



UNIVERSITE SIDI MOHAMED BEN ABDELLAH



FACULTE des SCIENCES DHAR EL MEHRAZ de FES

Laboratoire d'Electronique, Signaux-Systèmes et d'Informatique

U.F.R : Automatique et Analyse des systèmes

Contribution à l'Identification et la Commande Floue d'une Classe de Systèmes Non Linéaires

THÈSE

présentée le 27/06/2009

pour l'obtention du

DOCTORAT NATIONAL

Spécialité : Automatique Signaux et Systèmes

par

Hamid Ouakka

devant le Jury composé par :

<i>Président :</i>	HMAMED Abdelaziz	PES- Faculté des Sciences. DM. Fès
<i>Directeur de Thèse :</i>	BOUMHIDI Ismail	PES- Faculté des Sciences. DM. Fès
<i>Rapporteurs :</i>	BOUZOUBA Khalid	PES- Faculté des Sciences. DM. Fès
	SBAI El Hassan	PES-Ecole Supérieure de Technologie, Meknès
<i>Examineurs :</i>	ERRAHIMI Fatima	PES- Faculté des Sciences et Techniques de Fès
	ROUKHE Ahmed	PES-Faculté des Sciences Meknès
	QJIDAA Hassan	PES- Faculté des Sciences. DM. Fès

Remerciements

L'ensemble des travaux présentés dans ce mémoire a été effectué au laboratoire d'Electronique, Signaux-Systèmes et d'Informatique de la Faculté des Sciences Dhar El Mahraz de Fès.

*Mes sincères remerciements à Monsieur **El Hassan SBAI** Professeur à l'école supérieure de technologie de Meknès, et Monsieur **Khalid BOUZOUBA** Professeur à la Faculté des Sciences Dhar El Mahraz de Fès d'avoir pris le temps de juger ce travail et de m'avoir fait l'honneur d'être rapporteurs de ma thèse.*

*J'adresse mes sincères remerciements à **Abdelaziz HMAMED**, Directeur du laboratoire L.E.S.S.I à la Faculté des Sciences Dhar El Mahraz de Fès, d'avoir accepté d'examiner ce travail et de présider le jury.*

*Toute ma profonde gratitude va également à Madame **Fatima ERRAHIMI** Professeur à la Faculté des Sciences et Techniques de Fès, à Monsieur **Hassan QJIDAA** Professeur à la Faculté des Sciences Dhar El Mahraz de Fès et Monsieur **Ahmed ROUKHE** Professeur à la Faculté des Sciences de Meknès, pour l'intérêt qu'ils ont témoigné à l'égard de ce travail et d'avoir accepté de participer au jury de thèse.*

*Je tiens à exprimer ma reconnaissance au Professeur **Ismail BOUMHIDI** en tant que directeur de thèse pour m'avoir accueilli au sein de son équipe, pour ses directives scientifiques, pédagogiques et même personnelles pour lesquelles je lui suis hautement redevable. De même, je lui suis extrêmement reconnaissant pour son soutien humain et moral et son aide précieuse à la rédaction de ce manuscrit de thèse.*

*Enfin, je tiens tout particulièrement à remercier Monsieur **Moha OUAMANE**, pour ces encouragements, son aide, sa disponibilité et chaleur humaine.*

*Je dédie cette thèse
à mes parents.
à ma femme Bouchra.
à mes enfants Zakariae, Youssef et Asmae.
à mes frères et soeurs.
à mon grand ami DODO et sa famille
à tous mes amis .*

Table des matières

Introduction générale	7
1 Etat de l’art sur la Modélisation et Commande floue des systèmes non linéaires	10
1.1 Introduction	10
1.2 Concepts de base de la logique floue	11
1.2.1 Variables linguistiques et ensembles flous	12
1.2.2 Fonctions d’appartenances	13
1.2.3 Opérations sur les ensembles flous	14
1.2.4 Les règles floues	14
1.2.5 Description et structure d’une commande par la logique floue	15
1.2.5.1 Interface de fuzzification	16
1.2.5.2 Interface d’inférence floue	16
1.2.5.3 Interface de défuzzification	17
1.3 Modélisation floue	17
1.3.1 Concept de la classification floue	18
1.3.1.1 Matrice de données	19
1.3.1.2 Partition floue	19
1.3.2 Algorithmes de classification floue	20
1.3.2.1 Algorithme des c-moyennes floues(FCM)	21
1.3.2.2 Algorithme de Gustafson-Kessel(GK)	23
1.3.3 Construction de modèles TS à partir des données	25
1.3.3.1 Sélection de la structure	25
1.3.3.2 Classification des données	25
1.3.3.3 Génération des fonctions d’appartenance des antécédents	25

1.3.3.4	Obtention des paramètres des conséquents	27
1.3.4	Modèle flou de Takagi-Sugeno	28
1.4	Conception d'une commande floue	31
1.4.1	Commande adaptative	32
1.4.2	Commande par l'approche mode de glissement	32
1.5	Conclusion	33

2	Identification des Systèmes Non Linéaires à base de Modèles de type TS	35
2.1	Introduction	35
2.2	Identification d'un système MIMO	36
2.2.1	Représentation des systèmes MISO	36
2.2.2	Identification hors ligne du modèle flou	38
2.2.3	Adaptation en ligne du modèle flou	39
2.2.4	Simulation	39
2.3	Méthodes de validation d'une structure optimale	42
2.3.1	Algorithmes d'optimisation	43
2.3.2	Indices d'évaluation de la qualité d'une partition	44
2.3.3	Validation numérique du modèle flou	46
2.4	Identification optimale des modèles Takagi-Sugeno	46
2.4.1	Formulation de la méthode proposée	47
2.4.2	Approximation des données par fonction polynomiale	47
2.4.3	Extraction du nombre globale de classes	50
2.4.4	Détermination du nombre optimal de classes	51
2.5	Simulations et résultats	54
2.6	Conclusion	56

3	Commande Floue d'une Classe de Systèmes Non Linéaires	57
3.1	Introduction	57
3.2	Modélisation floue de Takagi-Sugeno	58
3.3	Synthèse de la loi de commande pour un système SISO	59
3.3.1	Cas non adaptatif	59
3.3.2	Cas adaptatif	59
3.4	Synthèse de la loi de commande pour un système MISO	61

3.4.1	Description du modèle flou	61
3.4.2	Synthèse de la loi de commande	63
3.5	Synthèse d'une loi de commande décentralisée	65
3.5.1	Problématique	65
3.5.2	Contrôleur adaptatif flou direct	66
3.5.3	Contrôleur adaptatif flou indirect	66
3.6	Régulateur PI combiné à l'approche mode glissant	67
3.6.1	Formulation du problème	67
3.6.2	Structure de la loi de commande	68
3.6.3	Conception du Régulateur PI	69
3.7	Résultats de simulations	72
3.7.1	Cas du système SISO	72
3.7.2	Cas d'un Système MISO	73
3.7.3	Cas d'un grand système	77
3.7.4	Cas du régulateur PI combiné à l'approche mode glissant	80
3.8	Conclusion	82

Conclusion générale	83
----------------------------	-----------

Bibliographie	85
----------------------	-----------

Abréviations

FCM	:	Algorithme de classification des c-moyennes floues (Fuzzy c-means)
GK	:	Algorithme de classification de Gustafson-kessel
TS	:	Takagi-Sugeno
SISO	:	Mono entrée Mono sortie (Single Input Single Output)
MISO	:	Multi entrées Mono sortie (Multiple Input Single Output)
MIMO	:	Multi entrées Multi sorties (Multiple Input Multiple Output)
PI	:	Proportionnel intégral
PC	:	Coefficient de partition (Partition coefficient)
CE	:	Coefficient d'entropie de partition (classification entropy)
VAF	:	Comptabilisation en pourcentage de la variance (Variance Accounting For)
FMID	:	Boite à outils Modélisation et Identification floue (Fuzzy Modelling and Identification toolbox)
ANFIS	:	Système adaptatif d'inférence neuro-floue (Adaptive neuro-fuzzy inference system)

Introduction générale

Contrairement à l'automatique linéaire, l'automatique non linéaire ne dispose pas de solutions universelles ni pour l'analyse des systèmes ni pour la conception de leurs contrôleurs. L'analyse et la commande de ces systèmes ne sont pas, toujours, des tâches faciles. La plupart des travaux existants dans la littérature proposent des approches qui sont, généralement, limitées à des formes bien particulières de systèmes [Chaoui *et al.*, 2000, 2001]. De plus, les performances assurées sont, souvent, au prix de la complexité du schéma de commande et du développement théorique utilisé. La plupart des approches de commande non linéaires exigent la disponibilité d'un modèle mathématique du système. Les performances assurées, seront directement liées à l'exactitude du modèle utilisé.

En automatique, pour décrire le comportement d'un système, une hypothèse communément faite est la linéarité du système, car les techniques d'analyse des modèles linéaires, ont été largement développées dans la littérature. Cependant, l'hypothèse de linéarité n'est vérifiée que dans une plage de fonctionnement restreinte autour d'un point d'équilibre du système. Alors, les performances du modèle se dégradent dès qu'on s'en éloigne et la recherche d'un modèle plus adapté et notamment non linéaire devient nécessaire.

La structure mathématique qui puisse remédier à l'inconvénient cité ci-dessus, tout en gardant la simplicité mathématique des modèles linéaires, est l'approche dite multi-modèle ; c'est une représentation pouvant être obtenue soit directement à partir d'un modèle mathématique non linéaire par transformation directe d'un modèle affine [Morère, 2000] ou par linéarisation autour de différents points de fonctionnement [Murray-Smith et Johansen, 1999], soit à partir des données entrées sorties d'un système physique [Gasso, 2000].

Les systèmes flous, basés sur la théorie de la logique floue, ont été utilisés comme alternative pour construire de telles structures multi-modèles. Ils présentent l'avantage de tolérer l'incertitude du modèle et compensent son effet [Guesmi *et al.*, 2005; Pagès et Hajjaji, 2005], de traiter les non linéarités sans aucune hypothèse sur leur nature, de modéliser puis compenser les interactions entre les boucles et de réduire l'effet des perturbations externes. Leur qualité

d'approximateurs universels a été démontrée, notamment pour les applications en commande et en identification floue [Jain et Dubes, 1988; Wang et Mendel, 1992]. On distingue deux classes principales de modèles flous : les modèles flous de Mamdani [Mamdani et Assilian, 1975] et ceux de Takagi-Sugeno (TS)[Sugeno et Kang, 1988; Takagi et Sugeno, 1985]. Ils diffèrent au niveau de la conclusion.

La complexité du système est réduite par les algorithmes de classification floue qui procèdent en une première phase à la décomposition de l'espace de données en un nombre de sous ensembles de données, puis chaque groupe de données est approximé par un modèle linéaire à l'aides des techniques d'identification existantes dans la littérature [Bezdek *et al.*, 1987][Abo-ny *et al.*, 2002][Gustafson et Kessel, 1978][Ghat et Geva, 1989]. L'inconvénient majeur de cet approche est l'optimisation du nombre de classes.

La structure de la base de règles floues est un facteur important qui affecte la performance de la modélisation. Lorsque le nombre de règles est grand, la précision est bonne mais le modèle flou devient complexe.

Ce mémoire est structuré en trois chapitres.

Le premier chapitre est à caractère bibliographique, il présente les bases de la logique floue, allant du concept des ensembles flous aux principe de fuzzification et déffuzification en passant par la méthode d'inférence ainsi que la structure des fonctions d'appartenance pour le cas monovariante et multivariante. La modélisation type TS d'un système non linéaire ainsi que son extension au cas multivariante ont été aussi décrites dans ce chapitre. Nous terminons par la description d'un schéma général du contrôleur flou.

Le second chapitre présente deux stratégies de modélisation d'un système non linéaire par la technique de TS en tenant compte de tous les effets perturbateurs : La première concerne un système multivariante, l'excitation riche est choisie d'abord aléatoirement, ensuite carré, l'étude a été faite en boucle ouverte. Les résultats de simulation obtenus montrent une bonne poursuite de sorties réelles aux modèles flous correspondants. La deuxième approche proposée consiste à optimiser le nombre de classes sans passer par les méthodes itératives classiques qui nécessitent en plus des hypothèses restrictives d'un temps et charge de calcul élevés. La technique proposée est principalement basée sur l'approximation des données par une fonction polynomiale approchée, accompagnée d'un algorithme d'optimisation, facile à mettre en oeuvre est conduisant au même nombre de classes que l'approche classique. L'efficacité de l'approche proposée est testée sur un exemple de simulation.

Nous proposons dans **le Troisième chapitre** quatre de stratégies de commande.

- La première est destinée à commander les systèmes mono entrée mono sortie (SISO) ; elle est basée sur la combinaison de l'approche de linéarisation entrée sortie aux mécanismes de la logique floue. Dans un premier volet, la loi de commande est directement déterminée

à partir des paramètres du modèle flou, cette commande s'est avérée dégradée en terme de poursuite ce qui nous a amené à adapter les paramètres du modèle afin de suivre le système dans son évolution.

- La deuxième approche est une extension du cas monovariable au cas multivariable. Ce dernier est décomposé en un ensemble de sous systèmes MISO, afin d'éviter la charge lourde des calculs et de faciliter l'applicabilité. Chaque sous système MISO est décrit par un ensemble de modèles locaux linéaires. Notons que la modélisation est effectuée en deux étapes :

- Hors ligne pour déterminer la structure du modèle flou.
- En ligne pour déterminer récursivement les paramètres du modèle.

Cette technique nous a permis de compenser facilement le couplage croisé.

- La troisième approche concerne les grands systèmes, en dépit des interconnexions, la technique proposée traite les deux cas de commande direct et indirect. Chaque sous systèmes est supposé inconnu, il est modélisé en utilisant la technique de Mamdani. Pour l'approche directe la conclusion est faite sur la commande et ses paramètres sont ensuite adaptés par l'approche de Lyapunov. Pour l'approche indirecte la conclusion est faite sur le modèle, suivi de l'adaptation des paramètres et par la suite la synthèse de la commande. Pour chacune de ces deux approches, la stratégie de commande consiste à composer la commande de trois termes : un terme de poursuite, résultat du mécanisme flou ; un autre destiné à compenser les interconnexions et un dernier pour réduire les erreurs de modélisation.
- la dernière consiste à combiner le régulateur PI, la commande mode de glissement et la technique de la logique floue. Cette stratégie a permis d'éliminer le phénomène de réticence et d'assurer la convergence asymptotique de l'erreur de poursuite. La surface de glissement sera décrite par des ensembles flous et selon la prémisse de règle, une loi de commande, soit le régulateur PI ou la commande par mode de glissement est activée. Les paramètres de ce contrôleur sont déterminés par l'approche de Lyapunov de sorte à maintenir la stabilité.

Les approches proposées sont testées sur différents exemples de simulations. Les résultats obtenus montrent une grande efficacité en terme de poursuite et de convergence.

Etat de l'art sur la Modélisation et Commande floue des systèmes non linéaires

1.1 Introduction

Le développement de modèles mathématiques des systèmes est un sujet central dans plusieurs disciplines des sciences et de l'ingénierie. Traditionnellement, la modélisation est vue comme la double conjonction entre la compréhension de la nature et du comportement d'un système ainsi que le traitement mathématique approprié qui conduit à l'obtention d'un modèle utilisable. Néanmoins, le besoin d'une forte compréhension des éléments physiques de base, comme c'est le cas des systèmes à dynamiques inconnues ou partiellement connues, constitue une grande restriction au niveau pratique. Dans ce cas, le système étudié peut être représenté en utilisant une structure générale qui approxime son comportement. Le problème de modélisation consiste alors à proposer la structure appropriée pour l'approximateur et d'estimer les paramètres du modèle. Cette dernière étape se fait généralement en utilisant le critère des moindres carrés avec normalisation et projection des données.

Dans le sens de mieux maîtriser les informations imprécises, la logique floue a été introduite de sorte à formaliser les méthodes humaines de raisonnement en utilisant des bases de règles et variables linguistiques pour la représentation de connaissances [Zadeh, 1973]. La commande floue a été une des grandes applications de la logique floue, dans le sens de surmonter tous les problèmes relatifs aux non linéarités, perturbations externes et aux interactions entre les différentes boucles lorsqu'il s'agit des systèmes multivariables ou des systèmes interconnectés.

La plupart des applications développées reposent sur l'expertise de l'opérateur humain. On distingue ainsi, deux types de systèmes flous ; le système flou de Takagi-Sugeno et celui de Mamdani. Le premier est à conclusion numérique et le second est à conclusion symbolique, ce

qui nécessite d'autres approches pour ressortir l'information. L'application de l'un ou l'autre de ces systèmes flous permet de transformer l'étude et la commande d'un système non linéaire en systèmes plus simples, de forme linéaire dans le cas de la modélisation TS. Notons toutefois, que dans le cadre de la classification floue des données expérimentales, différentes méthodes itératives ont été présentées, avec lesquelles nous allons comparer l'approche proposée dans ce domaine.

Les étapes de classification, de modélisation et d'estimation paramétrique tiennent compte de tous les effets perturbateurs, et mènent généralement à des modèles linéaires à paramètres connus ce qui facilitera la mise en place d'une loi de commande robuste vis à vis des erreurs de modélisation, des incertitudes paramétriques et des perturbations non modélisables telle que l'approche mode glissant combiné au régulateur PI et les concepts de logique floue abordée dans ce mémoire.

1.2 Concepts de base de la logique floue

La plupart des systèmes non linéaires sont modélisables sous des hypothèses parfois très restrictives. Ces hypothèses, rendent difficiles la mise en oeuvre des schémas de commande résultants et leur application. Il est donc nécessaire de prendre en compte toutes les informations imprécises et incertaines relatives au système. La théorie des sous ensembles flous développée par Lotfi A. Zadeh en 1965, a permis de traiter les imprécisions et les incertitudes. De nombreuses applications sont alors développées dans divers domaines, là où aucun modèle déterministe n'existe ou n'est possible d'obtenir.

L'avantage d'un système flou est que seules les connaissances du comportement du procédé à commander sont suffisantes pour la synthèse de la loi de commande, et ils soulèvent un large intérêt, tant théorique que pratique, dans l'identification et la commande des processus complexes et non linéaires. Cela est dû essentiellement à trois traits principaux :

- Le premier est que les systèmes flous permettent une simple inclusion d'informations qualitatives dans la conception du contrôleur ;
- Le second est que les systèmes flous n'exigent pas l'existence d'un modèle analytique du processus à contrôler, et peu d'information est suffisant pour mettre en oeuvre la boucle de commande ;
- Le troisième est que les systèmes flous sont des systèmes non linéaires et de ce fait plus adaptés à la commande des processus non linéaires.

1.2.1 Variables linguistiques et ensembles flous

La description imprécise d'une certaine situation, d'un phénomène ou d'une grandeur physique ne peut se faire que par des expressions relatives ou floues. Ces différentes classes d'expressions floues dites ensembles flous forment ce qu'on appelle des variables linguistiques. Afin de pouvoir traiter numériquement ces variables linguistiques qui sont normalisées généralement sur un intervalle bien déterminé appelé univers de discours, il faut les soumettre à une définition mathématique à base de fonctions d'appartenance qui montrent le degré de vérification de ces variables linguistiques relativement aux différents sous ensembles flous de la même classe.

La figure (Fig.1.1) montre un exemple de variable linguistique associée au concept de température, représentée par les sous ensembles flous où les termes linguistiques sont définis par : {froide, moyenne, chaude} sur l'univers de discours représenté par les températures comprises dans l'intervalle $[0^\circ, 70^\circ]$.

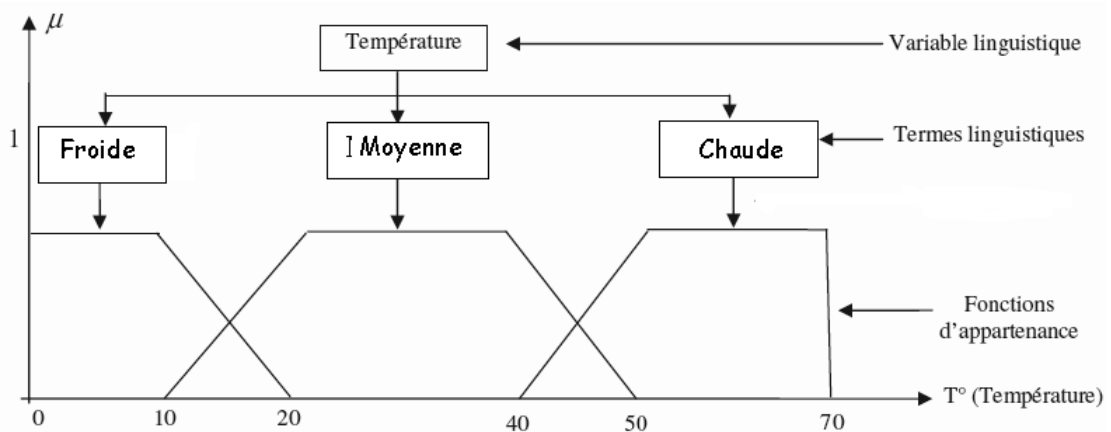


FIGURE 1.1 – Exemple d'ensembles flous pour la variable température

Une variable linguistique permet donc, d'une part de synthétiser l'information manipulée grâce aux sous ensembles flous, et d'autre part de représenter des concepts imprécis tels que l'homme en manipule quotidiennement. La détermination de la forme et de la position de ces sous ensembles flous sont définis a priori par des experts du domaine afin qu'ils représentent exactement leurs connaissances. Cependant, il n'est pas toujours possible d'obtenir une telle expertise, que ce soit à cause de la complexité du problème ou bien parce que les experts sont trop rares voir inexistants. Dans ces conditions, des algorithmes peuvent être mis en œuvre pour extraire automatiquement les sous ensembles flous. Une expertise du résultat peut éventuellement être faite par la suite afin de déterminer la signification des sous ensembles flous obtenus.

