- [Yam99] Y. Yam, P. Baranyi, C.T. Yang, "Reduction of fuzzy rule base via singular value decomposition", *IEEE Trans. on Fuzzy Systems*, vol. 7, no. 2, pp. 120-132, April 99.
- [Yeh99] Z.M. Yeh, "A systematic method for design of multivariable fuzzy logic control systems", *IEEE Trans. on Fuzzy Systems*, vol. 7, no. 6, pp. 741-752, Dec. 99.
- [Yen99] J. Yen, L. Wang, "Simplifying fuzzy rule-based models using orthogonal transformation methods", *IEEE Trans. on Systems, Man, and Cybernetics*, part B, vol. 29, no. 1, pp. 13-24, Feb. 99.
- [Ying90] H. Ying, W. Siler, J.J. Buckley, "Fuzzy control theory: a nonlinear case", *Automatica*, vol. 26, no. 3, pp. 513-520, 1990.
- [Ying93] H. Ying, "A Nonlinear Fuzzy Controller with Linear Control Rules is the Sum of a Global Two-dimensional Multilevel Relay and a Local Nonlinear Proportional-integral Controller", *Automatica*, vol. 29, no. 2, pp. 499-505, 1993.
- [Ying94] H. Ying, "Practical design of nonlinear fuzzy controllers with stability analysis for regulating processes with unknown mathematical models", *Automatica*, vol. 30, no. 7, pp. 1185-1195, 1994.
- [Ying97] H. Ying, G. Chen, "Necessary conditions for some typical fuzzy systems as universal approximators", *Automatica*, vol. 33, no. 7, pp. 1333-1338, 1997.
- [Yoshinari93] Y. Yoshinari, W. Pedrycz, K. Hirota, "Construction of fuzzy models through clustering techniques", *Fuzzy Sets and Systems*, vol. 54, pp. 157-165, 1993.
- [Zadeh71] L.A. Zadeh, "Quantitative fuzzy semantics", *Information Science*, vol. 3, no. 2, pp. 159-176, 1971.
- [Zadeh72] L.A. Zadeh, "A rationale for fuzzy control", *J. of Dynamical Systems, Measurement, and Control (Trans. ASME. Ser. G)*, vol. 94, no. 1, pp. 3-4, 1972.
- [Zadeh73] L.A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes", *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 3, no. 1, pp. 28-44, Jan. 73.
- [Zadeh96] L.A. Zadeh, "Fuzzy logic = Computing with words", *IEEE Trans. on Fuzzy Systems*, vol. 4, no. 2, pp. 103-111, 1996.
- [Zeng96a] X.J. Zeng, M.G. Singh, "Approximation accuracy analysis of fuzzy systems as function approximators", *IEEE Trans. on Fuzzy Systems*, vol. 4, no. 1, pp. 44-63, Feb. 96.

- [Zeng96b] X.J. Zeng, M.G. Singh, "A relationship between membership functions and approximation accuracy in fuzzy systems", IEEE Trans. on Systems, Man, and Cybernetics, part B, vol. 26, no. 1, pp. 176-180, Feb. 96.
- [Zeng96c] X.J. Zeng, M.G. Singh, "Decomposition property of fuzzy systems and its applications", IEEE Trans. on Fuzzy Systems, vol. 4, no. 2, pp. 149-165, May 96.

## Résumé

Utiliser l'ensemble des connaissances disponibles sur un procédé pour concevoir une loi de commande, tel est l'objectif de ce travail. Dans ce contexte, le choix d'un outil de représentation permettant d'intégrer dans un formalisme unique des informations de nature différente est essentiel. C'est donc tout naturellement que la dualité numérique/linguistique présente dans les systèmes flous à base de règles est exploitée pour réaliser un codage d'expertise dans un cadre numérique, autrement dit pour calculer avec les mots.

La première partie de ce travail est dédiée au caractère numérique des systèmes flous. A travers le principe des équivalences modales, nous proposons une stratégie de construction d'un contrôleur flou permettant d'implanter de façon exacte des lois de commande linéaires de type PI, PD, PID ou encore par retour d'état. L'approche développée est ensuite étendue à l'apprentissage de systèmes flous à partir d'exemples, et ainsi à la modélisation floue. Enfin, la synthèse d'un contrôleur à partir d'un modèle flou du procédé est abordée selon une approche par linéarisation entrée-sortie ou encore à partir d'une structure de commande à modèle interne. Si ces travaux permettent d'intégrer aisément des informations numériques pour concevoir un contrôleur flou, la prise en compte de connaissances expertes reste délicate et fait l'objet de la deuxième partie de ce mémoire.

A partir d'une typologie des systèmes flous, nous avons mis en évidence la possibilité d'implanter un contrôleur flou sous une forme purement symbolique. Dénotée SF4, cette mise en œuvre des systèmes flous exploite des règles linguistiques et autorise une pondération de ces dernières. A ce titre, elle est tout particulièrement adaptée à la représentation des connaissances d'expert et est utilisée tout au long de cette deuxième partie. Outre leur caractère linguistique, les systèmes SF4 sont directement exploitables dans un contexte numérique puisqu'ils permettent d'implanter tout système flou de Sugeno à conclusions constantes, et donc plus particulièrement ceux engendrés via le principe des équivalences modales. Basés sur cette cohérence numérique/linguistique, divers schémas d'introduction de la connaissance experte sont ainsi proposés, allant d'une synthèse directe du contrôleur flou à une modification d'un correcteur numérique initial. Enfin, la distribution de la connaissance dans différents composants de la structure de commande est étudiée (capteurs, superviseur, régulateurs).

Tout au long du mémoire, les diverses stratégies développées sont illustrées par le biais d'exemples tirés d'applications industrielles.

**Mots-clé :** Contrôle flou - Modèle flou - Expertise - Interpolation - Approximation