1.2.2 Fonctions d'appartenances

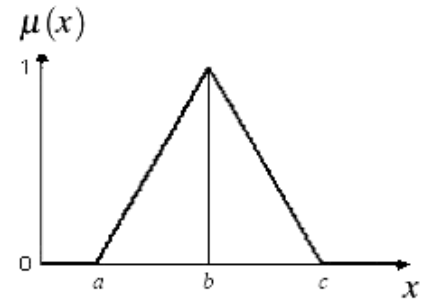
Une fonction d'appartenance d'un ensemble flou A définie sur l'univers de discours X , notée $\mu_A(x)$ tel que $x \in X$, est une courbe qui définit comment chaque point dans l'univers de discours est tracé avec une valeur d'appartenance comprise dans l'intervalle $[0, 1]$ [Mendel, 2000; Meunier, 1995] :

$$\begin{aligned}\mu_A(x) : X &\longrightarrow [0, 1] \\ x &\longrightarrow \mu_A(x)\end{aligned}$$

La valeur $\mu_A(x)$ mesure l'appartenance ou le degré avec lequel un élément x appartient à l'ensemble A . Il n'y a pas de règle précise pour la définition de fonction d'appartenance. Alors, chaque ensemble flou peut être représenté par sa fonction d'appartenance. Les fonctions d'appartenance peuvent être symétriques, régulièrement distribuées ou avoir une distribution non uniforme. En général, la forme des fonctions d'appartenance dépend de l'application et de la grandeur à modéliser et peuvent avoir différentes formes :

- *Fonction triangulaire*

$$\mu(x) = \begin{cases} \frac{a-x}{a-b} & \text{si } x \in [a, b] \\ \frac{b-x}{b-c} & \text{si } x \in [b, c] \end{cases}$$

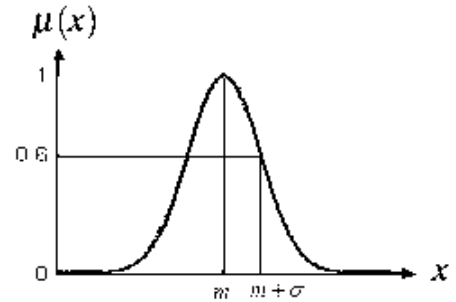


- *Fonction gaussienne*

$$\mu(x, m, \sigma) = e^{-\left(\frac{x-m}{2\sigma}\right)^2}$$

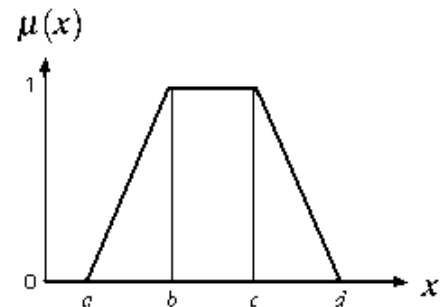
m : centre de la gaussienne

σ : sa largeur



- *Fonction trapézoïdale*

$$\mu(x) = \begin{cases} \frac{a-x}{a-b} & \text{si } x \in [a, b] \\ 1 & \text{si } x \in [b, c] \\ \frac{d-x}{c-d} & \text{si } x \in [c, d] \end{cases}$$



1.2.3 Opérations sur les ensembles flous

Supposons que A et B sont deux ensembles flous définis dans un univers de discours X par les fonctions d'appartenance μ_A et μ_B . On peut définir des opérations telles que l'égalité, l'inclusion, l'intersection, l'union et le complément grâce à des opérations sur les fonctions d'appartenance.

- **Egalité** : A et B sont dits égaux, propriété que l'on note $A = B$, si leurs fonctions d'appartenance prennent la même valeur en tout point de X :

$$\forall x \in X \quad \mu_A(x) = \mu_B(x)$$

- **Inclusion** : A est dit inclus dans B , propriété que l'on note $A \subseteq B$, si tout élément x de X qui appartient à A appartient aussi à B :

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

- **Intersection** : L'intersection de A et B , que l'on note $A \cap B$, est l'ensemble flou constitué des éléments de X affectés du plus petit des deux degrés d'appartenance μ_A et μ_B :

$$\forall x \in X \quad \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$$

- **Union** : L'union de A et B , que l'on note $A \cup B$, est l'ensemble flou constitué des éléments de X affectés du plus grand des deux degrés d'appartenance μ_A et μ_B :

$$\forall x \in X \quad \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$$

Dans ces définitions, $\min$ et $\max$ désignent, respectivement, l'opérateur de calcul du minimum et du maximum des deux valeurs.

- **Complément** : Le complément de A , que l'on note $\bar{A}$, est l'ensemble flou de X constitué des éléments x lui appartenant d'autant plus qu'ils appartiennent peu à A :

$$\forall x \in X \quad \mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

1.2.4 Les règles floues

Une règle floue R : **Si** x est A **Alors** y est B est une relation entre deux propositions floues ayant chacune un rôle particulier. La première (x est A) est appelée prémisse de la règle alors que la seconde (y est B) est la conclusion.

Dans le cas de propositions floues élémentaires, la prémisse et la conclusion sont définies à partir de deux variables linguistiques A et B décrivant les connaissances relatives aux univers de discours X_A et X_B de manière à prendre en compte l'imprécision relative aux modalités de A et B . Une proposition floue élémentaire est souvent insuffisante pour représenter l'ensemble des informations à manipuler. Plusieurs propositions floues peuvent alors être combinées pour enrichir et détailler la représentation.

La relation R entre la prémisse et la conclusion de la règle est déterminée par une implication floue dont le degré de vérité est défini par une fonction d'appartenance μ_R qui dépend du degré de vérité μ_A et μ_B de chacune des deux propositions élémentaires. Les implications les plus courantes permettant la détermination de la fonction d'appartenance résultante décrivant la proposition floue R sont donnés par :

$$\left\{ \begin{array}{ll} \text{l'implication de Mamdani :} & \mu_R(x,y) = \min(\mu_A(x), \mu_B(y)) \\ \text{l'implication de Larsen :} & \mu_R(x,y) = \mu_A(x) \cdot \mu_B(y) \end{array} \right. \quad (1.1)$$

$x \in X_A \text{ et } y \in X_B$

1.2.5 Description et structure d'une commande par la logique floue

Contrairement aux techniques de réglage classique, le réglage par la logique floue n'utilise pas des formules ou des relations mathématiques bien déterminées ou précises [Zeng et Singh, 1995]. Mais, il manipule des inférences avec plusieurs règles floues à base des opérateurs flous *ET*, *OU*, *ALORS*, ...

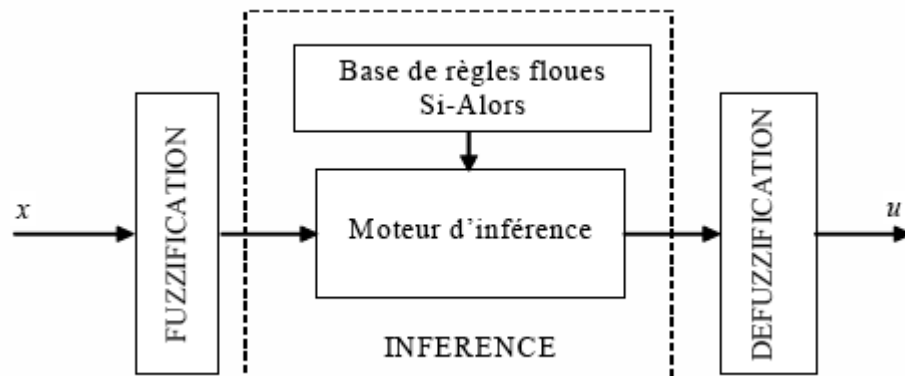


FIGURE 1.2 – Structure interne d'un contrôleur flou.

On peut distinguer trois parties principales constituant la structure d'un régulateur flou :

- une interface de Fuzzification,
- un mécanisme d'Inférence,

- et une interface de Défuzzification

La figure (Fig.1.2) représente, à titre d'illustration, la structure d'un régulateur flou à une entrée x et une sortie u .

1.2.5.1 Interface de fuzzification

La fuzzification proprement dite consiste à définir des fonctions d'appartenance pour les différentes variables linguistiques. Ceci a pour but la conversion d'une grandeur physique en une linguistique. Il s'agit d'une projection de la variable physique sur les ensembles flous caractérisant cette variable. Cette opération permet d'avoir une mesure précise sur le degré d'appartenance de la variable d'entrée à chaque ensemble flou.

D'une autre manière, l'entrée x varie dans l'univers de discours qui est partagé en un nombre fini d'ensembles flous de telle sorte que dans chaque zone il y ait une situation dominante. Afin de faciliter le traitement numérique et l'utilisation de ces ensembles, on les décrit par les fonctions d'appartenance. Elles admettent comme argument la position de x dans l'univers de discours, et comme sortie le degré d'appartenance de x à la situation décrite par la fonction.

1.2.5.2 Interface d'inférence floue

L'interface d'inférence est formée de deux blocs :

- La base de règles, composé d'un ensemble de relations liant les variables d'entrées aux variables de sorties du système à régler. Chaque relation est composée d'une condition précédée du symbole **Si** appelée *prémisse*, et d'une *conclusion* (action, décision, opération ou commande) précédée du symbole **Alors**.
- Le moteur d'inférence réalise le traitement numérique des règles d'inférence, décrites par des opérateurs flous, pour obtenir la sortie linguistique ou floue du régulateur. Cette opération est faite par différentes méthodes, on cite principalement :

La méthode d'inférence max-min : L'opérateur *ET* est réalisé par la formation du minimum, l'opérateur *OU* est réalisé par la formation du maximum, et l'implication (*ALORS*) est réalisée par la formation du minimum.

La méthode d'inférence max-produit : L'opérateur *ET* est réalisé par la formation du produit, l'opérateur *OU* est réalisé par la formation du maximum, et l'implication (*ALORS*) est réalisée par la formation du produit.

La méthode d'inférence somme-prod : On réalise au niveau de la condition, l'opérateur *OU* par la formation de la somme (valeur moyenne), et l'opérateur *ET* par la formation du produit. Pour la conclusion, l'opérateur *ALORS* est réalisé par un produit.

1.2.5.3 Interface de défuzzification

La transformation d'une information floue en une information déterminée est la défuzzification (concrétisation). Pendant cette étape se fait la déduction de la grandeur de sortie numérique à partir de l'inférence floue. Il s'agit de calculer, à partir des degrés d'appartenance à tous les ensembles flous des variables d'entrées et des ensembles flous de la variable de sortie, une valeur numérique de la variable de sortie en utilisant un ensemble de règles. Parmi les stratégies de défuzzification, on cite la méthode du centre de gravité, la méthode du maximum et la méthode de la moyenne des maximums.

La technique du maximum : elle est la plus simple. Elle consiste à ne considérer, pour chaque sortie, que la règle présentant le maximum de validité. Cette règle, ignore les règles secondaires qui peuvent néanmoins être importantes pour le fonctionnement et la stabilité du système. Elle est par conséquent peu employée.

La technique de la moyenne des maximums : elle considère, comme valeur de sortie, la moyenne de toutes les valeurs pour lesquelles la fonction d'appartenance issues de l'inférence est maximale.

La technique du centre de gravité : plus performante, elle consiste à tracer, sur un même diagramme représentant les ensembles flous de sortie, les différentes zones correspondantes à chacune des règles et à calculer le centre de gravité de la zone consolidée. La méthode de défuzzification la plus mentionnée dans la littérature est celle de la détermination de l'abscisse u_{G_r} du centre de gravité de la fonction d'appartenance résultante $\mu_r(u)$.

Nous avons introduit les notions de base de la théorie des ensembles flous et de la logique floue. Nous pouvons dire que la logique floue ouvre des possibilités remarquables d'exploitation des connaissances des experts. Les applications utilisant la logique floue sont faciles à réaliser et à utiliser.

Dans le cas des situations plus complexes, la capacité de l'être humain à produire une représentation et une description de ces systèmes est devenue primordiales . C'est ainsi que de nouvelles méthodologies ont été développées combinant la logique floue, les techniques de modélisations et les approches de commandes.

1.3 Modélisation floue

Le développement de modèles mathématiques pour les systèmes non linéaires est un sujet central dans plusieurs disciplines des sciences et de l'ingénierie. Traditionnellement, la modélisation est vue comme la double conjonction entre la compréhension de la nature et du comportement d'un système ainsi que le traitement mathématique approprié qui conduisent à l'obtention

d'un modèle utilisable. Le besoin d'une forte compréhension des éléments physiques de base constitue une grande restriction au niveau pratique quand on est confronté aux systèmes complexes ou mal connus.

En effet, la mise en équations des lois qui gouvernent de tels systèmes conduit généralement à un modèle de connaissance trop complexe et une mise en oeuvre délicate. Dans ce cas, le recours à des techniques de modélisation élaborées à partir des mesures d'entrée/sortie recueillies sur le système s'impose.

L'identification et la modélisation floues à partir de données expérimentales sont des outils efficaces pour approximer les systèmes non linéaires. Parmi les modèles largement utilisés, nous trouvons ceux de Takagi-Sugeno (TS) [Takagi et Sugeno, 1985][Angelov et Filev, 2004]. L'une des techniques utilisées pour construire ces modèles est la classification floue [Bezdek et Dunn, 1975][Babuska et Verbruggen, 1995] combinée aux techniques des moindres carrés [Abonyi *et al.*, 2002].

Dans cette section, nous allons présenter le principe de la classification floue ainsi que les différents algorithmes associés, la méthode de construction des modèles de Takagi-Sugeno à partir des données et un tour d'horizon des approches de validation de la qualité de classification et de la performance de la modélisation.

1.3.1 Concept de la classification floue

Lorsque la connaissance de l'expert n'est pas disponible, l'identification d'une structure doit être faite à partir des données. Une méthode de classification floue peut alors permettre de partitionner l'espace des données en plusieurs classes. Chacune de ces classes ou régions floues est caractérisée par un vecteur appelé centre de classe.

L'appartenance des données à un groupe est basé sur la vérification d'un degré de similitude. Habituellement, il est calculé en utilisant une mesure appropriée de distance qui quantifie la distance entre les données, représentées comme des points dans l'espace caractéristique, et les centres des groupes.

L'utilisation d'un algorithme de classification floue a l'avantage essentiel de permettre la génération automatique des fonctions d'appartenance ou des régions floues à partir des données. Elles sont construites, en minimisant une fonction coût. Parmi les techniques de classification floue les plus utilisées, les méthodes d'apprentissage supervisées et non supervisées. Ces dernières ne demandent aucune connaissance a priori sur la structure des données [Babuska, 1998]. Leur principal inconvénient est la nécessité d'initialiser les algorithmes par un nombre de classe, paramètre d'entrée, qui doit être estimé et fixé à priori [Frigui et Krishnapuram, 1996]. Ce paramètre joue un rôle important dans la réalisation d'une classification optimale.

1.3.1.1 Matrice de données

Dans les méthodes de classification floue, les données sont typiquement des observations (mesures) issues d'un certain processus physique. Chaque k -ième observation constitue un vecteur noté par $Z_k = [z_{1k}, z_{2k}, \dots, z_{nk}]$, avec $1 \leq K \leq N$, $z_k \in \mathbb{R}^n$, ou N représente le nombre des observations et n correspond au nombre de variables mesurées. Un ensemble de N observations est dénoté par $Z = \{z_k, \quad k = 1, 2, \dots, N\}$ et il est représenté comme étant une matrice donnée par :

$$Z = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1N} \\ z_{21} & z_{22} & \dots & z_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \dots & z_{nN} \end{bmatrix} \quad (1.2)$$

Z sera appelée la matrice de données.

1.3.1.2 Partition floue

Les algorithmes de classification floue visent à réaliser un partitionnement d'un ensemble de données en établissant une *partition* floue des observations en un certain nombre de classes. La notion de partition floue, qui s'est avérée d'une grande utilité pour le développement des techniques floues de classification, a été introduite par Ruspini [Ruspini, 1969]. Au sens de Ruspini (partition floue stricte), une c -partition floue d'un ensemble Z peut être obtenue en définissant des c sous-ensembles flous de Z tel que la somme des degrés d'appartenance pour chaque observation de Z soit unitaire.

En fait, on associe à chaque observation $z_k \in Z$, un vecteur de degrés d'appartenance aux différentes classes. La juxtaposition de ces vecteurs pour l'ensemble des N observations de Z amène à la définition d'une *matrice d'appartenance* U (de dimension $c \times N$) où l'élément μ_{ik} représente le coefficient d'appartenance de l'observation z_k à la classe i , $i = 1, \dots, c$. Cette matrice établit une relation d'ordre floue et traduit l'idée d'une partition floue en c classes.

La matrice d'appartenance $U = [\mu_{ik}]$ est également appelée matrice de partition floue et respecte les propriétés suivantes :

$$U = \begin{bmatrix} \mu_{11} & \mu_{12} & \dots & \mu_{1j} & \dots & \mu_{1N} \\ \mu_{21} & \mu_{22} & \dots & \mu_{2j} & \dots & \mu_{2N} \\ \cdot & \cdot & \dots & \cdot & \dots & \cdot \\ \mu_{i1} & \mu_{i2} & \dots & \mu_{ij} & \dots & \mu_{iN} \\ \cdot & \cdot & \dots & \cdot & \dots & \cdot \\ \mu_{c1} & \mu_{c2} & \dots & \mu_{cj} & \dots & \mu_{cN} \end{bmatrix} \quad (1.3)$$

1. la i -ème ligne de U , $U_i = (\mu_{i1}, \mu_{i2}, \dots, \mu_{in})$ contient les n degrés d'appartenance au i -ième sous ensemble flou,
2. la j -ième colonne $U^j = (\mu_{1j}, \mu_{2j}, \dots, \mu_{cj})$ contient les c degrés d'appartenance du j -ième élément de Z au c sous ensembles flous,
3. la somme de tous les degrés d'appartenance d'une donnée z_k quelconque étant égale à 1, la somme de tous les éléments d'une même colonne vaut par conséquent 1 :

$$\sum_{i=1}^c \mu_{ik} = 1, \quad \forall k \in \{1, 2, \dots, N\}, \quad (1.4)$$

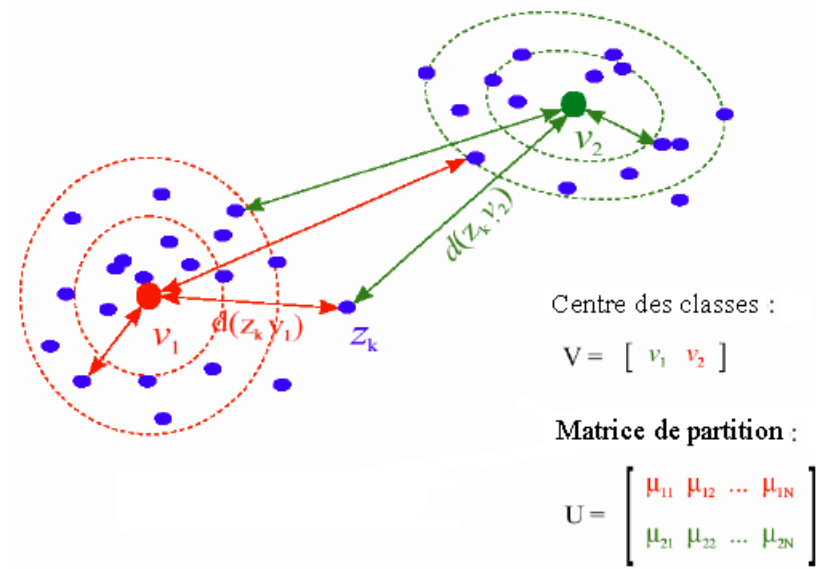


FIGURE 1.3 – Principe de la classification floue

Dans la figure (Fig.1.3), l'ensemble de données est partitionné en deux classes de centres v_1 et v_2 . La matrice de partition rassemble les degrés d'appartenance des différents points aux deux classes. La donnée z_k est à une distance $d(z_k, v_1)$ de la première classe et à $d(z_k, v_2)$ de la deuxième. Du fait que $d(z_k, v_1) < d(z_k, v_2)$, le degré d'appartenance μ_{1k} sera donc supérieur à μ_{2k} .

1.3.2 Algorithmes de classification floue

Ces algorithmes ont comme paramètre d'entrée le nombre de classe. Ils partagent l'ensemble de N objets en c groupes, la similarité à l'intérieur d'un même groupe est élevée, mais faible entre les différentes classes. Pour ce faire, ces algorithmes itèrent en deux étapes, d'abord ils

calculent les centres des groupes et ensuite ils assignent chaque objet au centre le plus proche. Chaque classe est caractérisée par le centre et par ses éléments.

Les algorithmes de classification flou optimisent itérativement un critère de classification afin d'établir une partition floue d'un ensemble de données en un certain nombre de classes. Le groupement des données est fait à partir d'une phase d'apprentissage en utilisant une mesure de similitude par l'intermédiaire de techniques de classification.

Dans cette section, nous allons présenter les deux méthodes les plus utilisées dans le domaine de l'identification de structure des systèmes : les FCM (C-moyenne floue) classiques développés par Bezdek [Bezdek, 1981] et l'algorithme proposé par Gustafson-Kessel (GK) [Gustafson et Kessel, 1978]. La première méthode permet la détection de classes hypersphériques, tandis que la deuxième détecte des classes hyperellipsoïdales, typiquement mieux adaptées à la géométrie des observations.

1.3.2.1 Algorithme des c-moyennes floues(FCM)

L'algorithme (FCM), issu des travaux de Dunn [Dunn, 1974] et amélioré plus tard par Bezdek [Bezdek, 1981], constitue une référence parmi les différentes méthodes de classification floue basées sur la minimisation de la fonction objectif de la forme :

$$J_{FCM}(Z; U, V) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m D_{ikA}^2 \quad (1.5)$$

où

Z est l'ensemble des données,

$U = [\mu_{ik}]$ est la matrice de partition floue (de dimension $c \times N$),

$V = [v_1, v_2, \dots, v_c]$ est le vecteur de centre de classes qui doit être déterminé, avec $v_i \in \mathbb{R}^n$ centre de la i -ème classe, $1 < i < c$.

$D_{ikA}^2 = \|z_k - v_i\|_A^2 = (z_k - v_i)^T A (z_k - v_i)$, $1 \leq i \leq c$, $1 \leq k \leq N$ est une norme de distance quadratique dans l'espace considéré, qui définit la mesure de distance entre l'observation z_k et le centre v_i au sens de la métrique induite par A .

$m \in [1, \infty]$ est un facteur qui désigne le degré de flou de la partition.

Dans l'équation (1.5), la mesure de non-similarité exprimée par le terme $J_{FCM}(Z; U, V)$ est la somme des carrés des distances entre chaque vecteur de données z_k et le centre v_i de la classe correspondante. L'effet de cette distance est pondéré par le degré d'activation (μ_{ik}^m) correspondant au vecteur de données z_k .

La valeur de la fonction coût $J_{FCM}(Z; U, V)$ peut être vue comme une mesure de la variance totale de z_k par rapport aux centres v_i .

La minimisation de la fonction objective (1.5) nous donne :

$$\mu_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{D_{ikA}}{D_{jkA}} \right)^{\frac{2}{m-1}}} \quad 1 \leq i \leq c, \quad 1 \leq k \leq N \quad (1.6)$$

$$v_i = \frac{\sum_{k=1}^N (\mu_{ik})^m z_k}{\sum_{k=1}^N (\mu_{ik})^m} \quad 1 \leq i \leq c \quad (1.7)$$

Le principe de l'algorithme est donné par :

Etant donné l'ensemble de données Z , choisir un nombre de classe $1 < c < N$, l'exposant $m > 1$, la tolérance d'arrêt $\varepsilon > 0$ et la matrice de norme A . Initialiser aléatoirement la matrice de partition U :

Répéter pour $l = 0, 1, 2, \dots$

Etape 1 : Calculer les centres des classes :

$$v_i^l = \frac{\sum_{k=1}^N (\mu_{ik}^{(l)})^m z_k}{\sum_{k=1}^N (\mu_{ik}^{(l)})^m} \quad 1 \leq i \leq c$$

Etape 2 : Calculer les distances :

$$D_{ikA_i}^2 = (z_k - v_i^{(l)})^T A (z_k - v_i^{(l)}) \quad 1 \leq i \leq c, 1 \leq k \leq N$$

Etape 3 : Mettre à jour la matrice de partition : Si $D_{ikA} > 0$ pour $1 \leq i \leq c, \quad 1 \leq k \leq N$

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^c \left(\frac{D_{ikA}}{D_{jkA}} \right)^{\frac{2}{m-1}}}$$

Autrement

$$\mu_{ik}^{(l)} = 0 \text{ si } D_{ikA} < 0, \text{ et } \mu_{ik}^{(l)} \in [0, 1] \text{ avec } \sum_{i=1}^c \mu_{ik}^{(l)} = 1$$

Jusqu'à $\|U^{(l)} - U^{(l-1)}\| < \varepsilon$

La forme des classes est déterminée par le choix de la matrice A dans l'équation. Le choix particulier $A = I$, induit la norme Euclidienne standard :

$$D_{ikA}^2 = (z_k - v_i)^T (z_k - v_i) \quad (1.8)$$

dans ce cas les classes détectées ont des formes sphériques.

1.3.2.2 Algorithme de Gustafson-Kessel(GK)

En 1979, Gustafson et Kessel [Gustafson et Kessel, 1978] ont généralisé l'algorithme FCM en employant une norme de distance adaptative dans le but de détecter des classes de différentes formes géométriques dans un ensemble de données. Dans ce cas, chaque classe possède sa propre matrice de norme, ce qui entraîne :

$$D_{ikA}^2 = \| z_k - v_i \|_A^2 = (z_k - v_i)^T A_i (z_k - v_i) \quad (1.9)$$

On suppose que la matrice A_i vérifie l'hypothèse :

$$|A_i| = \rho_i, \rho_i > 0 \quad (1.10)$$

où ρ_i est fixé pour chaque classe.

Dans ce cas, l'optimisation de (1.5) nous donne l'expression suivante pour A_i :

$$A_i = [\rho_i \det(F_i)]^{\frac{1}{n}} F_i^{-1} \quad (1.11)$$

où F_i est la matrice de covariance floue de la i -ième classe donnée par :

$$F_i = \frac{\sum_{k=1}^N (\mu_{ik})^m (z_k - v_i)(z_k - v_i)^T}{\sum_{k=1}^N (\mu_{ik})^m} \quad (1.12)$$

ce qui nous mène à l'algorithme de (GK) suivant :

Etant donné l'ensemble de données Z , choisir un nombre c de classes $1 < c < N$, l'exposant $m > 1$, la tolérance d'arrêt $\varepsilon > 0$ et la matrice de norme A . Initialiser aléatoirement la matrice de partition U :

Répéter pour $l = 1, 2, \dots$

Etape 1 : Calculer les centres des classes :

$$v_i^l = \frac{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m z_k}{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m} \quad 1 \leq i \leq c$$

Etape 2 : Calculer les matrices de covariances des classes :

$$F_i = \frac{\sum_{k=1}^N (\mu_{ik}^{l-1})^m (z_k - v_i^{(l)})(z_k - v_i^{(l)})^T}{\sum_{k=1}^N (\mu_{ik})^m} \quad 1 \leq i \leq c$$

Etape 3 : Calculer les distances :

$$D_{ikA_i}^2 = (z_k - v_i^{(l)})^T \left[(\rho_i \det(F_i))^{\frac{1}{n}} F_i^{-1} \right] (z_k - v_i^{(l)}) \quad 1 \leq i \leq c, 1 \leq k \leq N$$

Etape 4 : Mettre à jour la matrice de partition :

Si $D_{ikA_i} > 0$ pour $1 \leq i \leq c$, $1 \leq k \leq N$

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^c \left(\frac{D_{ikA_i}}{D_{jkA_i}} \right)^{\frac{2}{m-1}}}$$

Autrement

$$\mu_{ik}^{(l)} = 0, \quad \text{si } D_{ikA_i} < 0, \text{ et } \mu_{ik}^{(l)} \in [0, 1] \quad \text{avec } \sum_{i=1}^c \mu_{ik}^{(l)} = 1$$

$$\text{Jusqu'à } \|U^{(l)} - U^{(l-1)}\| \leq \epsilon$$

L'avantage de l'algorithme de GK par rapport à l'algorithme FCM est sa capacité de détecter des classes possédant des formes et des orientations différentes dans un seul ensemble de données comme le montre la figure (Fig.1.4).

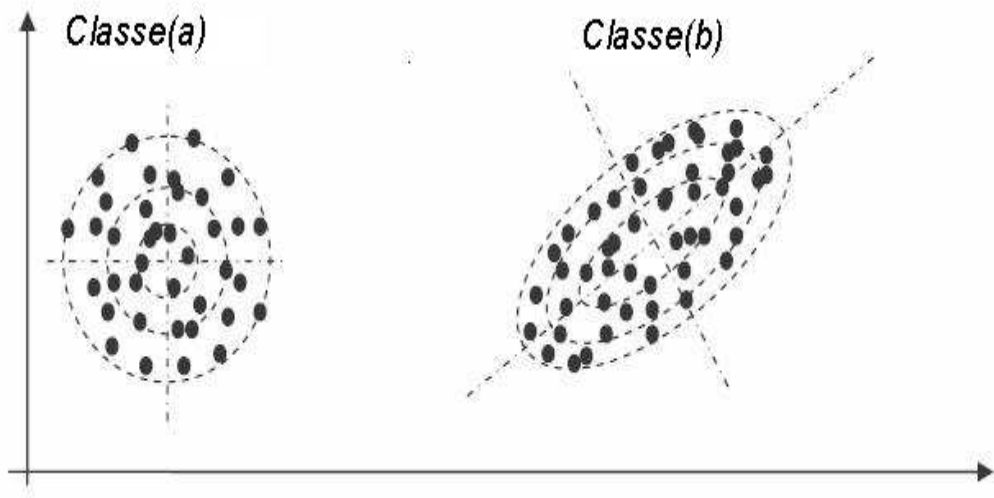


FIGURE 1.4 – (a) Classe sphérique détectable par les algorithmes FCM et GK, (b) Classe Elliptique détectable seulement par l'algorithme GK

1.3.3 Construction de modèles TS à partir des données

Dans le cadre de la modélisation floue des systèmes, l'intérêt porte sur l'obtention des modèles Takagi-Sugeno qui permettent une décomposition d'un système non linéaire en un ensemble de sous systèmes linéaires. Nous abordons ainsi une description d'une méthodologie générale pour la construction des modèles flous du type TS, en mettant l'accent sur les besoins communs qui sont : la génération des fonctions d'appartenance et l'obtention des paramètres des conséquents.

1.3.3.1 Sélection de la structure

L'objectif de cette étape est la détermination des entrées et sorties prépondérantes par rapport au but final de la modélisation. Quand il s'agit de l'identification des systèmes dynamiques, il faut choisir la structure et l'ordre du modèle dynamique. Pour le cas des systèmes non linéaires, cela s'effectuera en deux étapes :

- Une étude hors ligne pour déterminer le modèle linéaire.
- Une étude en ligne pour adapter les paramètres du modèle.

1.3.3.2 Classification des données

La sélection de la structure conduit à un problème de régression non linéaire statique, qui est alors approximée par une collection de sous modèles linéaires locaux. La localisation et les paramètres des sous modèles sont établis en partitionnant les données disponibles en classes. Chacune des classes définit une région floue dans laquelle le système peut être approximé localement par un sous modèle linéaire. Dans le cas des algorithmes (FCM), l'estimation des paramètres des sous modèles linéaires affines fait parti du processus de classification. Par contre, dans le cas de l'algorithme de GK, l'obtention des paramètres des conséquents du modèle TS correspondant est faite lors d'une étape postérieure au processus de la classification. D'autre part, il faut remarquer aussi que pour ces algorithmes, l'utilisateur doit définir à l'avance le nombre c de classe. Ce nombre reste constant pendant toute la durée d'exécution de l'algorithme.

1.3.3.3 Génération des fonctions d'appartenance des antécédents

Les fonctions d'appartenance des antécédents peuvent être obtenues en calculant les degrés d'appartenance directement dans l'espace produit des variables de l'antécédent. Elles peuvent aussi être extraites à partir de la matrice de partition floue U en appliquant le mécanisme de projection sur ces variables.

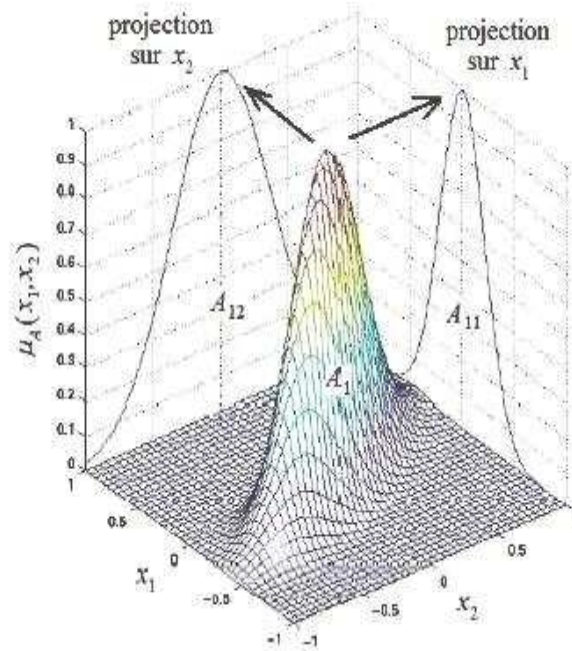


FIGURE 1.5 – Illustration du mécanisme de projection

Génération par projection

Le principe de cette méthode est de projeter pour chaque règle, les ensembles flous définis point par point dans la matrice de partition floue U sur les variables individuelles des antécédents. On projette donc la matrice de partition floue sur chacun des axes des variables (i.e., sur les régresseurs).

La figure (Fig.1.5) illustre le mécanisme de projection d'un ensemble flou A_1 de dimension 2, sur les deux axes des antécédents x_1 et x_2 .

Génération par Paramétrisation

Une définition point par point de l'ensemble flou A_{ij} est obtenue en projetant la i -ième ligne μ_i de la matrice de partition floue U sur la variable x_j de l'antécédent. Afin d'obtenir un modèle dans un but de prédiction ou de commande, les fonctions d'appartenance de l'antécédent doivent être exprimées sous une forme qui permet le calcul des degrés d'appartenance, même pour des données d'entrée non contenues dans l'ensemble des données Z . Cela est réalisé en approximant la fonction d'appartenance définie point par point par une fonction paramétrique appropriée [Babuska, 1998].

La figure (Fig.1.6) montre le mécanisme de projection pour la cas d'un ensemble de données partitionné en deux classes de centres v_1 et v_2 .

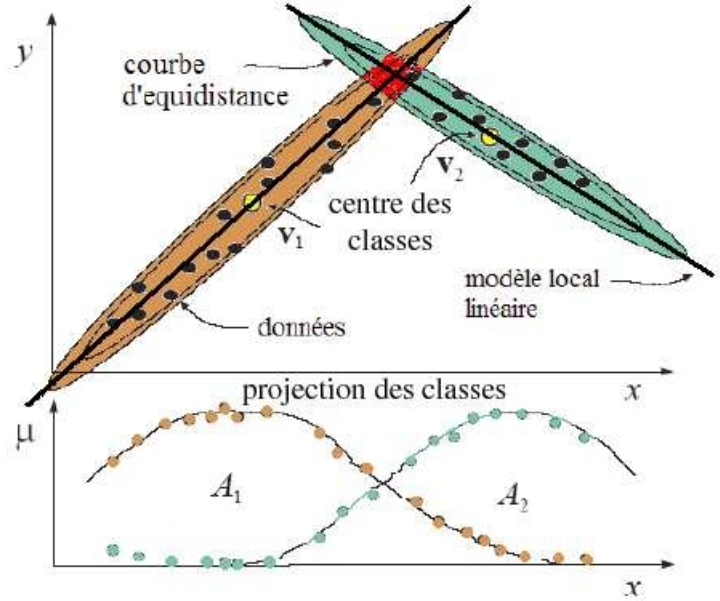


FIGURE 1.6 – Approximation des données projetées par une fonction d'appartenance paramétrique.

1.3.3.4 Obtention des paramètres des conséquents

Les paramètres des conséquents peuvent être établis par la technique des moindres carrés, en utilisant comme facteurs de pondération des données les degrés d'appartenance de la matrice de partition floue U issus du processus de classification. Cette approche conduit à une formulation de c problèmes indépendants de type moindres carrés pondérés dans laquelle les degrés d'appartenance expriment l'importance de la paire de données (x_k, y_k) par rapport à chaque i -ème sous modèle linéaire local $y_i = a_i + d_i$, avec $1 \leq i \leq c$.

Les données d'identification entrée-sortie $\mathbf{Z}_k = [\mathbf{z}_k^T, y_k]^T$, avec $1 \leq k \leq N$, et les degrés d'appartenance μ_{ik} de la matrice de partition floue sont regroupés dans les matrices suivantes :

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \quad \mathbf{W}_i = \begin{bmatrix} \mu_{i1} & 0 & \dots & 0 \\ 0 & \mu_{i2} & \dots & 0 \\ \vdots & \vdots & & 0 \\ 0 & 0 & \dots & \mu_{iN} \end{bmatrix} \quad (1.13)$$

Les paramètres des conséquents $\mathbf{a}_i$ et d_i appartenant à la règle correspondant à la i -ème classe sont concaténés dans un seul vecteur de paramètres θ_i , donné par :

$$\theta_i = [\mathbf{a}_i^T, d_i]^T \quad (1.14)$$

Afin de faciliter le calcul, la matrice de régression $\mathbf{X}$ est augmentée en ajoutant un vecteur-colonne unitaire, selon l'expression :

$$\mathbf{X}_e = [\mathbf{X}, \mathbf{1}] \quad (1.15)$$

Si les colonnes de $\mathbf{X}_e$ sont linéairement indépendantes et $\mu_{ik} > 0$ pour $1 \leq k \leq N$, alors la solution des moindres carrés de $\mathbf{y} = \mathbf{X}_e \boldsymbol{\theta} + \boldsymbol{\varepsilon}$, où la k -ème paire de données (x_k, y_k) est pondérée par μ_{ik} , est donnée finalement par l'expression :

$$\theta_i = (\mathbf{X}_e^T \mathbf{W}_i \mathbf{X}_e)^{-1} \mathbf{X}_e^T \mathbf{W}_i \mathbf{y} \quad (1.16)$$

Les paramètres $\mathbf{a}_i$ et d_i sont donnés respectivement par :

$$\mathbf{a}_i = [\theta_1, \theta_2, \dots, \theta_p], \quad d_i = \theta_{p+1} \quad (1.17)$$

1.3.4 Modèle flou de Takagi-Sugeno

Le modèle flou Takagi et Sugeno (TS) [Takagi et Sugeno, 1985] est structuré comme une interpolation de systèmes linéaires. Il est prouvé que les modèles flous TS sont des approxima-teurs universels [Buckley, 1992][Castro, 1995]. Le modèle de TS est, comme celui de Mam-dani, construit à partir d'une base de règles « Si... Alors... », dans laquelle si la prémisse est toujours exprimée linguistiquement, le conséquent utilise des variables numériques plutôt que des variables linguistiques. Le conséquent peut s'exprimer par exemple, sous la forme d'une constante, d'un polynôme ou de manière plus générale d'une fonction ou d'une équation diffé-rentielle dépendant des variables associées à l'antécédent. D'une manière générale, un modèle de type TS est basé sur une collection des règles R_i du type :

$$R_i: \text{ Si } x \text{ est } A_i \text{ Alors } y_i = f_i(x) \quad i = 1, \dots, r \quad (1.18)$$

où R_i dénote la i -ème règle du modèle et r est le nombre de règles que contient la base de règles. $x \in \mathbb{R}^p$ est la variable d'entrée (*antécédent*) et $y \in \mathbb{R}$ est la variable de sortie (*conséquent*). A_i est le sous-ensemble flou de l'antécédent de la i -ème règle, définie, dans ce cas, par une fonction d'appartenance multivariable de la forme :

$$\mu_{A_i}(x): \mathbb{R}^p \rightarrow [0, 1] \quad (1.19)$$

Comme dans le modèle linguistique, la proposition de l'antécédent « x est A » est nor-malement exprimée comme une combinaison logique de propositions simples avec des sous-

ensembles flous unidimensionnels définis pour les composants individuels du vecteur x , usuellement dans la forme conjonctive suivante :

$$R_i : \text{ Si } x_1 \text{ est } A_{i1} \text{ et } x_2 \text{ est } A_{i2} \text{ et } \dots \text{ et } x_p \text{ est } A_{ip} \text{ Alors } y_i = f_i(x) \quad i = 1, \dots, r \quad (1.20)$$

Typiquement les fonctions f_i sont choisies comme des fonctions paramétrées appropriées, avec la même structure pour chaque règle où seuls les paramètres varient. Une forme de paramétrisation souvent utilisée est la forme affine, donnée par :

$$y_i = a_i^T x + d_i \quad (1.21)$$

où $a_i \in \mathbb{R}^p$ est un vecteur de paramètres et d_i est un scalaire. Ce modèle est appelé le modèle affine TS. Les conclusions des règles dans ce modèle sont alors des hyperplans (sous-espaces linéaires p -dimensionnels) dans l'espace $\mathbb{R}^{p+1}$.

Ainsi, en modélisation floue des systèmes, l'antécédent de chaque règle définit une région floue de validité pour le sous modèle correspondant du conséquent. Le modèle global est composé par la concaténation des modèles locaux (linéaires) et peut être vue comme une approximation par morceaux d'une surface non linéaire correspondant à la sortie du système.

Avant de pouvoir inférer la sortie, il faut calculer d'abord le degré d'accomplissement $\omega_i(x)$ de l'antécédent. Pour les règles avec des sous-ensembles flous multivariables dans l'antécédent, le degré d'accomplissement est simplement égal au degré d'appartenance de l'entrée multidimensionnelle x ; c'est-à-dire $\omega_i = \mu_{A_i}(x)$. Quand des relations logiques sont utilisées, le degré d'accomplissement de l'antécédent est calculé comme une combinaison des degrés d'appartenance des propositions individuelles en utilisant les opérateurs de la logique floue.

Dans la modélisation TS, l'obtention de la sortie du modèle est réalisée à partir d'une combinaison des opérations d'inférence et de défuzzification. La sortie finale se calcule comme la moyenne des sorties correspondants aux règles R_i , pondérées par le degré d'accomplissement normalisé, selon l'expression [Takagi et Sugeno, 1985] :

$$y = \frac{\sum_{i=1}^r \omega_i(x) y_i}{\sum_{i=1}^r \omega_i} \quad (1.22)$$

En notant μ_i le degré d'accomplissement normalisé conformément à l'expression :

$$\mu_i = \frac{\omega_i(x) y_i}{\sum_{i=1}^r \omega_i} \quad (1.23)$$

le modèle affine TS, avec une structure commune du conséquent, peut être exprimé comme

un modèle linéaire avec des paramètres dépendants des entrées :

$$y = \left(\sum_{i=1}^r \mu_i(x) a_i^T \right) x + \sum_{i=1}^r \mu_i(x) d_i = a^T(x) + d(x) \quad (1.24)$$

Dans le cas des modèles dynamiques affines Takagi-Sugeno en temps discret sous la forme non linéaire Auto-Régressive avec entrée exogène (NARX), la relation entre les valeurs précédentes des entrées et sorties avec la sortie prédite $\hat{y}(k+1)$ est établie de la manière suivante :

$$\hat{y}(k+1) = f(y(k), \dots, y(k-n_y+1), u(k), \dots, u(k-n_u+1)) \quad (1.25)$$

où k dénote l'instant d'échantillonnage, n_y et n_u sont des entiers liés à l'ordre du système. Une fois que la structure appropriée est établie, la fonction f peut être approximée en utilisant une régression statique non linéaire, correspondant dans notre cas au modèle flou de type TS.

Dans le modèle NARX, Le vecteur de régression est composé par un certain nombre d'entrées et de sorties précédentes :

$$x = [y(k), \dots, y(k-n_y+1), u(k), \dots, u(k-n_u+1)]^T$$

avec $\hat{y}(k+1)$ est la sortie prédite. Pour le cas des systèmes avec retard pur entre l'entrée et la sortie peut être directement incorporé sur le vecteur de régression sous la forme donnée par :

$$x = [y(k), \dots, y(k-n_y+1), u(k-n_d+1), \dots, u(k-n_d-n_u+2)]^T$$

où n_d est une valeur entière du retard en nombre d'échantillons. Par simplicité, nous prenons par la suite $n_d = 1$.

Dans ce cas, les règles du modèle dynamique TS prennent la forme :

$$\begin{aligned} R_i : & \text{ Si } y(k) \text{ est } A_{i1} \text{ et } y(k-1) \text{ est } A_{i2} \text{ et } \dots \text{ et } y(k-n_y+1) \text{ est } A_{in_y} \\ & \text{ et } u(k) \text{ est } B_{i1} \text{ et } u(k-1) \text{ est } B_{i2} \text{ et } \dots \text{ et } u(k-n_u+1) \text{ est } B_{in_u} \\ & \text{ Alors } y_i(k+1) = \sum_{j=1}^{n_y} a_{ij} y(k-j+1) + \sum_{j=1}^{n_u} b_{ij} u(k-j+1) + d_i \quad i = 1, \dots, r \end{aligned} \quad (1.26)$$

La sortie globale du modèle est calculée à partir de l'expression :

$$y(k+1) = \sum_{i=1}^r \omega_i(k) y_i(k+1) \quad (1.27)$$

Dans laquelle $\omega_i(k) \in [0, 1]$ est le degré d'accomplissement normalisé et $\sum_{i=1}^r \omega_i(k) = 1$.

En plus de la représentation des structures entrée-sortie les plus fréquentes, les modèles

flous dynamiques de type Takagi-Sugeno peuvent aussi représenter les systèmes non linéaires sous la forme d'espace d'état.

Un exemple d'une représentation basée sur des règles d'un système non linéaire dans l'espace d'état continu est le modèle dynamique Takagi-Sugeno suivant :

La $i^{\text{ème}}$ règle du modèle s'écrit :

$$R_i : \text{ Si } z_1 \text{ est } F_{i1} \text{ et } \dots \text{ et } z_p \text{ est } F_{ip} \text{ Alors } \begin{cases} \dot{x}(t) = A_i x(t) + B_i u(t) \\ y(t) = C_i(t) \end{cases} \quad i = 1, \dots, r \quad (1.28)$$

F_{ij} est l'ensemble flou. $x(t) \in \mathbb{R}^n$ représente le vecteur d'état, $u(t) \in \mathbb{R}^m$ est le vecteur des commandes, $y(t) \in \mathbb{R}^q$ est le vecteur de sortie du système, $A_i \in \mathbb{R}^{n \times n}$ est la matrice d'état, $B_i \in \mathbb{R}^{n \times m}$, est la matrice d'entrée du système, $C_i \in \mathbb{R}^{q \times n}$ est la matrice de sortie et $z_1(t) \dots z_p(t)$ sont les prémisses fonctions de l'état.

Pour une paire $(y(t), u(t))$ donnée, l'inférence du système flou est donnée par :

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^r \mu_i(z(t))(A_i x(t) + B_i u(t)) \\ y(k) = \sum_{i=1}^r \mu_i(z(t)) C_i x(t) \end{cases} \quad (1.29)$$

1.4 Conception d'une commande floue

La théorie d'approximation universelle garantit la possibilité de modéliser la dynamique des systèmes non linéaires par des systèmes flous. Ces derniers sont utilisés pour concevoir des contrôleurs flous. Reste qu'une des propriétés fondamentales de la commande, à savoir la stabilité, doit être assurée.

Différentes méthodes ont été proposées pour élaborer la stratégie de commande adéquate. Le choix de la commande dépend du système et son interaction avec l'environnement. Parmi les commandes les plus utilisées on retrouve : commande par modèle de référence, par placement de pôles, par minimisation d'un critère quadratique ainsi que celles que nous avons utilisées dans ce mémoire, à savoir, par mode glissant, les méthodes H_∞ et LMI [Lagrat *et al.*, 2006], commande décentralisée [Errahmani *et al.*, 2007], et la commande adaptative [Lagrat *et al.*, 2008][Lagrat *et al.*, 2007a].

La figure (Fig.1.7) illustre le principe d'intégration d'un contrôleur flou pour la commande d'un procédé non linéaire. La description des blocs constituant le contrôleur flou est décrite dans la section(cf.1.2.5).

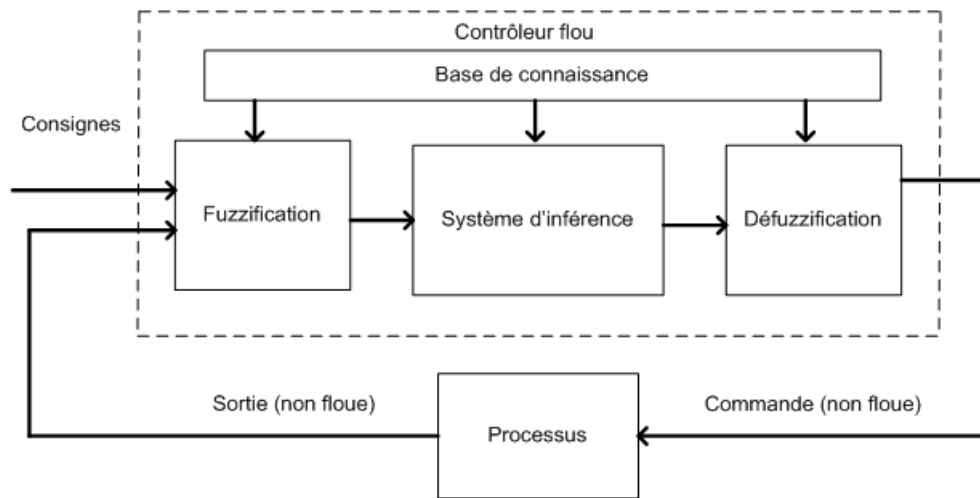


FIGURE 1.7 – Structure générale de la commande d'un processus par un contrôleur flou.

1.4.1 Commande adaptative

La commande adaptative permet de maintenir les performances quand la dynamique du système à commander varie dans le temps. Les paramètres sont adaptés de manière à poursuivre le système dans son évolution [Babuska et Verbruggen, 1997][Kim *et al.*, 1997].

Il est à noter que les propriétés de stabilité, de convergence et de robustesse des algorithmes d'adaptation doivent être convenablement choisies en fonction des circonstances et de l'environnement dans lequel s'effectue l'expérience [Landau et Karimi, 1997][Lagrat *et al.*, 2007a].

Deux types de commande sont possibles :

- *La commande indirecte*

Cette approche s'effectue en deux étapes :

- Estimation des paramètres du modèle,
- Calcul des paramètres du contrôleur à partir des paramètres estimés.

- *La commande directe*

Elle conduit directement à l'estimation des paramètres de la loi de commande. Ce type de schéma est obtenu en réécrivant le modèle du processus en fonction des paramètres de la loi de commande.

1.4.2 Commande par l'approche mode de glissement

La mise en oeuvre de la commande par mode glissant est basée sur trois étapes. La première consiste à choisir la surface de glissement qui permet d'assurer la convergence, la deuxième établit la condition d'existence du mode de glissement et la dernière étape détermine la loi de

commande qui aura pour rôle de maintenir les conditions de glissement (attractivité).

- *Choix de la surface de glissement*

Soit la surface de glissement $\sigma(\underline{x}, t) = 0$ définie dans l'espace d'état R^n par :

$$\sigma(\underline{x}, t) = e^{(n-1)} + k_{n-1}e^{(n-2)} + \dots + k_2\dot{e} + k_1e \quad (1.30)$$

Les coefficients k_i , $i = 1, 2, \dots, n-1$, sont choisis de telle sorte que toutes les racines du polynôme $S = s^{n-1} + k_{n-1}s^{n-2} + \dots + k_1s$, se trouvent dans le demi-plan gauche de l'espace complexe afin de garantir la convergence asymptotiquement vers zéro de l'erreur de poursuite $e(t)$. Le problème de la poursuite nécessite la conception d'une loi de commande qui assure : $\sigma(\underline{x}, t) = 0$ pour $t \geq 0$.

- *Condition d'existence d'une surface de glissement*

L'étude de l'existence du mode de glissement est basée sur la méthode de Lyapunov. Afin de garantir l'attractivité de la surface $\sigma(\underline{x}, t) = 0$, on considère une fonction de Lyapunov V définie positive :

$$V = \frac{1}{2}\sigma^2 \quad (1.31)$$

Une condition suffisante pour que le système soit stable est :

$$\dot{V} = \sigma \cdot \dot{\sigma} \leq -\eta|\sigma| \quad (1.32)$$

où $\eta > 0$

- *Synthèse de la loi de commande*

La loi de commande par mode de glissement est donnée par l'équation suivante :

$$u_s = -k_d \text{sign}(\sigma) \quad (1.33)$$

k_d est une constante positive qui représente le gain de la commande.

La fonction $\text{sign}(\sigma)$ est définie comme suit :

$$\text{sign}(\sigma) = \begin{cases} 1 & \text{Si } \sigma > 0 \\ 0 & \text{Si } \sigma = 0 \\ -1 & \text{Si } \sigma < 0 \end{cases} \quad (1.34)$$

1.5 Conclusion

Dans ce chapitre, nous avons donné un aperçu général sur les fondements de la logique floue tout en décrivant les algorithmes permettant la modélisation d'un système non linéaire, affectés

de perturbations et qui peut être à dynamique inconnues, par le modèle de TS. Une extension au cas multivariable a été présentée accompagnée par le principe de détermination des fonctions d'appartenance. Par la suite, un tour d'horizon sur les approches de classification des données a été présenté, ce qui nous permettra au niveau du chapitre 3, de faire des comparaisons avec notre contribution. Enfin, l'accent a été mis sur la commande floue des systèmes non linéaires et les hypothèses restrictives l'accompagnant. Tous ces outils, serviront pour la description des travaux que nous allons aborder dans la suite.

Identification des Systèmes Non Linéaires à base de Modèles de type TS

2.1 Introduction

Les modèles de Takagi-Sugeno (TS) sont souvent utilisés pour approximer les systèmes non linéaires. Les algorithmes d'identification par les moindres carrés combinés aux techniques de classification des données entrées sorties sont à la base de cette approximation surtout lorsque l'information sur le système non linéaire n'est pas disponible ou mal connue.

Dans ce chapitre, nous présentons deux stratégies de classification :

La première concerne les systèmes MIMO. Il s'agit en fait, de décomposer le système MIMO en un sous ensemble de systèmes MISO, ce qui permettra de réduire la complexité de l'approche globale en tenant compte de tous les effets perturbateurs, à savoir ; le couplage croisé, les non linéarités et les perturbations externes. L'application de cette technique au cas monovariante est directe et ne pose aucun problème.

La modélisation floue est donc la représentation du comportement du système en utilisant les concepts de la logique floue. Il s'agit d'un ensemble de règles Si-Alors, de chaque règle résulte un modèle localement linéaire et le modèle global est obtenu par agrégation des modèles locaux. Il est à noter que dans le cas de Takagi-Sugeno, les conséquences des règles sont des fonctions ordinaires.

La deuxième est introduite dans le sens de réduire le nombre de classes et donc de modèles locaux. Cette réduction entraînera automatiquement une réduction du nombre de paramètres à estimer et par conséquent nous permettra d'éviter la lourde charge des calculs. Cependant, La détermination du nombre optimal de classes se fait d'une façon expérimentale tout en effectuant un ensemble de tests comparatifs sur le facteur qualité de la classification et sur l'erreur de

modélisation. La stratégie proposée est simple et ne nécessite pas de passer par les approches classiques de validation. Elle consiste à représenter d'abord l'ensemble des données par une fonction polynomiale décrivant l'allure approchée de ces données. Le degré optimal de cette fonction est facilement déterminé à partir de la formule proposée. L'algorithme d'optimisation du nombre de classe que nous proposons [Ouakka et Boumhidi, 2009](Annexe F) est décrit en détail dans la suite de ce chapitre et les résultats de simulation obtenus, pour une classe des systèmes non linéaires monovariés, montre bien qu'ils répondent à l'objectif principal de recherche d'une structure optimale modélisant ce type de systèmes.

2.2 Identification d'un système MIMO

Afin de réduire la complexité des systèmes MIMO (Multi entrées multi sorties), la décomposition de ces derniers en sous systèmes MISO (Multi entrées mono sortie) est une approche qui permet d'une part d'éviter la lourde charge de l'approche globale et d'autre part la facilité de mise en oeuvre des schémas de commande de ces systèmes [Lagrat *et al.*, 2007b].

Par leur qualité d'approximateur universel, l'identification floue à base de modèles de Takagi-Sugeno, permet de transformer ces systèmes non linéaires en un ensemble de systèmes linéaires.

2.2.1 Représentation des systèmes MISO

Considérons un système MIMO à n entrées et n sorties. Après décomposition, chaque système MISO sera décrit par :

$$y_i(k+1) = f_i(x(k)), \quad i = 1, 2, \dots, n. \quad (2.1)$$

avec le vecteur de régression donné par :

$$x_i(k) = [\{y_i(k)\}_0^n, \{u_i(k-d_{ii})\}_0^n, \{u_1(k-d_{1i})\}_0^n, \dots, \{u_n(k-d_{ni})\}_0^n]$$

où k dénote l'échantillon du temps discret, n est un entier relatif à l'ordre du système et d_{ij} est le retard pur.

Les systèmes MISO sont identifiés indépendamment, pour des raisons de simplification de la notation, l'index i sera ignoré.

La sortie du système s'écrit :

$$y(k+1) = Ay(k) + Bu(k) + \alpha \quad (2.2)$$

α est une constante.

La fonction inconnue $f(\cdot)$ est approximée par un modèle flou de Takagi-Sugeno. Par application de l'algorithme GK , l'espace des données est divisé en un ensemble de zones floues, décrites dans notre cas par des fonctions d'appartenance gaussiennes, tandis que les parties conséquentes décrivent le fonctionnement du système dans ces zones. La base de règle pour le système MISO devient :

$$R_l : \text{ Si } x(k) \text{ est } \Omega_l \text{ Alors } y^l(k+1) = A_l y(k) + B_l u(k) + \alpha_l, \quad l = 1, 2, \dots, K. \quad (2.3)$$

Ω_l est l'ensemble flou antécédent de la $l^{ème}$ règle, $A_l = [A_{l1}, \dots, A_{ln}]$ et $B_l = [B_{l1}, \dots, B_{ln}]$ sont les vecteurs des deux polynômes A_l et B_l , K est le nombre de règles.

La règle (2.3) peut alors s'écrire :

$$R_l : \text{ Si } x_1(k) \text{ est } \Omega_{l1} \text{ et } \dots \text{ et } x_p(k) \text{ est } \Omega_{lp} \text{ Alors } y^l(k+1) = A_l y(k) + B_l u(k) + \alpha_l, \quad l = 1, 2, \dots, K. \quad (2.4)$$

ou $p = 2n^2 + 1$.

La sortie du modèle TS est alors évaluée par :

$$y(k+1) = \frac{\sum_{i=1}^K \mu_i(x(k)) y^i(k+1)}{\sum_{i=1}^K \mu_i(x(k))} \quad (2.5)$$

En posant

$$\Phi_j(x, c_i, \sigma_i) = \frac{\mu_j(x(k))}{\sum_{i=1}^K \mu_i(x(k))} \quad (2.6)$$

où

$\Phi_j(x, c_i, \sigma_i)$ est la fonction de validation de la fonction gaussienne ayant comme paramètres les centres c_i et les variances σ_i .

$$\mu_i(x(k)) = \exp\left(\frac{-1}{2} \frac{(x_i - c_{i1})^2}{\sigma_{i1}^2}\right) \dots \exp\left(\frac{-1}{2} \frac{(x_p - c_{ip})^2}{\sigma_{ip}^2}\right) \quad (2.7)$$

La formule (2.5) devient :

$$y(k+1) = \sum_{i=1}^K y^i(k+1) \Phi_j(x, c_i, \sigma_i) \quad (2.8)$$

L'identification des systèmes MISO est généralement réalisée en deux étapes : la première s'effectue, hors ligne, pour déterminer les paramètres des antécédents (le cas d'une fonction gaussienne le centre c_i et la variance σ_i) et les paramètres linéaires des conséquents. La

deuxième étape, en ligne, réalise l'adaptation des paramètres des modèles locaux par l'algorithme des moindres carrés récurrents.

2.2.2 Identification hors ligne du modèle flou

L'ensemble des données, noté Z , est construit par la concaténation de la matrice de régression X et du vecteur régressant Y :

$$X = \begin{bmatrix} \dots \\ x(k) \\ \dots \\ x(N-1) \end{bmatrix}, \quad Y = \begin{bmatrix} \dots \\ y(k) \\ \dots \\ y(N-1) \end{bmatrix}, \quad Z^T = \begin{bmatrix} X & Y \end{bmatrix} \quad (2.9)$$

N est le nombre d'observations.

En appliquant la classification floue, l'ensemble de données Z sera partitionner en N_c sous ensembles flous. Il existe plusieurs algorithmes réalisant cette opération, le C-means, l'algorithme Gatha-Geva [Abonyi *et al.*, 2002] et l'algorithme (GK) que nous utiliserons dans la suite.

Les valeurs d'appartenance des données aux groupes seront décrites par une matrice de partition floue $U = [\mu_{ik}]_{N_c \times N}$, où $\mu_{ik} \in [0, 1]$ représente le degré d'appartenance de l'observation x_k au groupe i . Chaque groupe est caractérisé par un centre c_i , $C = [c_1, \dots, c_{N_c}]$ est le vecteur des centres. Et une matrice de covariance, $F = [F_1, \dots, F_{N_c}]$, décrivant la variance des données dans le groupe F_i .

Les fonctions d'appartenance type gaussienne choisies dans le cadre de ce chapitre, sont données par :

$$\Omega_{ij}(x_j(k)) = \exp \left(\frac{-1}{2} \frac{(x_j - c_{ij}^2)}{\sigma_{ij}^2} \right) \quad (2.10)$$

Les paramètres des conséquents $\theta_i = [A_i, B_i, C_i]$ sont estimés séparément par l'algorithme des moindres carrés récurrents en minimisant la fonction objective suivante :

$$\min_{\theta_i} \frac{1}{N} [Y - \xi \theta_i]^T Q_i [Y - \xi \theta_i] \quad (2.11)$$

où $\xi = [X \ 1]$ est la matrice de régression augmentée en ajoutant un vecteur-colonne unitaire et Q_i est une matrice contenant les valeurs des fonctions de validités Φ_i du ième local linéaire

modèle de chaque groupe de donnée.

$$Q_i = \begin{bmatrix} \Phi_i(x(1), c_i, \sigma_i) & 0 & \dots & 0 \\ 0 & \Phi_i(x(2), c_i, \sigma_i) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \Phi_i(x(N), c_i, \sigma_i) \end{bmatrix} \quad (2.12)$$

L'estimation par les moindres carrés des paramètres conséquents, $(\theta_i = [A_i, B_i, \alpha_i])$ est donnée par :

$$\theta_i = [\xi^T Q_i \xi]^{-1} \xi^T Q_i Y \quad (2.13)$$

2.2.3 Adaptation en ligne du modèle flou

Dans le cas des systèmes non linéaires multivariables, l'adaptation en ligne est nécessaire pour obtenir un modèle capable de poursuivre le système dans son évolution. Il en résulte un modèle TS décrit par :

$$y^i(k+1) = A_i(k)y(k) + B_i(k)u(k) + \alpha_i(k) \quad (2.14)$$

Pendant cette phase d'adaptation, les paramètres des antécédents restent fixes, seuls les paramètres conséquents sont adaptés.

Pour chaque $j^{ème}$ modèle localement linéaire, les nouveaux paramètres estimés $\theta_j(k)$ sont donnés à l'instant k par :

$$\theta_j(k) = \theta_j(k-1) + \delta_j(k)(y(k) - \xi^T(k)(\theta_j(k-1))) \quad (2.15)$$

$$\delta_j(k) = \frac{P_j(k-1)\xi(k)}{\xi^T(k)P_j(k-1)\xi(k) + \lambda / \Phi_j(\xi(k), c_j, \sigma_j)} \quad (2.16)$$

$$P_j(k) = \frac{1}{\lambda} [I - \delta_j(k)\xi^T(k)]P_j(k-1) \quad (2.17)$$

2.2.4 Simulation

Considérons un système MIMO à deux entrées et deux sorties décrit par :

$$\begin{aligned} A_1(q^{-1})y_1(t) &= q^{-d_{11}}B_{11}(q^{-1})Z_1(t) + q^{-d_{12}}B_{12}(q^{-1})Z_2(t) \\ A_2(q^{-1})y_2(t) &= q^{-d_{22}}B_{22}(q^{-1})Z_2(t) + q^{-d_{21}}B_{21}(q^{-1})Z_1(t) \end{aligned} \quad (2.18)$$

où q^{-1} est l'opérateur retard défini par : $q^{-1}x(t) = x(t-1)$.

$y(t) \in \mathbb{R}^2$ et $Z(t) \in \mathbb{R}^2$ sont respectivement, la sortie, l'entrée non linéaire et le vecteur perturbation.

Avec

$$A = [A_1 A_2]^T = \begin{bmatrix} 0.1q^{-1} \\ 0.2q^{-1} \end{bmatrix} \quad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} 1 + 0.5q^{-1} & 0.05q^{-1} \\ 0.3q^{-1} & 1 + 0.8q^{-1} \end{bmatrix}$$

$$d_{11} = 1, \quad d_{12} = 0, \quad d_{21} = 0, \quad d_{22} = 1$$

$$Z_1(t) = 2u_1(t) + u_1^2(t)$$

$$Z_2(t) = u_2^2(t)$$

d_{ii} et d_{ij} sont respectivement, les retards entre l'entrée $u_i(t)$ et la sortie $y_i(t)$, et entre l'entrée $u_j(t)$ et la sortie $y_i(t)$.

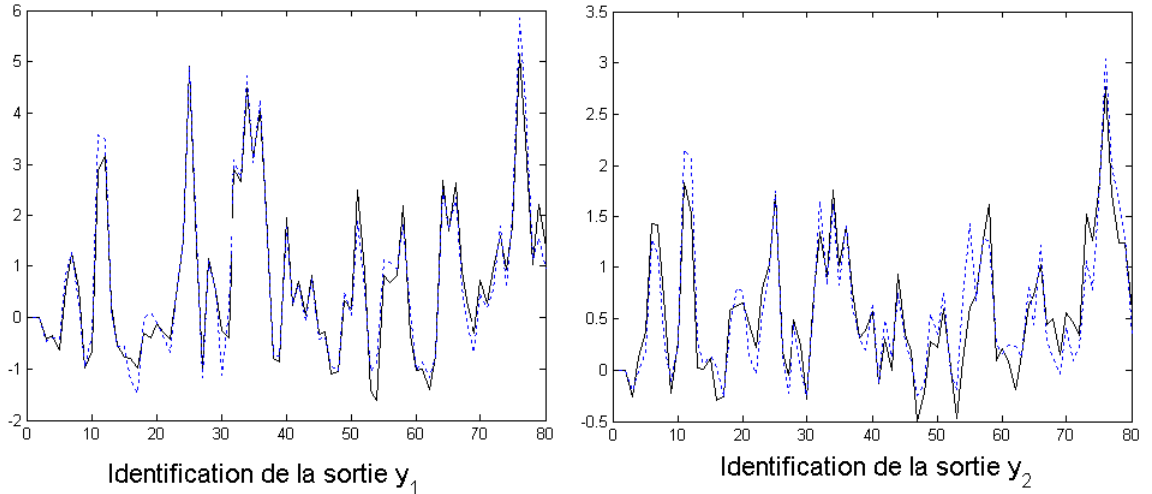


FIGURE 2.1 – Comparaison des réponses $y_1(t)$ et $y_2(t)$ (ligne continue) et les modèles flous correspondants(ligne discontinue)

La procédure d'identification est réalisée par l'ensemble de données des entrées $u_1(k)$ et $u_2(k)$, générées aléatoirement et uniformément réparties dans l'intervalle $[-1, 1]$. L'algorithme de classification initialisé par un nombre de classes $c = 3$, détermine les paramètres des antécédents (U, C, F) ainsi que les paramètres conséquents des modèles locaux linéaires. La figure (Fig.2.1) présente une comparaison entre les sorties du système et les modèles flous correspondants. On remarque que les deux modèles TS suivent correctement l'allure des sorties y_1 et y_2

La deuxième expérience consiste à tester la validation du modèle flou. Deux signaux carrée de période $T = 15$ ont été appliqués aux entrées $u_1(t)$ et $u_2(t)$. La figure (Fig.2.2) illustre les réponses obtenues par le système et le modèle flou identifié par l'algorithme GK sans adaptation des paramètres. On constate que l'erreur de modélisation entre les sorties est relativement

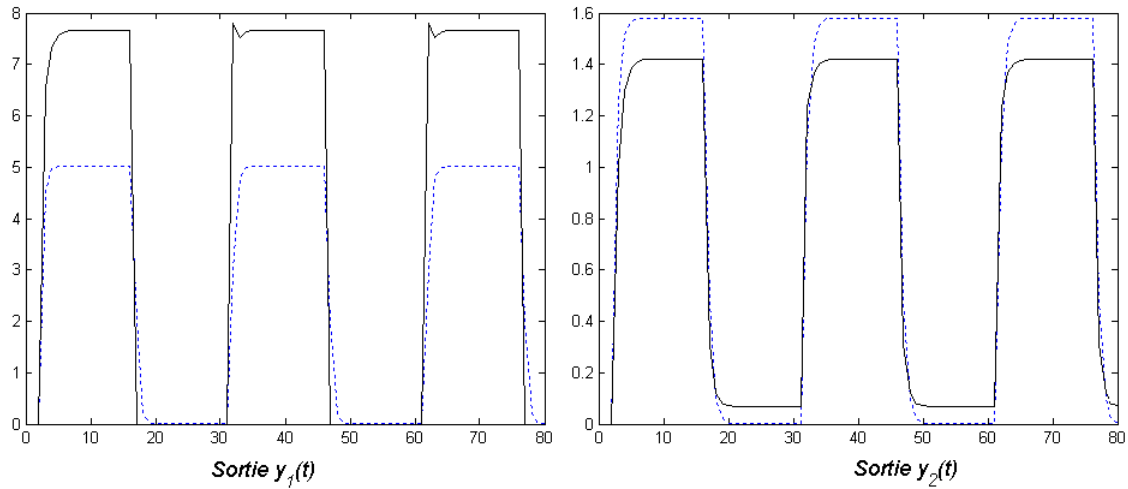


FIGURE 2.2 – Phase de validation : comparaison des réponses du système (ligne discontinue) et les modèles flous (ligne continue)

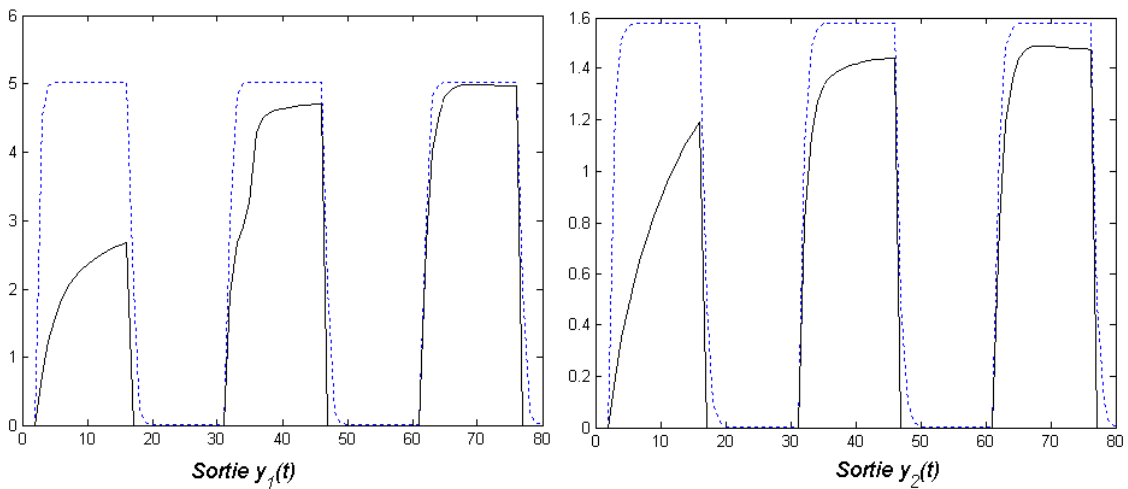


FIGURE 2.3 – Phase d'adaptation des paramètres : comparaison des réponses du système (ligne discontinue) et les modèles flous (ligne continue)

grande.

Afin de pallier à ce problème, nous avons procédé à l'adaptation des paramètres par l'application de l'algorithme des moindres carrés récurrents avec un facteur d'oubli $\lambda = 0.99$. La figure (Fig.2.3) montre qu'au bout d'un certain nombre d'itérations, les sorties des modèles flous suivent presque les sorties $y_1(t)$ et $y_2(t)$.

Les figures (Fig.2.4) et (Fig.2.5) donnent l'évolution des paramètres des modèles flous TS. On remarque que ces paramètres convergent vers les vraies valeurs après un certain nombre d'itérations qui correspond au régime transitoire.

La structure des deux sous systèmes MISO est définie de la façon suivante :

$$y_1(k) = a_{11}^1(k)y_1(k-1) + b_{21}^1(k)u_1(k-1) + b_{12}^1(k)u_1(k-2) + b_{22}^1(k)u_2(k-1) + C_{11}^1$$

$$y_2(k) = a_{21}^2(k)y_2(k-1) + b_{11}^2(k)u_1(k-1) + b_{12}^2(k)u_2(k-1) + b_{22}^2(k)u_2(k-2) + C_{11}^1$$

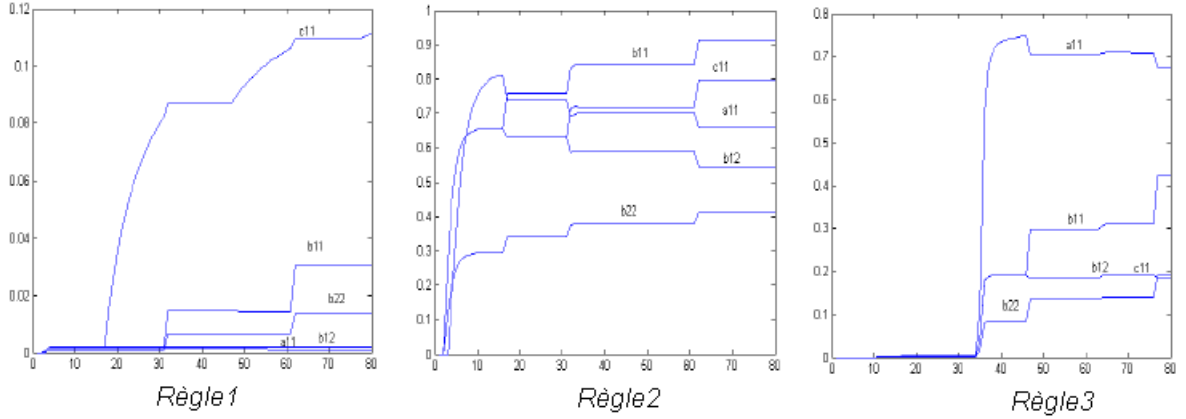


FIGURE 2.4 – Evolution des paramètres pour la sortie y_1

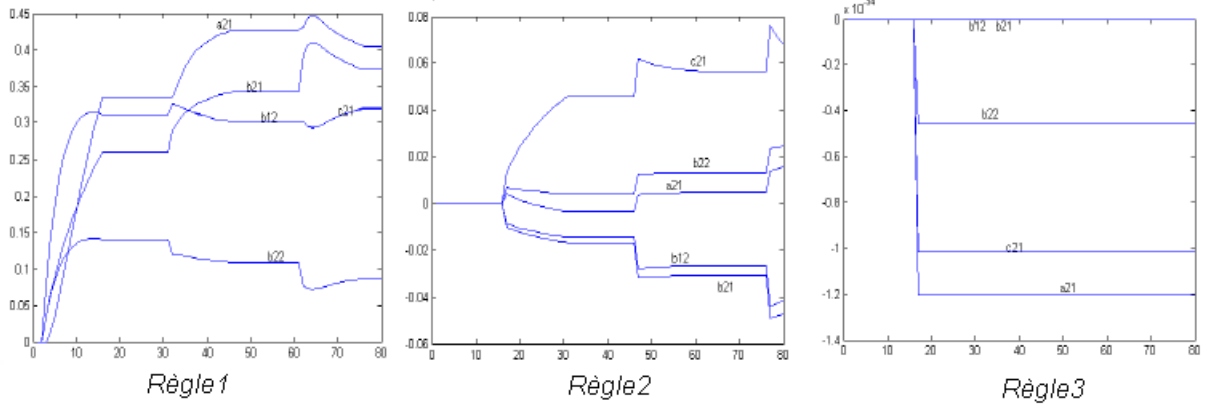


FIGURE 2.5 – Evolution des paramètres pour la sortie y_2

2.3 Méthodes de validation d'une structure optimale

Les algorithmes de classification partitionnent l'espace de données en un nombre de classes caractérisant les zones de fonctionnements du système. La qualité du modèle obtenu dépend du nombre de zones représentatives du système. Un nombre élevé de classes génère un modèle correct mais souvent complexe de point de vue nombre de paramètres et donc de calcul. En

revanche, si on se contente d'un nombre très faible de classes, le modèle serait imprécis [kaymak et Babuska, 1995]. Pour le cas particulier des deux algorithmes *FCM* et *GK* souvent utilisés pour l'identification des modèles TS, le nombre de classes, paramètre d'initialisation, doit être déterminé et fixé à l'avance [Abonyi *et al.*, 2002][Sun *et al.*, 2004].

Le problème de détermination du nombre optimal de classe est devenu un sujet de recherche et plusieurs méthodes ont été proposées dans la littérature. D'une manière générale, on peut les classer en trois catégories :

- Méthodes d'optimisation basées sur la minimisation d'une fonction coût intégrée dans les algorithmes de classification.
- Méthodes d'évaluation de la qualité de partitionnement basées sur les performances des indices de classification.
- Méthodes utilisant les critères numériques pour estimer l'erreur de modélisation.

2.3.1 Algorithmes d'optimisation

La plupart des algorithmes de classification existants dans la littérature réalisent la classification soit par la méthode hiérarchique ou par partition.

- Les algorithmes hiérarchiques procèdent à une décomposition hiérarchique des données, en générant une séquence de partitions arborescentes. Selon le principe de fonctionnement, ces algorithmes peuvent être divisés à leur tour en deux catégories figure (Fig.2.6).
 1. Les algorithmes agglomératifs commencent par un grand nombre de classes individuelles et progressivement les fusionnent selon une mesure de distance. La classification peut s'arrêter quand toutes les données sont dans un seul groupe ou bien à n'importe quel point intermédiaire désiré. Ces méthodes suivent une stratégie de fusion de classes de type constructive [Kukolj et Levi, 2004][Tsekouras, 2005].
 2. Les algorithmes par division suivent la stratégie opposée. Ils commencent par un groupe qui contient toutes les données et progressivement le groupe est divisé en groupes plus petits jusqu'à atteindre la situation limite dans laquelle chaque donnée constitue un groupe, ou bien avant cette limite, selon l'objectif [Tafazoli *et al.*, 2006][Tsekouras *et al.*, 2005].

Cependant, ces techniques hiérarchiques souffrent du fait qu'une fois qu'une fusion ou qu'une division est faite, elle ne peut pas être défaite ou raffinée.

- Les algorithmes de classification par partition, ils diffèrent des techniques hiérarchiques dans le fait qu'ils admettent la relocation des données dans les classes (dans le processus itératif, les données peuvent se déplacer d'une classe à une autre). Cela permet qu'une partition initiale mal estimée, puisse être corrigée à une étape postérieure. La formulation

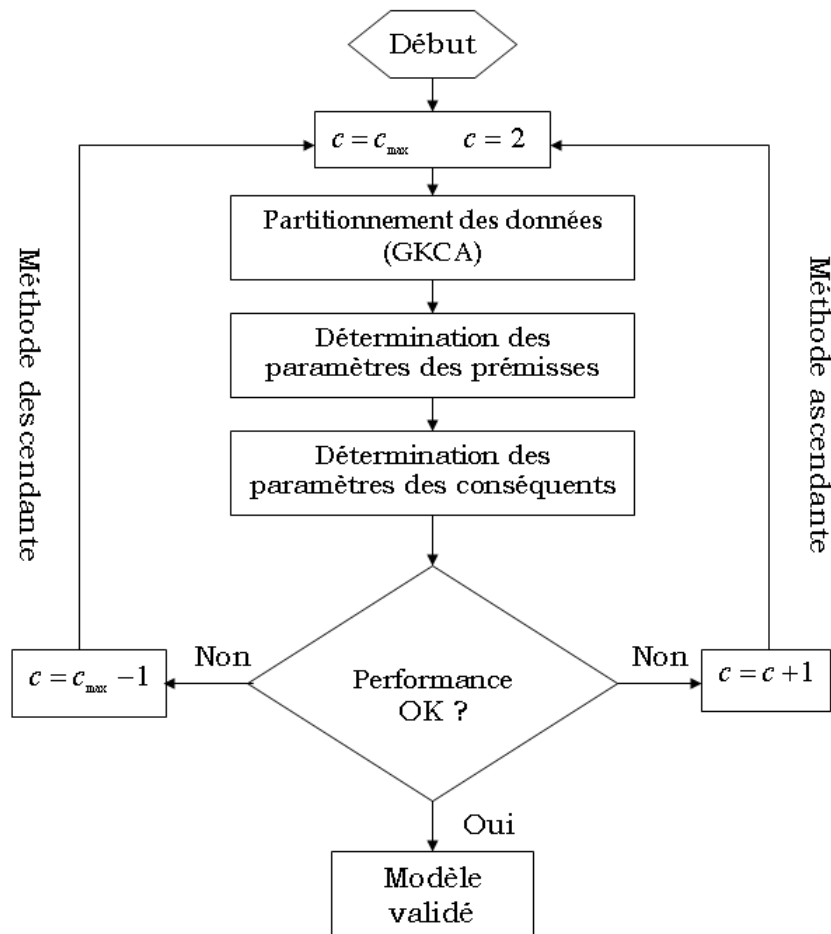


FIGURE 2.6 – Méthodes d'optimisation du nombre de classes

de ces algorithmes est basée sur l'utilisation d'une fonction objective (critère de classification) pour établir itérativement une partition floue appropriée d'un ensemble de données. Des algorithmes d'optimisation non linéaires sont utilisés pour chercher l'extremum local d'une telle fonction. Le principal inconvénient de l'approche par partition est la nécessité d'initialiser l'algorithme par le paramètre nombre de classe [Frigui et Krishnapuram, 1996].

2.3.2 Indices d'évaluation de la qualité d'une partition

Différents indices basés sur l'analyse de la dispersion et la distance entre les classes ont été proposés dans la littérature pour déterminer la qualité d'une partition. Quelques indices de validation standards sont présentés dans [Maulik et Bandyopadhyay, 2002][C.Franco *et al.*, 2002][Alexiuk et Pizzi, 2004]. La performance de la plupart des indices dépend du but de la classification (identification de modèles TS, traitement d'image, diagnostics ...etc.) ainsi que la

nature des algorithmes utilisés [Hadad *et al.*, 2006][Young *et al.*, 2004].

Une fonction de validité a pour but d'attribuer, à une partition donnée, un coefficient qui reflète la qualité de la classification obtenue à l'aide de l'algorithme utilisé. Dans le cas des algorithmes FCM et GK, par exemple, en évaluant ces coefficients pour différents choix de valeurs de c , on peut identifier les valeurs optimales de ce paramètre qui correspond à une partition reproduisant au mieux la structure des données traitées. Dans le cas général, une partition est d'autant meilleure que les éléments attribués à une classe donnée sont plus proches du centre de cette classe.

Parmi les coefficients qui donnent une idée sur la qualité de la classification, pour un point x_k donné à partir de ses c degrés d'appartenance μ_{ik} , on retrouve :

$$s_k = \sum_{i=1}^c (\mu_{ik})^2, \quad (2.19)$$

$$e_k = - \sum_{i=1}^c \mu_{ik} \cdot \ln \mu_{ik}, \quad (2.20)$$

Une mesure globale de la validité de la partition est le coefficient de partition qui n'est autre que la moyenne, sur tous les vecteurs x_k , des quantités s_k :

$$PC = \frac{\sum_{k=1}^n \sum_{i=1}^c (\mu_{ik})^2}{n} \quad (2.21)$$

Théoriquement, la classification est d'autant plus satisfaisante que le coefficient de partition est élevé, et donc, plus proche de 1. Cet indicateur mesure la séparation des classes et détermine le degré de chevauchement des partitions [Bezdek, 1981][Trauwaert, 1988].

Une autre mesure globale est l'entropie moyenne de la partition [Bezdek, 1974a][Bezdek, 1974b]. Elle est similaire au coefficient de partition précédent dans le sens que cette mesure est associée au degré du flou de la partition. Le critère est donné par l'expression suivante :

$$CE(c) = -\frac{1}{n} \sum_{i=1}^c \sum_{j=1}^n \mu_{ij}^2 \quad (2.22)$$

En général cet indicateur est significatif pour une valeur minimale.

Pour l'Index de partition (C_{IP}) [Bensaid *et al.*, 1996], il tient compte du rapport de la compacité et la séparation des classes, il est donné par :

$$C_{IP} = \sum_{i=1}^c \frac{\sum_{k=1}^N (\mu_{ik})^m \| \mathbf{x}_k - \mathbf{v}_i \|^2}{\sum_i \cdot \sum_{j=1}^c \| \mathbf{v}_j - \mathbf{v}_i \|^2} \quad (2.23)$$

Ce critère est utile aussi pour la comparaison de différentes partitions avec le même nombre

de classes. Une valeur faible du critère C_{IP} indique une meilleure partition.

La détermination de ces indices est réalisée par la *Fuzzy Clustering and Data Analysis Toolbox* [Abony, 2003]. Cette boîte à outils offre trois principales catégories de fonctions implémentées sous MATLAB : les différents algorithmes classification (FCMclust, GKclust, GGclust), la visualisation graphique des résultats de classification et le calcul des valeurs des différents critères de validation .

2.3.3 Validation numérique du modèle flou

Pour évaluer la qualité de l'approximation (performance numérique) obtenue par les modèles flous Takagi-Sugeno, nous utilisons le critère suivant :

La mesure VAF : Pourcentage de la variance

Introduit par [Babuska *et al.*, 1998], ce critère permet d'évaluer en pourcentage, la qualité d'un modèle en mesurant l'écart normalisé de la variance entre deux signaux. Sa valeur optimale est 100% quand les deux signaux sont égaux, plus ils sont différents, plus sa valeur devient faible. Le critère VAF est donné par l'expression (2.24) :

$$VAF = 100\% \left[1 - \frac{\text{var}(y - \hat{y})}{\text{var}(y)} \right] \quad (2.24)$$

La détermination de cet indice est réalisé par la **FMID** toolbox (*Fuzzy Modelling and Identification*). Cette boîte à outils, basée sur des fonctions MATLAB, permet l'identification des modèles Takagi-Sugeno et le modèle flou approximant les données moyennant l'algorithme GK. L'algorithme **ANFIS**(*système adaptatif d'inférence neuro-floue*) intégré à cette toolbox [Jang, 1993][Guillén *et al.*, 2007], permet d'améliorer l'erreur des modèles flous sur la base d'apprentissage et de prédiction.

2.4 Identification optimale des modèles Takagi-Sugeno

Dans cette section nous présentons une nouvelle méthode de détermination du nombre optimal de classes à partir des données entrée sortie d'un système non linéaire inconnu [Ouakka et Boumhidi, 2009]. Contrairement aux méthodes présentées dans les sections précédentes, basées sur le résultat d'un processus itératif pour minimiser certaines fonctions de coût ou pour valider les performances d'un certain nombres d'indices de classification, notre approche procède à la détermination directe du nombre optimal de classes par simple étude de la structure des données, sans nécessité d'utiliser les critères de validation conventionnels. En effet, il n'existe aucun critère universel qui puisse décider de ce qu'un algorithme donné soit adapté à un en-

semble de données quelconque, et c'est souvent sur la base de constatations empiriques qu'on se fait une idée sur la distribution réelle des données traitées.

En fait, les algorithmes de classification ont été développés à l'origine pour la classification des données, l'identification des modèles TS à partir de ces algorithmes n'est qu'une application des techniques de régression pour construire des modèles locaux linéaires pour chaque groupe de données d'une classe identifiée [Bortolet et Palm, 1997][Abonyi *et al.*, 2002][Kukolj et Levi, 2004]. L'algorithme **GK** est le plus utilisé pour l'identification des modèles TS. Mais, cet algorithme nécessite d'être initialisé par le nombre de classe défini à l'avance par un expert ou par l'exploitation des résultats des critères de validation réalisant la meilleur partition optimale parmi toutes celles obtenues.

2.4.1 Formulation de la méthode proposée

Les informations disponibles pour un système non linéaire à dynamique inconnues sont restreintes à un ensemble de données, généralement collectées par des capteurs à l'entrée et à la sortie du système. Dans ce sens, les approches de régression non linéaires et les formalismes des statistiques sont les méthodes classiques utilisées pour reconstruire un modèle approximatif représentatif du système. L'allure général des données, représentée par une fonction polynomiale permet la détection des extrema de la fonction. Ces extrema, généralement maximums et minimums, déterminent le nombre de classes (une classe est localisée entre un minimum et un maximum). Cette approche de partitionnement est orientée dès le départ, dans ce sens, pour minimiser la dispersion des données autour du modèle localement linéaire construit par une régression linéaire.

Un algorithme de réduction est aussi proposé, en vue de fusionner les classes dont les modèles linéaires associés présentent un certain degré de corrélation. Le principe générale de la méthode est présenté sur la figure (Fig.2.7).

2.4.2 Approximation des données par fonction polynomiale

Le problème d'identification des données par une fonction polynomiale peut être formulé de la façon suivante :

Considérons un système non linéaire, de type SISO, représenté par un ensemble de N données collectées à partir de mesures expérimentales :

$$(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$$

où $x_i \in X$ est la $i^{ème}$ entrée, et $y_i \in Y$ est la sortie correspondante.

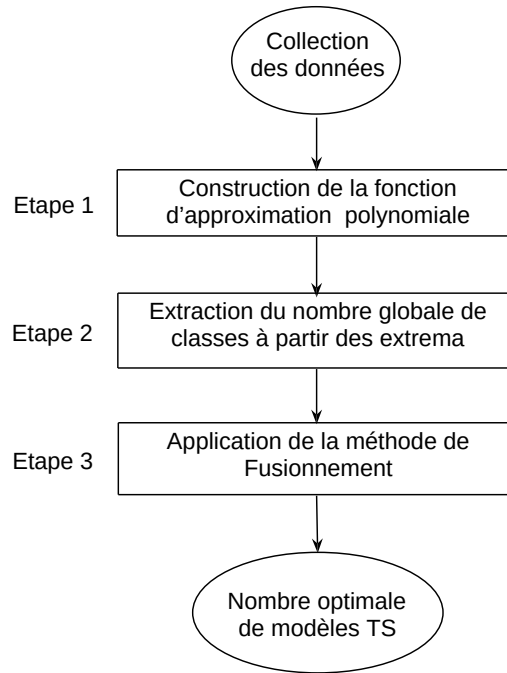


FIGURE 2.7 – Principe de la méthode d’optimisation du nombre de modèles TS

Dans le domaine de l’analyse numérique des données, on a souvent besoin d’établir un modèle mathématique liant plusieurs séries de données expérimentales. L’interpolation polynomiale consiste à approcher la courbe liant les deux séries de mesures par un polynôme. Les coefficients optimaux de ce polynôme sont ceux qui minimisent la variance de l’erreur d’interpolation. La fonction MATLAB **polyfit** retrouve le polynôme P de degré n permettant d’approcher la structure des données au sens des moindres carrés.

La fonction polynomiale est donnée par :

$$P = f(X) = \text{polyfit}(X, Y, n) \quad (2.25)$$

$$P(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n \quad (2.26)$$

où n est le degré du polynôme.

Dans un problème à n points d’interpolations, le polynôme d’interpolation doit coïncider avec la fonction en chacun des n points et peut être de degré aussi élevé que $n - 1$. Lorsque n augmente, on pourrait s’attendre à ce que la fonction et le polynôme d’interpolation deviennent de plus en plus proches. Cependant le phénomène de Runge peut se produire [Fornberg et Zuev, 2007]. Afin d’éviter ce problème, seule l’allure générale des données est recherchée, une

formule heuristique est proposée pour déterminer la valeur optimale du degré n [Ouakka et Boumhidi, 2009].

Soit r le coefficient de corrélation globale des données :

$$r = \text{corrélacion}(X, Y) \quad (2.27)$$

le degré optimale est donnée par la formule :

$$n = \sqrt{\frac{N}{1+r^2}} \quad (2.28)$$

La validité de cette approche est testée sur les deux exemples ci dessous :

$$y = \text{sinc}(x) + \varepsilon \quad (2.29)$$

$$y = 0.6 \sin(\pi x) + 0.3 \sin(3\pi x) + 0.1 \sin(5\pi x) + \delta \quad (2.30)$$

où $\varepsilon = 0.3\text{rand}(1, N)$ et $\delta = 0.1\text{rand}(1, N)$ sont des perturbations.

La valeur évaluée par la formule (2.28) pour déterminer les degrés optimaux des fonctions polynomiales représentant les systèmes (2.29) et (2.30) sont respectivement $n = 12$ et $n = 11$.

Comme illustré sur les figures (Fig.2.8) et (Fig.2.9), les fonctions polynomiales construites à partir de ces degrés optimaux, représentent au mieux le comportement des données traitées.

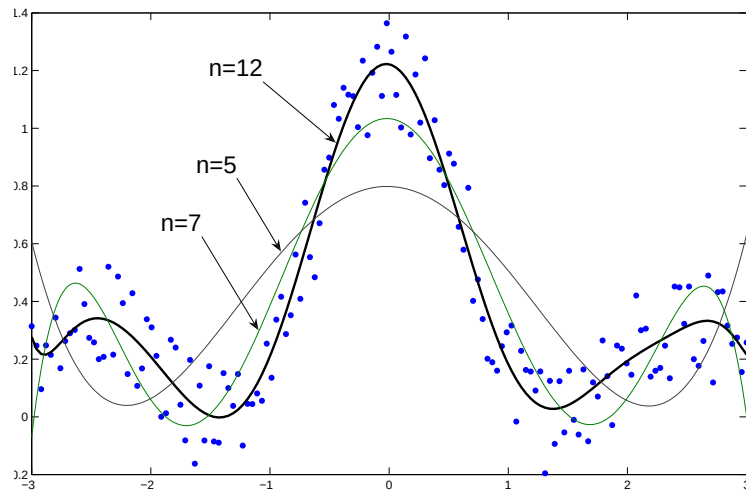


FIGURE 2.8 – Fonction d'approximation polynomiale pour le système (2.29)

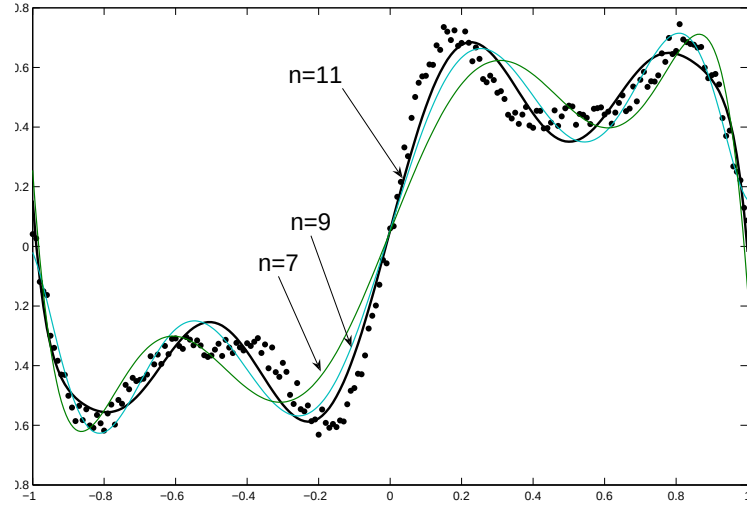


FIGURE 2.9 – Fonction d’approximation polynomiale pour le système (2.30)

2.4.3 Extraction du nombre globale de classes

En analysant la courbe du polynôme représentant le système (2.29), on remarque que la structure des données peut être partitionnée en 7 groupes délimitées chacun par les points extrema de la fonction d’approximation. Quant au nombre de minimum et de maximum détectés, il est égale à 6. De ce fait on constate que le nombre de groupes est égale au nombre des extrema augmenté d’une unité.

On peut alors proposer le résultat suivant : *la détermination du nombre de classes d’une structure de données, d’un système non linéaire, est équivalent à la détermination du nombre de points extrema de la fonction polynomiale optimale approximant l’allure des données du système correspondant.*

En se basant sur les deux théorèmes fondamentaux de l’algèbre :

Théorème 2.1 *Tout polynôme $P(x) = a_0 + \sum_{k=1}^n a_k x^k$ de degré n , à coefficients réels, admet exactement n racines dans l’ensemble $\mathbb{C}$.*

Théorème 2.2 *Une fonction polynomiale, de degré n , peut avoir au plus $(n - 1)$ points extrema.*

On conclut que le nombre de points extrema est égale au nombre de racines du polynôme dérivée $P'(x)$. Les racines du polynôme $P'(x)$ sont déterminées par la fonction MATLAB **polyder**. Le polynôme $P'(x)$ de degré $n - 1$ admet donc $(n - 1)$ racines réelles ou complexes.

Supposons que α est le nombre de racines réelles et β le nombre de racine complexes, en se référant aux résultats cités ci dessus, le degré du polynôme et les racines sont liées par la

relation :

$$\alpha + \beta = n - 1$$

Du fait que les données de mesures sont réelles $(X \times Y) \in \mathbb{R}^2$, et vu que l'ensemble des données d'entrées sont contenues dans l'ensemble X , on suppose que α^* est le nombre de ces racines contenues dans X .

λ_i définissent les coordonnées des abscisses des points extrema de la fonction polynomiale tel que :

$$\lambda = [\lambda_1, \lambda_2, \dots, \lambda_{\alpha^*}] \in X$$

L'ensemble des données d'entrées peut être partitionnée en α^* intervalles :

$$X = \underbrace{[x_1, \dots, \lambda_1]}_{X_1} \cup \underbrace{[\lambda_1, \dots, \lambda_2]}_{X_2} \cup \dots \cup \underbrace{[\lambda_{\alpha^*}, \dots, x_N]}_{X_{c^*}}$$

et l'espace des données est partitionné en c^* groupes situé chacun dans la zone délimitée par les points extremas :

$$X \times Y = \bigcup_{i=1}^{c^*} \overbrace{[X_i, Y_i]}^{classe} \quad (2.31)$$

où X_i est le i ème sous intervalle de l'ensemble des données d'entrée et Y_i son correspondant de sortie.

Le nombre globale de classes est ainsi donné par :

$$c^* = \alpha^* + 1 \quad (2.32)$$

En appliquant la régression par les moindres carrés, chaque groupe de données est approximé par un modèle linéaire de la forme :

$$y_i = a_i x + b_i, \quad i = 1, \dots, c^* \quad (2.33)$$

La figure (Fig.2.10) donne la structure du partitionnement des données, du système (2.29). Les sept sous ensembles ont été identifiés à partir de la positions des racines λ_i , $i = 1, \dots, 6$.

2.4.4 Détermination du nombre optimal de classes

L'objectif de cette partie est de présenter une méthode de réduction du nombre de classes identifiées. En fait, dans la modélisation et commande des systèmes non linéaires, on a tendance à réduire au maximum les paramètres du système afin de pouvoir mettre en oeuvre des lois de

commande simple et réduire le temps de calcul.

Le principe de fusionnement élaboré, consiste à fusionner deux classes successives dont les vecteurs associés, ayant comme supports les modèles locaux linéaires, sont colinéaires ou présentent une assez forte corrélation.

Soit $\vec{V}_i$ et $\vec{V}_{i+1}$ deux vecteurs centrés associés aux deux modèles linéaires y_i et y_{i+1} . L'angle formé par ces deux vecteurs est donné par (Fig.2.10) :

$$\theta = \text{angle}(\vec{V}_i, \vec{V}_{i+1}), \quad i = 1, 2, \dots, (c^* - 1) \quad (2.34)$$

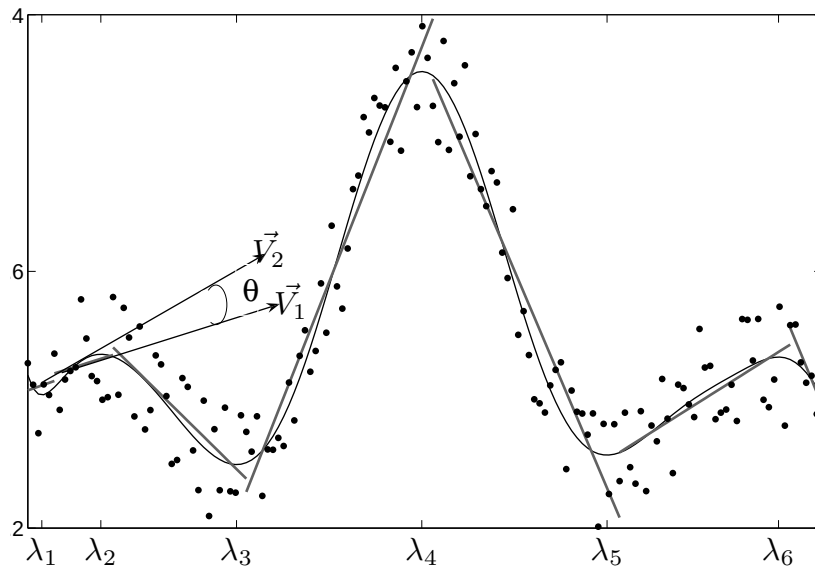


FIGURE 2.10 – Illustration du résultat de partitionnement des données du système (2.29) en 7 classes.

En géométrie vectorielle, le cosinus de l'angle θ et le coefficient de corrélation de deux vecteurs sont liés par la formule suivante :

$$\cos(\theta) = r \quad (2.35)$$

Pour un coefficient de corrélation $r = 0.82$, correspondant à la valeur de l'angle $\theta_r = 30^\circ$, la colinéarité des deux segments i et $i + 1$ est considérée assez forte. Par conséquent, le critère de validation de fusionnement est pris égale à θ_r .

La procédure de réduction du nombre de classes est décrite par l'algorithme suivant :

1. Etape 1 : Définition des paramètres d'initialisation et de contrôle

- k , Compteur du nombre de fusionnement, initialisé à $k = 0$;

- $\theta_r = 30^\circ$, Critère de validation de fusionnement ;
 - $c = \alpha^* + 1$, Nombre de classes identifiées par la méthode de partitionnement.
- Pour $i = 1, 2, \dots, (c - 1)$.

2. Etape 2 : Construction des vecteurs et détermination l'angle θ

- Vecteur $\vec{V}_i$ associé au modèle linéaire $\rightarrow y_i = a_i x + b_i, x \in X_i$;
- Vecteur $\vec{V}_{i+1}$ associé au modèle linéaire $\rightarrow y_{i+1} = a_{i+1} x + b_{i+1}, x \in X_{i+1}$;
- $\theta = \text{angle}(\vec{V}_i, \vec{V}_{i+1})$ formé par les deux vecteurs.

3. Etape 3 : Vérification du critère de validation de fusionnement

- Si $\theta \leq \theta_r$, pour une corrélation des deux vecteurs $r \geq 0.82$;
- Incrémenter le compteur du nombre de fusionnement, $k = k + 1$ et $\mathbf{c} = \mathbf{c} - \mathbf{k}$.

4. Etape 4 : Construction de la nouvelle classe et du modèle linéaire correspondant

- Fusionnement des deux intervalles X_i et X_{i+1} tel que $X'_i = X_i \cup X_{i+1}$;
- Identification du nouveau modèle linéaire pour la classe $(X'_i * Y'_i)$ tel que $y' = a'_i x + b'_i$.

Le nombre optimal de classes correspondant à une représentation optimale des données est donc donnée par :

$$\mathbf{C} = \mathbf{c}^* - \mathbf{k}. \quad (2.36)$$

Le nombre de classes identifié initialement pour le système (2.29) est de 7. En appliquant la méthode de fusionnement, ce nombre est réduit à 6. La figure (Fig.2.11) illustre le résultat du nouveau partitionnement. Les données des classes 1 et 2 ont été fusionnées pour construire une nouvelle classe, dont le sous intervalle correspondant est $[x_1, \dots, \lambda_2]$.

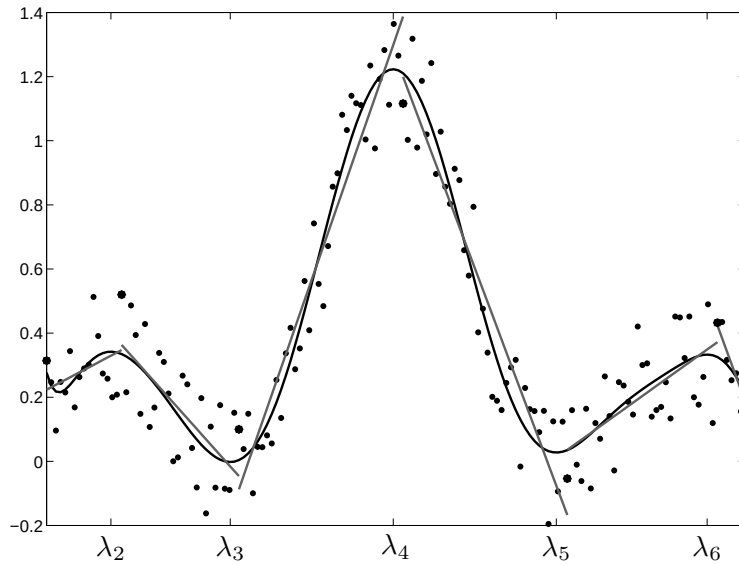


FIGURE 2.11 – Résultat de la nouvelle partition des données du système (2.29)

2.5 Simulations et résultats

Afin d'illustrer la performance de la méthode présentée, pour estimer le nombre optimal de classes, i.e. modèles Takagi-Sugeno, nous avons travaillé sur le système donné par [Riid *et al.*, 2001][Riid et Rüstern, 2004].

Soit donc la fonction non linéaire :

$$y(x) = 0.6 \sin(\pi x) + 0.3 \sin(3\pi x) + 0.1 \sin(5\pi x) + \delta \quad (2.37)$$

δ est une perturbation.

Celle ci est composée par un ensemble de 201 points équidistants dans l'intervalle $[-1, 1]$.

L'algorithme de GK a été adopté pour réaliser l'opération de classification et de construction du modèle flou.

La validation de la qualité de la classification sera réalisée par deux indicateurs : le coefficient de partition (PC) et le coefficient de classification entropie (CE).

Pour le système étudié, le degré optimal de la fonction polynomiale, évalué par la formule (2.28), est $n = 11$. Le nombre optimal de classes obtenu est égale à $c = 7$ (2.36).

Les figures (Fig.2.12) et (Fig.2.13) présentent le résultat des valeurs des indicateurs PC, PE, VAF(FMID) et VAF(ANFIS) pour un nombre de classes variant entre $c = 2$ et $c = 12$.

Tous ces indicateurs prouvent une bonne performance pour la valeur $c = 7$.

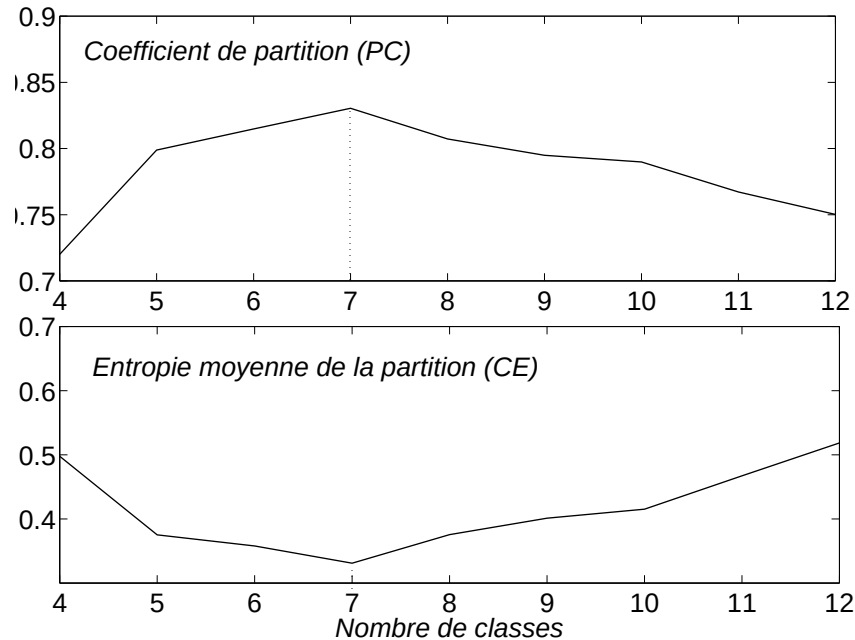
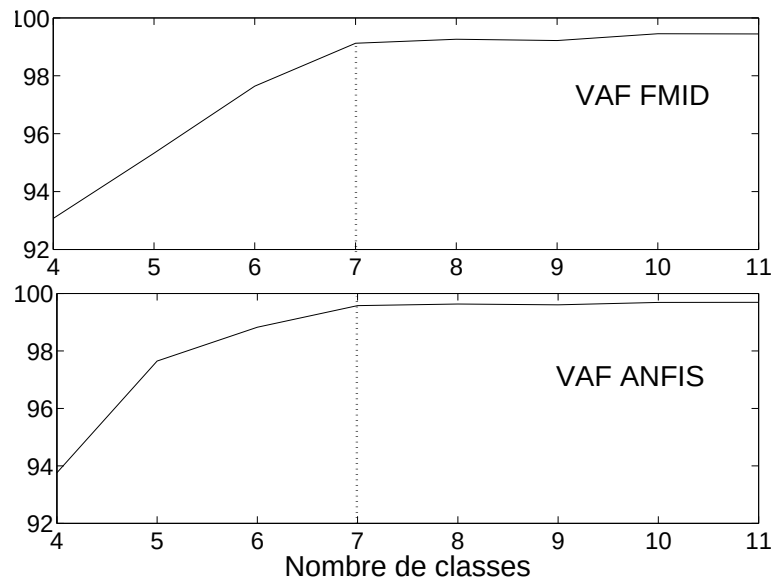
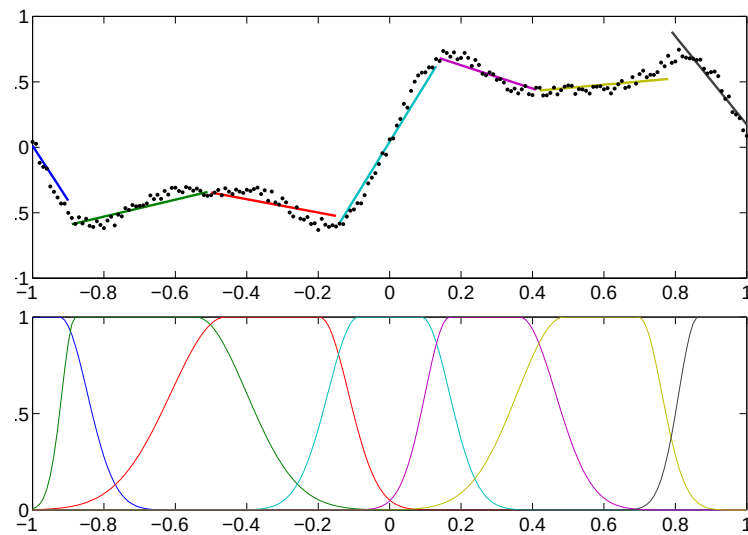


FIGURE 2.12 – Performances et résultats des deux indices de classification pour différentes valeurs de c .

FIGURE 2.13 – Performance et résultat de l'indices *VAF* pour différentes valeurs de c .FIGURE 2.14 – Modèles linéaires locaux et fonctions d'appartenance gaussiennes générées par l'algorithme GK pour $c = 7$.

La figure (Fig.2.14) montre l'allure des modèles linéaires locaux (modèles de Takagi-Sugeno) approximant les données de chaque classe, ainsi que la distribution de fonctions d'appartenances de type gaussiennes générées par l'algorithme GK.

Par l'approche proposée, on retrouve le même nombre optimal de classe, sans passer par les méthodes itératives classiques ou par l'analyse d'un expert humain.

2.6 Conclusion

Dans ce chapitre nous avons proposé deux stratégies de classification des données en se basant sur les modèles flous de TS, moyennant un ensemble de règles Si-Alors. Ces deux stratégies tiennent compte de tous les effets perturbateurs et représentent la clé de la commande des systèmes non linéaires. La première approche est destinée à modéliser les systèmes multivariables et laisse ainsi prétendre une large application aux systèmes réels sans aucune hypothèse restrictive sur l'interaction entre les différentes boucles.

La deuxième consiste à déterminer automatiquement le nombre optimal de classes afin d'identifier une structure correcte du système non linéaire à base de modèles TS.

Les deux approches ont été testées sur des exemples de simulation et ont prouvé une grande efficacité de point de vue analyse et interprétation sans avoir recours au jugement d'un expert humain et aux techniques de validation classiques souvent très lourdes en temps et en charge de calcul.

Commande Floue d'une Classe de Systèmes Non Linéaires

3.1 Introduction

Ce chapitre présente quatre stratégies de commande floue pour une classe de systèmes non linéaires à dynamiques inconnues. La première est destinée à commander les systèmes mono entrée mono sortie (SISO) ; elle est basée sur la combinaison de l'approche de linéarisation entrée sortie aux mécanismes de la logique floue [Feng et Chen, 2005]. La deuxième est l'extension du cas monovariable à celui multivariable. Dans ce cas, le système multivariable est décomposé en sous systèmes multi entrées mono sortie (MISO), ce qui permet d'éviter la charge lourde des calculs de l'approche globale et permet aussi l'application des algorithmes d'adaptation des paramètres sans aucune condition restrictive sur la richesse de l'excitation, donc sur la commande. Notons aussi que, par cette décomposition, le problème des non linéarités, du couplage croisé et des perturbations externes est facilement résolu dans le cadre de la modélisation par la logique floue. La synthèse de la loi de commande pour la première approche est indirecte, elle est basée sur la linéarisation entrée sortie. La modélisation floue type Takagi-Sugeno est adoptée et permet de transformer le système non linéaire à un ensemble de modèles localement linéaires. L'algorithme des moindres carrés récursifs avec normalisation et projection est ensuite utilisé afin d'estimer les paramètres des modèles linéaires.

La synthèse de la loi de commande dans le cas MISO passera d'abord par une classification des données afin d'extraire les fonctions d'appartenance ainsi que les conclusions de chaque règle et par conséquent le modèle de Takagi-Sugeno correspondant. Le même algorithme d'estimation paramétrique est utilisé pour chaque sous système et permet donc la détermination des paramètres des modèles locaux qui sont utilisés par la suite dans la conception de la loi de

commande.

Devant la complexité croissante des grands systèmes, les outils classiques développés pour l'analyse et la commande de ce type de systèmes demeurent limités vis à vis des hypothèses restrictives qui ne sont plus vérifiées surtout quand l'information sur le système en question est indisponible. L'idée de base, consiste à exploiter l'expérience des opérateurs humains pour construire une loi de commande décentralisée. Un jeu de règles floues traduit alors le comportement des opérateurs en termes de stratégie de commande. Ainsi, deux lois de commande décentralisées directe et indirecte sont proposées et font l'objet de la troisième approche.

La quatrième approche consiste à combiner le régulateur PI, la commande mode de glissement et la technique de logique floue. Afin d'éviter le phénomène de réticence et d'assurer la convergence de l'erreur, autrement dit pour éviter de se retrouver au voisinage de zéro lorsque la technique mode de glissement est appliquée, la surface de glissement sera décrite par des ensembles flous. La description linguistique du contrôleur flou est caractérisée par deux règles et selon la prémisse de la règle, une loi de commande PI ou mode de glissement est activée. Les paramètres de ce contrôleur sont déterminés par l'approche de Lyapunov.

Les quatre lois de commandes proposées assurent une bonne poursuite et maintiennent la stabilité en présence de tous les effets perturbateurs.

3.2 Modélisation floue de Takagi-Sugeno

Soit la classe de systèmes non linéaires décrite par l'équation suivante :

$$y(t+1) = f(x(t)) \quad (3.1)$$

Avec $x(t)$ est le vecteur de régression donné par :

$$x(t) = [y(t), \dots, y(t-n), u(t), \dots, u(t-n)] \quad (3.2)$$

La dynamique inconnue est approximée par le modèle flou T-S dont la description linguistique de la base de règles est définie par :

$$\begin{aligned} R^i : \quad & \text{Si } x_1 \text{ est } F_1^i \text{ et } x_s \text{ est } F_s^i \text{ Alors} \\ & y_i(t+1) = a_{i1}y(t) + \dots + a_{in}y(t-n) + b_{i1}u(t) + \dots + b_{in}u(t-n) + c_i, \quad i = 1, \dots, m \end{aligned} \quad (3.3)$$

Avec $F_i^j, i = 1, \dots, n$ sont des variables floues caractérisées par des fonctions d'appartenance $\mu_{F_i^j}$ et $\theta_i = [a_{i1}, \dots, a_{in}, b_{i1}, \dots, b_{in}, c_i]$ est le vecteur paramètre correspondant au modèle flou. La méthode d'inférence choisie est le produit, et l'opération de défuzzification est obtenue par

la méthode du centre de gravité.

La sortie générée par le modèle flou est donnée par :

$$y(t+1) = \sum_{i=1}^m \mu_i [a_{i1}y(t) + \dots + a_{in}y(t-n) + b_{i1}u(t) + \dots + b_{in}u(t-n) + c_i] \quad (3.4)$$

3.3 Synthèse de la loi de commande pour un système SISO

3.3.1 Cas non adaptatif

La structure du contrôleur est basée sur l'approche de linéarisation entrées sorties. La même base de règles adoptée pour modéliser la sortie $y(t+1)$ est maintenue. Une fois les paramètres du modèle déterminés, ils sont directement utilisés par le contrôleur. Là aussi, la conclusion des règles est portée sur la commande selon l'équation :

$$R^i : \text{ Si } x_1 \text{ est } F_1^i \text{ et } x_s \text{ est } F_s^i \text{ Alors} \\ u_i(t) = \frac{1}{b_{i1}} [-h_i(t) + y_r(t+1) - k_1 e(t-1) - \dots - k_n e(t-n+1)], \quad i = 1, 2, \dots, m \quad (3.5)$$

où : $h_i(t) = a_{i1}y(t) + \dots + a_{in}y(t-n+1) + b_{i2}u(t-1) + \dots + b_{in}u(t-n+1) + c_i$.

La loi de commande globale obtenue par la défuzzification :

$$u(t) = \sum_{i=1}^m \mu_i u_i(t) = \sum_{i=1}^m \mu_i \frac{1}{b_i} [-h_i(t) + y_r(t+1) - k_1 e(t) - k_2 e(t-1) - \dots - k_n e(t-n+1)] \quad (3.6)$$

Pour assurer la stabilité, les coefficients k_i sont choisis de façon à ce que le polynôme Ω soit de Hurwitz :

$$\Omega(z) = z^n + k_1 z^{n-1} + \dots + k_{n-1} z + k_n$$

Substituons la loi de commande (3.6) dans (3.4), l'équation de l'erreur dynamique devient :

$$e(t+1) + k_1 e(t) + \dots + k_n e(t-n+1) = 0 \quad (3.7)$$

Ce qui démontre la convergence asymptotique de l'erreur de poursuite.

3.3.2 Cas adaptatif

Comme tout système réel évolue dans le temps, les paramètres de son modèle varient aussi en fonction des circonstances de l'expérience, ce qui impose une adaptation de ces paramètres. L'algorithme d'adaptation utilisé est celui des moindres carrés récursif avec normalisation et projection [Praly, 1983] décrit par :

$$\phi_i(t-1) = [\mu_{iy}(t-1), \dots, \mu_{iy}(t-n), \mu_{iu}(t-1), \dots, \mu_{iu}(t-n), 1]^T$$

$$\theta_i = [a_{i1} \dots a_{in}, b_{i1}, \dots, b_{in}, c_i]$$

$$\phi(t-1) = [\phi_1^T(t-1) \phi_2^T(t-1) \dots \phi_m^T(t-1)]^T$$

$$\theta = [\theta_1^T \theta_2^T \dots \theta_m^T]^T$$

L'équation (3.4) peut être réécrite sous la forme : $y(t) = \theta^T \phi(t-1)$

L'obtention du vecteur de paramètres estimés $\hat{\theta}$ à partir du vecteur de paramètres du système θ est donnée par l'algorithme des moindres carrés suivant :

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{p(t-1)\bar{\phi}(t-1)\bar{e}_1(t)}{\lambda(t) + \bar{\phi}^T(t-1)p(t-1)\bar{\phi}(t-1)} \quad (3.8)$$

Avec :

- $\bar{e}_1(t) = \bar{y}(t) - \bar{\phi}'(t-1)\theta(t-1)$
- $p(t) = p(t-1) - \frac{p(t-1)\bar{\phi}(t)\bar{\phi}'(t)p(t-1)}{\lambda(t) + \bar{\phi}^T(t-1)p(t-1)\bar{\phi}(t-1)}$
- $\lambda(t) = \lambda_0\lambda(t-1) + (1 - \lambda_0)$, avec $\lambda_0 = 0.95$, $\lambda(0) = 0.96$, $p(0) = \varepsilon I$, $0 < \varepsilon < \infty$.
- $\bar{y}(t) = \frac{y(t)}{\eta(t-1)}$ $\bar{\phi}(t) = \frac{\phi(t-1)}{\eta(t-1)}$, $\eta(t) = \max\{1, \|\phi(t)\|\}$, $\eta(t)$ est un signal de normalisation.

La description linguistique de base de règle du contrôleur est donnée par :

$$R^i : \text{ Si } x_1 \text{ est } F_1^i \text{ et } x_s \text{ est } F_s^i \text{ Alors } u_i(t) = \frac{1}{\hat{b}_i} [-\hat{h}_i(t) + y_r(t+1) - k_1 e(t-1) - \dots - k_n e(t-n+1)], \quad i = 1, 2, \dots, m \quad (3.9)$$

où : $\hat{h}_i(t) = \hat{a}_{i1}y(t) + \dots + \hat{a}_{in}y(t-n+1) + \hat{b}_{i2}u(t-1) + \dots + \hat{b}_{in}u(t-n+1) + \hat{c}_i$.

Par la méthode de défuzzification, la loi de commande globale proposée [Lagrat *et al.*, 2007a](Annexe C) est de la forme suivante :

$$u(t) = \sum_{i=1}^m \mu_i u_i(t) = \sum_{i=1}^m \mu_i \frac{1}{\hat{b}_i} [-\hat{h}_i(t) + y_r(t+1) - k_1 e(t) - k_2 e(t-1) - \dots - k_n e(t-n+1)] \quad (3.10)$$

En remplaçant la loi de commande (3.10) dans (3.4), l'équation de l'erreur de poursuite

devient :

$$e(t+1) + k_1 e(t) + \dots + k_n e(t-n+1) = \sum_{i=1}^m \mu_i [h_i(t) - \hat{h}_i(t)] + \sum_{i=1}^m \mu_i [b_{1i} - \hat{b}_{1i}] u(t) \quad (3.11)$$

En définissant $x_e(t) = [e(t-n+1) \ e(t-n+2) \ \dots \ e(t)]^T$, le système en boucle fermé peut être représenté dans l'espace d'état comme suit :

$$x_e(t+1) = Ax_e(t) + Be(t+1) \quad (3.12)$$

où

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & 0 & \ddots & 0 \\ 0 & 0 & \dots & 0 & 1 \\ -k_n & -k_{n-1} & \dots & -k_2 & -k_1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (3.13)$$

Théorème 3.1 : Pour le système (3.4), si la loi de commande adaptative est définie par (3.10), alors le système en boucle fermée est stable dans le sens que tous les signaux entrée sortie sont bornés et l'erreur de poursuite converge asymptotiquement vers zéro.

La démonstration du théorème se trouve sur l'annexe E.

3.4 Synthèse de la loi de commande pour un système MISO

3.4.1 Description du modèle flou

L'objectif principal est de développer une stratégie de commande de la classe des systèmes multivariables. Dans ce contexte, la technique de linéarisation entrée sortie est combinée avec l'approche floue. Afin d'éviter la charge des calculs du traitement global des systèmes MIMO, nous opterons pour la décomposition du système MIMO en un ensemble de sous systèmes MISO [Lagrat *et al.*, 2008]. Dans ce cas, chaque sous système sera traité indépendamment des autres, l'algorithme d'estimation paramétrique est appliqué à chaque sous système et nous permet de déduire à chaque fois le vecteur paramètres à la place de la matrice paramètre. Ainsi, la condition restrictive de la richesse de l'excitation riche dans le cas global est évitée.

L'interaction entre les différentes boucles, les non linéarités et les perturbations externes sont modélisés par le système flou de Takagi-Sugeno. Chaque sous système MISO est donc

transformé en un ensemble de modèles localement linéaires. L'agrégation des règles conduit par la suite à un seul modèle. L'algorithme d'estimation paramétrique des moindres carrés récursif avec normalisation et projection des données décrit dans la section précédente est utilisé pour adapter les paramètres.

L'identification hors ligne du modèle flou a permis la classification des données, à savoir, les antécédents et les caractéristiques de la fonction d'appartenance (le centre v_{ji} et la variance σ_{ji} pour le cas d'une gaussienne).

Le détail de la modélisation Takagi-Sugeno pour les systèmes MISO est décrit dans le chapitre 2.

Soit le système non linéaire MIMO représenté par :

$$y(k+1) = f(\phi(k)) \quad (3.14)$$

$$y(k+1) = f(y(k), \dots, y(k-n+1), u_1(k), u_1(k-n+1), \dots, u_2(k), \dots, u_2(k-n+1), \dots, u_{nu}(k)(k-n+1), \eta(k))$$

f représente la fonction non linéaire inconnue et η est un vecteur de perturbation.

On définit le vecteur de régression $\phi(k)$ par :

$$\phi(k) = [y(k), \dots, y(k-n+1), u_1(k), u_1(k-n+1), \dots, u_2(k), \dots, u_2(k-n+1), \dots, u_{nu}(k)(k-n+1), \eta(k)]$$

Le comportement non linéaire est transformé donc en un ensemble de modèles linéaires par une composition de règles Si Alors [Lagrat *et al.*, 2007b](Annexe B), le modèle flou est basé sur la collection de règles R_j du type :

$$R_j: \text{ Si } y(k) \text{ est } \Omega_{j1} \text{ et } \dots \text{ et } u_{nu}(k-n+1) \text{ est } \Omega_{jnu} \text{ Alors} \\ y_j(k+1) = \sum_{r=1}^n a_{jr} y_i(k-r+1) + \sum_{r=1}^n b_{jr} u(k-r+1) + \sum_{l=1}^{nu} \sum_{r=1}^n b_{jlr} u_l(k-r+1) + c_j \\ j = 1, 2, \dots, M$$

où R_j dénote la j ème règle du modèle et M est le nombre de règles que contient la base de règles. Ω_{ij} est le sous ensemble flou de l'antécédent $\phi_i(k)$, définie par une fonction d'appartenance gaussienne de centre v_j et de variance σ_j .

On utilise l'implication floue type produit pour calculer le degré d'activation d'une proposition floue :

$$\mu_j = \exp\left(\frac{-1}{2} \frac{(y(k) - v_{j1})^2}{\sigma_{j1}^2}\right) \cdots \exp\left(\frac{-1}{2} \frac{(u_{nu}(k-n+1) - v_{jnu})^2}{\sigma_{jnu}^2}\right) \quad (3.15)$$

Le degré d'activation normalisé est défini par :

$$\Phi_j(\phi(k), v_j, \sigma_j) = \frac{\mu_j(\phi(k))}{\sum_{i=1}^M \mu_i(\phi(k))} \quad (3.16)$$

Le modèle global est obtenu par l'agrégation des modèles locaux linéaires :

$$y(k+1) = \sum_{j=1}^M \Phi_j(\phi(k), v_j, \sigma_j) y_j(k+1) \quad (3.17)$$

Comme chaque identification hors ligne est suivi d'une autre en ligne pour permettre l'adaptation des paramètres, la méthode de classification des données telle que décrite au chapitre 2 est appliquée et va de soi pour l'algorithme d'identification paramétrique.

3.4.2 Synthèse de la loi de commande

La synthèse du contrôleur flou proposé est basée sur l'architecture du modèle flou obtenu. Le contrôleur a la forme suivante [Lagrat *et al.*, 2008](Annexe E) :

$$\begin{aligned} R^j : \quad & \text{Si } y(k) \text{ est } \Omega_{j1} \text{ et } u_{nu}(k-n+1) \text{ est } \Omega_{jnu} \text{ Alors} \\ & u_j(k) = \frac{1}{\hat{b}_j} [-\sum_{r=1}^n \hat{a}_{jr} y_i(k-r+1) - \sum_{r=1}^n \hat{b}_{jr} u(k-r+1) - \sum_{l=1}^{nu} \sum_{r=1}^n \hat{b}_{jlr} u_l(k-r+1) \\ & - \hat{c}_j + y_m(k+1) - \lambda_1 e(k) - \dots - \lambda_n e(k-n+1)] \quad j = 1, 2, \dots, M \end{aligned}$$

où $e = y - y_m$ est l'erreur de poursuite.

Pour résoudre le problème d'instabilité, une stratégie de placement de pôles est utilisée. Les coefficients $\{\lambda_i\}$ sont choisis de façon à ce que le polynôme (3.18) soit Hurwitzien.

$$\lambda(z) = z^n + \lambda_1 z^{n-1} + \dots + \lambda_{n-1} z + \lambda_n = 0 \quad (3.18)$$

L'ensemble des facteurs perturbateurs y compris les interactions entre les variables d'entrées sont modélisés par le système flou proposé et sont donc tenus en compte dans la synthèse du contrôleur :

$$u(k) = \sum_{j=1}^M \Phi_j(\phi(k), v_j, \sigma_j) u_j(k) \quad (3.19)$$

Théorème 3.2 : *Considérons le modèle dynamique flou du système (3.17). Si la loi de commande proposée (3.19) et l'algorithme d'adaptation sont appliqués, alors le système en boucle fermée est stable dans le sens que tous les signaux entrées sorties du système sont bornés et l'erreur de poursuite converge asymptotiquement vers zéro.*

Preuve :

En combinant (3.19) et (3.17) , l'erreur en boucle fermée est donnée par :

$$e(k+1) + \lambda_1 e(k) + \dots + \lambda_{n-1} e(k-n+2) + \lambda_n e(k-n+1) = H(k) - \hat{H}(k) + \sum_{j=1}^M \Phi_j(\phi(k), v_j, \sigma_j) (b_{j1} - \hat{b}_{j1}) u_j(k) \quad (3.20)$$

avec

$$H_j(k) = \sum_{r=1}^n a_{jr} y_i(k-r+1) - \sum_{r=1}^n b_{jr} u(k-r+1) - \sum_{l=1}^{nu} \sum_{r=1}^n b_{jlr} u_l(k-r+1) + c_j \quad (3.21)$$

$$\hat{H}_j(k) = \sum_{r=1}^n \hat{a}_{jr} y_i(k-r+1) + \sum_{r=1}^n \hat{b}_{jr} u(k-r+1) + \sum_{l=1}^{nu} \sum_{r=1}^n \hat{b}_{jlr} u_l(k-r+1) + \hat{c}_j \quad (3.22)$$

On définit :

$$x_e(k) = [e(k-n+1) \quad e(k-n+2) \dots e(k)]^T \quad (3.23)$$

L'équation d'erreur dans l'espace d'état devient :

$$x_e(k+1) = A x_e(k) + B \left\{ H(k) - \hat{H}(k) + \sum_{j=1}^M \Phi_j(\phi(k), v_j, \sigma_j) (b_{j1} - \hat{b}_{j1}) u_j(k) \right\} \quad (3.24)$$

avec

$$\begin{aligned} y(k+1) &= H(k) + \sum_{j=1}^M \Phi_j(\phi(k), v_j, \sigma_j) b_{j1} u_j(k) \\ \hat{y}(k+1) &= \hat{H}(k) + \sum_{j=1}^M \Phi_j(\phi(k), v_j, \sigma_j) \hat{b}_{j1} u_j(k) \end{aligned} \quad (3.25)$$

si

$$e(k+1) = y(k+1) - \hat{y}(k+1) \quad (3.26)$$

L'équation (3.24) devient :

$$x_e(k+1) = A x_e(k) + B e(k+1) \quad (3.27)$$

Les matrices A et B définies comme suit :

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & 0 & \ddots & 0 \\ 0 & 0 & \dots & 0 & 1 \\ -\lambda_n & -\lambda_{n-1} & \dots & -\lambda_2 & -\lambda_1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (3.28)$$

on remarque que les valeurs propres de la matrice se trouvent à l'intérieur du cercle unité ce qui permet de garantir la stabilité.

3.5 Synthèse d'une loi de commande décentralisée

3.5.1 Problématique

L'application de la commande centralisée pour les grands systèmes interconnectés, exige deux conditions fondamentales : un contrôleur sophistiqué généralement très coûteux et complexe, difficile à mettre en oeuvre à cause du manque de centralisation de l'information. Ce défi à la théorie du contrôle centralisé, pousse les chercheurs à envisager le problème d'une autre manière, surtout lorsque le grand système est décomposable d'une façon physique et naturelle. Une telle décomposition rajoute le terme décentralisé à la littérature du contrôle centralisé des grands systèmes.

Dans la suite, on va présenter deux approches de commandes floues décentralisées, le cas direct et indirect. Ces deux approches sont proposées pour une classe à grande échelle des systèmes non linéaires interconnectés. Afin d'assurer une poursuite asymptotique de la trajectoire de référence, les réactions et les mécanismes d'adaptation pour chaque sous-système dépendent uniquement des mesures locales fournies. Pour les deux approches, le but des commandes proposées est d'approximer les sous-systèmes inconnus. En outre, chaque sous système est en mesure de compenser les termes d'interconnexions et des perturbations.

On considère le système non linéaire décrit par un ensemble de sous systèmes SISO interconnectés (S_i) :

$$\begin{cases} \dot{x}_{i,1} = x_{i,2} \\ \vdots \\ \dot{x}_{i,n_i} = f_i(x_i) + g_i(x_i)u_i + \Delta_i(\underline{x}) \\ y_i = x_{i,1} \end{cases} \quad (3.29)$$

où $\underline{x} = [x_1^T, x_2^T, \dots, x_N^T]$, $x_i \in \mathbb{R}^{n_i}$, est le vecteur d'état global, $u_i(t) \in \mathbb{R}$ est l'entrée, y_i est la sortie du sous-système S_i . Les fonctions $f_i(\cdot)$, $g_i(\cdot)$ ($i = 1, 2, \dots, N$) sont inconnues.

Les interconnexions entre le sous système i et les sous systèmes j ($j = 1, 2, \dots, N; j \neq i$) sont représentées par $\Delta_i(\underline{x}) \in \mathbb{R}$

Hypothèse :

On suppose que le terme des interconnexions $\Delta_i(\underline{x})$ vérifie l'équation suivante :

$$|\Delta_i(\underline{x})| \leq \sum_{j=1}^N q_{ij} \|e_j\|_2 \quad (3.30)$$

Le scalaire q_{ij} quantifie l'effet du sous système j sur le sous système i , et $\|\cdot\|_2$ la norme Euclidienne.

$e_i = y_{ir} - y_r$ est l'erreur de poursuite pour le système S_i .

le vecteur erreur pour le ième sous système est donnée par $e_i = [e_{i0}, \dot{e}_{i0}, \dots, e_{i0}^{(d_i-1)}]^T$.

Si $f_i(\cdot)$ et $g_i(\cdot)$ sont connues ($g_i(x_i) \neq 0$, et $\Delta_i(\underline{x}) = 0$), alors la loi de commande la plus appropriée pour le ième sous ensemble a l'expression suivante :

$$u_i^* = \frac{1}{g_i(x_i)} \left(-f_i(x_i) + K_i^T e_i + y^{(d_i)} \right) \quad (3.31)$$

dont l'objectif est de garantir

$$\lim_{t \rightarrow +\infty} e_{i0} = 0$$

En pratique, cet objectif ne peut pas être atteint.

3.5.2 Contrôleur adaptatif flou direct

La stratégie de commande proposée [Errahmani *et al.*, 2007](Annexe D) est basée sur l'élaboration d'un contrôleur composé de trois termes :

- un terme de poursuite $u_i(\underline{x}_i, \underline{\theta}_i)$ déterminé à partir du mécanisme de la logique floue. Ce terme est approximé par un modèle flou matérialisé par un ensemble de règles Si-Alors.
- le terme $a_i(t)e_i^T p_i b_i$ synthétisé de façon à compenser les termes d'interconnexions inconnus (p_i est une matrice définie positive déterminée par l'approche de lyapunov).
- u_{ih} est un signal auxiliaire incorporé dans la loi de commande pour compenser les erreurs de modélisation.

La loi de commande est donnée par :

$$u_i = u_i(\underline{x}_i, \underline{\theta}_i) + g_i^{-1} \left(a_i(t)e_i^T p_i b_i + u_{ih} \right) \quad (3.32)$$

3.5.3 Contrôleur adaptatif flou indirect

Cette approche se fait en deux étapes, la première consiste d'abord à approximer par un modèle flou les dynamiques $f_i(x_i)$ et $g_i(x_i)$, ce qui permet d'obtenir des modèles linéaires. La deuxième étape consiste à synthétiser les lois de commande en se basant sur les estimés de $f_i(x_i)$ et $g_i(x_i)$. Ainsi la loi de commande proposée est donnée par :

$$u_i = u_{if} + \frac{1}{\hat{g}_i(\underline{x}_i/\theta_{i2})} \left(-\hat{f}_i(\underline{x}_i/\theta_{i1}) + K_i^T e_i + y_{ir}^{d_i} + a_i(t)e_i^T p_i b_i/2 + u_{ih} \right) \quad (3.33)$$

où u_{if} est le terme de commande du contrôleur flou.

Le détail des calculs et la démonstration de la stabilité avec la technique de la logique floue utilisée sont donnés dans l'annexe D.

3.6 Régulateur PI combiné à l'approche mode glissant

3.6.1 Formulation du problème

Considérons la classe de systèmes non linéaires étudiée décrite par l'équation suivante :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \vdots \\ \dot{x}_n = f(\underline{x}, t) + g(\underline{x}, t)u + d(t) \end{cases} \quad (3.34)$$

où $f(\underline{x}, t)$ et $g(\underline{x}, t)$ sont deux fonctions continues et inconnues. $u \in R$ et $y \in R$ sont respectivement l'entrée et la sortie du système. Les perturbations externes sont représentées par $d(t)$.

Nous pouvons résoudre le problème de suivi de trajectoire par la loi de commande [Lagrat *et al.*, 2006](Annexe A) :

$$u = \frac{1}{\hat{g}(\underline{x}, \underline{\theta}_g)} \left[-\hat{f}(\underline{x}, \underline{\theta}_f) - \sum_{i=1}^{n-1} c_i e^i + y_r^{(n)} + u_s \right] \quad (3.35)$$

$\hat{f}(\underline{x}, \underline{\theta}_f)$ et $\hat{g}(\underline{x}, \underline{\theta}_g)$ sont respectivement les fonctions estimées correspondantes de $f(\underline{x}, t)$ et $g(\underline{x}, t)$.

Le terme u_s désigne la commande additive, elle est synthétisée par la technique de mode glissant :

$$u_s = -k_d \text{sign}(\sigma) \quad (3.36)$$

k_d est une constante positive qui représente le gain de la commande.

La fonction $\text{sign}(\sigma)$ est définie comme suit :

$$\text{sign}(\sigma) = \begin{cases} 1 & \text{si } \sigma > 0 \\ 0 & \text{si } \sigma = 0 \\ -1 & \text{si } \sigma < 0 \end{cases} \quad (3.37)$$

avec σ la surface de glissement.

L'avantage que procure cette commande est la robustesse vis-à-vis des perturbations et des incertitudes paramétriques. Le phénomène de réticence est provoqué par la partie discontinue de

cette commande. Ceci peut dégrader les performances et déstabiliser le système [Utkin, 1992]. Parmi les solutions proposées à ce problème on peut citer la commande par mode glissant à bande limite [Slotine et Li, 1991] qui consiste à effectuer une approximation continue des discontinuités présentes dans la loi de commande (3.36) au voisinage de la surface de glissement. Cette méthode permet de réduire l'effet de réticence est de remplacer la fonction discontinue par une fonction de saturation. D'autre part, ce problème peut être facilement résolu par la stratégie de commande floue basée sur la combinaison du régulateur PI à l'approche de mode de glissement.

3.6.2 Structure de la loi de commande

La synthèse de la commande consiste à traiter la surface de glissement par un raisonnement flou dont le mécanisme est le suivant :

- *Fuzzification*

La surface de glissement σ est fuzzifiée, elle est décrite par des ensembles flous SM et LR caractérisant les fonctions d'appartenances comme décrit sur la figure (Fig.3.1) :

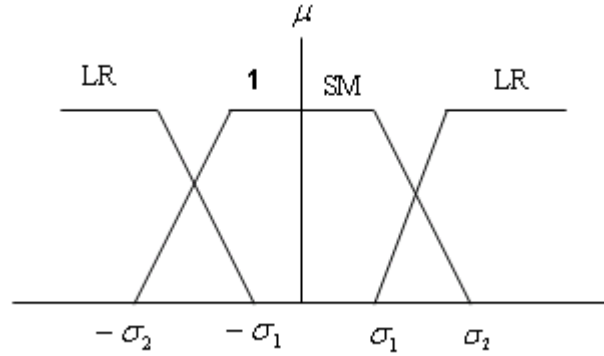


FIGURE 3.1 – Fonctions d'appartenance

- *Base de règles*

La description linguistique du régulateur est donnée par deux règles [Lagrat *et al.*, 2006](Annexe A) qui sont matérialisées par des implications logiques de la forme Si-Alors :

Règle 1 : Si σ est SM Alors

$$u = u_1 = g(\underline{x}, t)^{-1} \left[-\sum_{i=1}^{n-1} c_i e^{(i)} - f(\underline{x}, t) + x_d^{(n)} - k_i \delta - k_p \sigma \right] \quad (3.38)$$

Règle 2 : Si σ est LR Alors

$$u = u_2 = g(\underline{x}, t)^{-1} \left[-\sum_{i=1}^{n-1} c_i e^{(i)} - f(\underline{x}, t) + x_d^{(n)} - k_d \text{sign}(\sigma) \right] \quad (3.39)$$

où σ est la surface de glissement qui peut prendre deux termes SM et LR. L'intérêt de la logique floue est d'intégrer deux structures de lois de commande différentes. La commande par un PI (3.38) et par mode de glissement (3.39) pour assurer la convergence de l'erreur. k_i , k_p et k_d sont les paramètres du contrôleur flou proposé. δ définit l'action du régulateur PI (3.42).

- *Défuzzification*

L'opération de défuzzification est effectuée par la méthode de centre de gravité. On obtient la sortie suivante :

$$u = \frac{\mu_{SM}u_1 + \mu_{LR}u_2}{\mu_{SM} + \mu_{LR}} \quad (3.40)$$

3.6.3 Conception du Régulateur PI

Le régulateur PI est défini comme suit :

$$u_{PI} = k_p\sigma + k_i \int \sigma dt \quad (3.41)$$

Cette solution permet d'éliminer le phénomène de réticence avec les performances de suivi de trajectoire satisfaisantes.

Pour l'analyse du régulateur PI, nous définissons la variable δ :

$$\begin{cases} \delta = \int \sigma dt & \text{si le régulateur PI est activé} \\ \delta \text{ est constante} & \text{si le régulateur PI n'est pas activé} \end{cases} \quad (3.42)$$

soit

$$\delta_r = \frac{d}{k_i} \quad (3.43)$$

Cette expression est déterminée à partir de la boucle fermée dans le sens de réduire l'effet des perturbations inconnues $d(t)$. Dans ce cas, l'objectif de l'action intégrale (3.41) est de faire tendre δ vers δ_r . Afin de remplir cet objectif, δ_r est maximisée par $\delta_{rb}(\delta_{rb} > 0)$:

$$\delta_{rb} = \frac{\max(d)}{k_i} \quad (3.44)$$

Pour effectuer l'analyse de la stabilité, nous choisissons la borne supérieure de δ :

$$|\delta| < 10\delta_{rb} \quad (3.45)$$

L'erreur est définie par :

$$e_\delta = \delta_{rb} - \delta \quad (3.46)$$

Sa dérivée est donnée par :

$$\dot{e}_\delta = \begin{cases} -\sigma & \text{si le régulateur PI est activé} \\ 0 & \text{si le régulateur PI n'est pas activé} \end{cases} \quad (3.47)$$

On considère :

$$|e_\delta| < 11\delta_{rb} \quad (3.48)$$

Substituons (3.38) à (3.34) on trouve :

$$\begin{aligned} x^{(n)} &= f(\underline{x}, t) + g(\underline{x}, t) \left[g(\underline{x}, t)^{-1} \left(-\sum_{i=1}^{n-1} c_i e^{(i-1)} - f(\underline{x}, t) + x_d^{(n)} - k_i \delta - k_p \sigma \right) \right] + d(t) \\ \dot{\sigma} &= k_i \delta - k_p \sigma + d(t) \\ &= k_i (e_\delta - \delta_{rb}) - k_p \sigma + d(t) \end{aligned} \quad (3.49)$$

Où

$$\dot{\sigma} = k_i e_\delta - k_p \sigma \quad (3.50)$$

ce qui conduit à la forme suivante :

$$\begin{bmatrix} \dot{e}_\delta \\ \dot{\sigma} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ k_i & k_p \end{bmatrix} \begin{bmatrix} e_\delta \\ \sigma \end{bmatrix} = A \begin{bmatrix} e_\delta \\ \sigma \end{bmatrix} \quad (3.51)$$

S'il existe une matrice symétrique définie positive P satisfaisant l'équation suivante :

$$A^T P + P A = -Q \quad (3.52)$$

avec A est une matrice stable et Q est une matrice symétrique définie positive.

La fonction de Lyapunov V choisie a la forme :

$$V = \begin{bmatrix} e_\delta & \sigma \end{bmatrix} \begin{bmatrix} P_1 & P_2 \\ P_2 & P_4 \end{bmatrix} \begin{bmatrix} e_\delta \\ \sigma \end{bmatrix} \quad (3.53)$$

D'après l'équation (3.53), $\dot{V} \leq 0$. Ce qui prouve la stabilité du régulateur PI proposé.

D'après la figure (3.1), on constate que la surface de glissement est définie par deux régions $|\sigma| \geq \sigma_2$ et $\sigma_1 \leq |\sigma| \leq \sigma_2$.

Le dérivé de (3.53) donne :

$$\dot{V} = P_1 e_\delta \dot{e}_\delta + P_2 e_\delta \dot{\sigma} + P_4 \sigma \dot{\sigma} \quad (3.54)$$

Pour assurer la stabilité lorsque l'approche de mode de glissement est appliquée, il faut étudier la stabilité dans tout le domaine de σ .

cas 1 : $|\sigma| \geq \sigma_2$

D'après l'équation (3.46), on peut écrire $\dot{e}_\delta = 0$ et par conséquent (3.54) devient :

$$\dot{V} = P_2 \dot{e}_\delta \sigma + P_4 \sigma \dot{\sigma} \quad (3.55)$$

De (3.55), on déduit que $\dot{V} \leq 0$ si les paramètres k_i , k_p et k_d vérifient les conditions suivantes :

$$k_d > k_i \delta_{rb} \quad (3.56)$$

$$\sigma_2 > 11 \delta_{rb} \quad (3.57)$$

$$|P_2| < P_4 \quad (3.58)$$

cas2 : $\sigma_1 \geq \sigma \geq \sigma_2$

De l'équation (3.45), nous remarquons que $\dot{e}_\delta = -\sigma$ dans ce cas (3.52) devient :

$$\dot{V} = -P_1 e_\delta \sigma - P_2 \sigma^2 + P_2 e_\delta \dot{\sigma} + P_4 \sigma \dot{\sigma} \quad (3.59)$$

Pour prouver $\dot{V} < 0$, nous avons choisi la condition suivante :

$$k_d = \frac{2m}{P_4} \quad (3.60)$$

avec $m = 11P_1 \delta_{rb} + |P_2| \sigma_2 + P_4 k_i \delta_{rb}$.

Ce qui conduit à

$$\sigma_1 = 11 \delta_{rb} |P_2| \left(\frac{k_i}{m} + \frac{2}{P_4} \right) \quad (3.61)$$

3.7 Résultats de simulations

3.7.1 Cas du système SISO

Cette section présente un exemple de simulation pour illustrer l'efficacité de la méthode proposée, testée sur le système de Henon [Feng et Chen, 2005] donnée par :

$$y(t) = -y(t-1)^2 + 0.3y(t-2) + 1.4 + u(t-1) \quad (3.62)$$

avec les deux fonctions d'appartenance de la forme :

$$\begin{aligned} \mu_1(y(t-1)) &= 0.5 \left(1 - \frac{y(t-1)}{d} \right) \\ \mu_2(y(t-1)) &= 0.5 \left(1 + \frac{y(t-1)}{d} \right) \end{aligned} \quad (3.63)$$

$d = 0.7$ une constante. Le modèle (3.62) est décrit par les deux règles :

$$\begin{aligned} \text{R\`egle 1 : Si } y(t-1) \text{ est } \mu_1 \text{ Alors } y(t) &= dy(t-1) + 0.3y(t-2) + u(t) + 1.4 \\ \text{R\`egle 2 : Si } y(t-1) \text{ est } \mu_2 \text{ Alors } y(t) &= -dy(t-1) + 0.3y(t-2) + u(t) + 1.4 \end{aligned} \quad (3.64)$$

Les paramètres initiaux ont été choisis comme suit :

$$[y(0) \ y(-1)] = [0 \ 1]^T \text{ et } \hat{\theta}(0) = [d \ 1 \ 1 \ 1.4 \ -d \ 1 \ 1 \ 1.4]^T$$

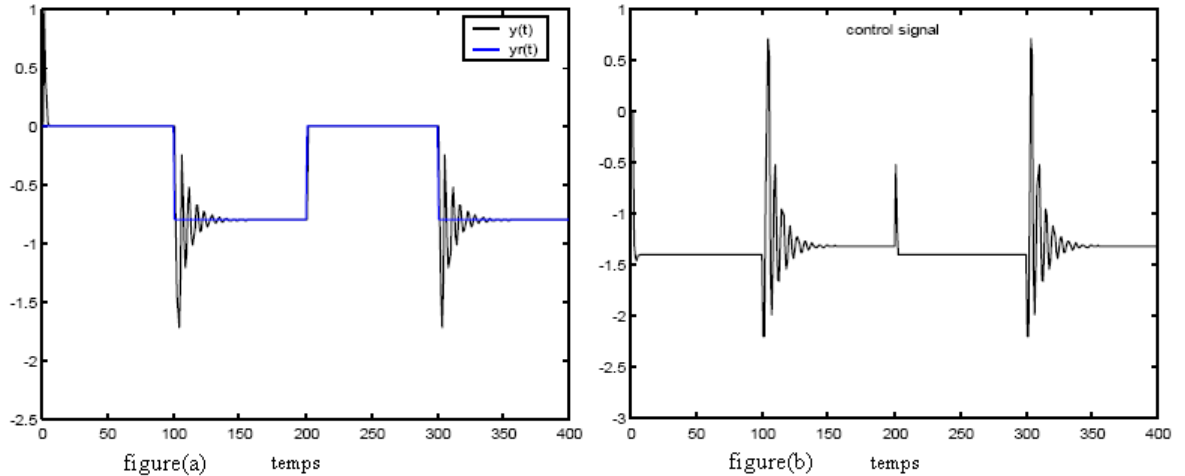


FIGURE 3.2 – Réponse du système sans adaptation des paramètres

La figure (Fig.3.2a) représente la sortie $y(t)$ et le signal de référence $y_r(t)$. Ce dernier est

choisi dans cet exemple carré, de période 200. Nous remarquons que la performance en poursuite est relativement dégradée, ce qui est clair par les différentes fluctuations qui apparaissent sur chaque période, cela est principalement dû à la non adaptation des paramètres. Le signal de commande correspondant est donné sur la figure (Fig.3.2b). Ce problème de fluctuations est

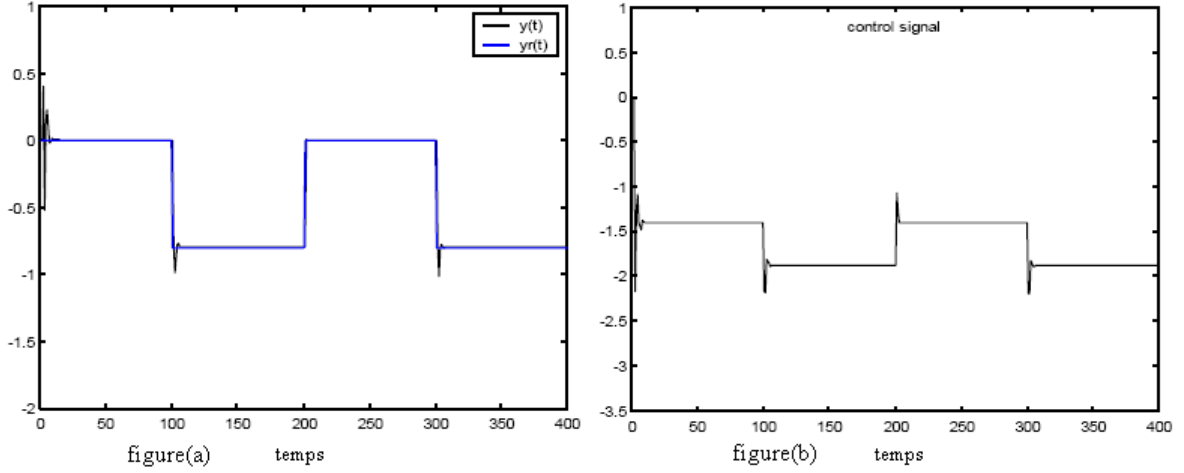


FIGURE 3.3 – Réponse du système avec application de la commande adaptative

facilement surmonté dans le cas où les paramètres sont adaptés par l'algorithme des moindres carrés récursif avec normalisation et projection des données. Les résultats obtenus, en appliquant la même approche avec adaptation des paramètres, sont donnés sur la figure (Fig.3.3). La figure (Fig.3.3a) montre une bonne poursuite de la sortie $y(t)$ à la référence $y_r(t)$, la figure (Fig.3.3b) présente le signal de commande correspondant.

3.7.2 Cas d'un Système MISO

Pour montrer l'efficacité de l'approche proposée pour le cas des systèmes MIMO, nous considérons un système composé de deux entrées et deux sorties. Les entrées sont choisies non linéaires [Weng et Lang, 1990], ce système sera par la suite décomposé en deux sous systèmes MISO.

Considérons le cas général où un système MIMO est décrit par l'équation :

$$A(q^{-1})y(t) = B_d(q^{-1})Z(t) + \xi(t) \quad (3.65)$$

où q^{-1} est l'opérateur retard défini par : $q^{-1}x(t) = x(t-1)$.

Les matrices polynomiales $A(q^{-1})$ et $B_d(q^{-1})$ sont définies par :

$$A(q^{-1}) = \text{diag} [A_i(q^{-1})], \quad \text{et} \quad B_d(q^{-1}) = [q^{-d_{ij}} B_{ij}(q^{-1})]$$

avec $A_i(q^{-1}) = \sum_{r=0}^{na(i)} a_{ir} q^{-r}$, $a_{i0} = 1$, $B_{ij}(q^{-1}) = \sum_{r=0}^{nb(i,j)} b_{ijr} q^{-r}$ et $b_{ij0} = 0$
 d_{ij} représente le temps de retard entre la j -ème entrée et la i -ème sortie. $y(t) \in \mathbf{R}^n$, $Z(t) \in \mathbf{R}^n$
 et $\xi(t) \in \mathbf{R}^n$ sont respectivement la sortie, l'entrée non linéaire et le vecteur de perturbation.

La décomposition du système (3.65) donne :

$$A_i(q^{-1})y_i(t) = q^{-d_{ii}}B_{ii}(q^{-1})Z_i(t) + \sum_{j=1, i \neq j} q^{-d_{ij}}B_{ij}(q^{-1})Z_j(t) + \xi_i(t) \quad (3.66)$$

Les non linéarités considérées sont de la forme [Jingxin et Shijun, 1983] :

$$Z_i(t) = f_{i0} + f_{i1}u_i(t) + \dots + f_{ip}u_i^{p_i-1}(t) \quad (3.67)$$

Nous nous limiterons à un système ayant deux entrées et deux sorties :

$$\begin{aligned} A_1(q^{-1})y_1(t) &= q^{-d_{11}}B_{11}(q^{-1})Z_1(t) + q^{-d_{12}}B_{12}(q^{-1})Z_2(t) + \xi_1(t) \\ A_2(q^{-1})y_2(t) &= q^{-d_{22}}B_{22}(q^{-1})Z_2(t) + q^{-d_{21}}B_{21}(q^{-1})Z_1(t) + \xi_2(t) \end{aligned} \quad (3.68)$$

tel que :

$$\begin{aligned} A &= [A_1 A_2]^T = \begin{bmatrix} 1 + 0.35q^{-1} + 0.15q^{-2} \\ 1 + 0.72q^{-1} + 0.05q^{-2} \end{bmatrix} \\ B &= \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} 1 + 0.85q^{-1} & 2 + 1.25q^{-1} \\ 0.12 + 0.65q^{-1} & 1.65 + 0.23q^{-1} \end{bmatrix} \\ d_{11} &= 1, \quad d_{12} = 1, \quad d_{21} = 1, \quad d_{22} = 1 \end{aligned}$$

Les entrées de sous systèmes sont considérées comme suit :

$$\begin{aligned} Z_1(t) &= 0.5u_1(t) + 0.25u_1^2(t) \\ Z_2(t) &= u_2 + 0.83u_2^2 \end{aligned}$$

Les perturbations sont présentées par : $\xi_1(t) = 0.1rand(1, 600)$, $\xi_2(t) = 0.1rand(1, 600)$.

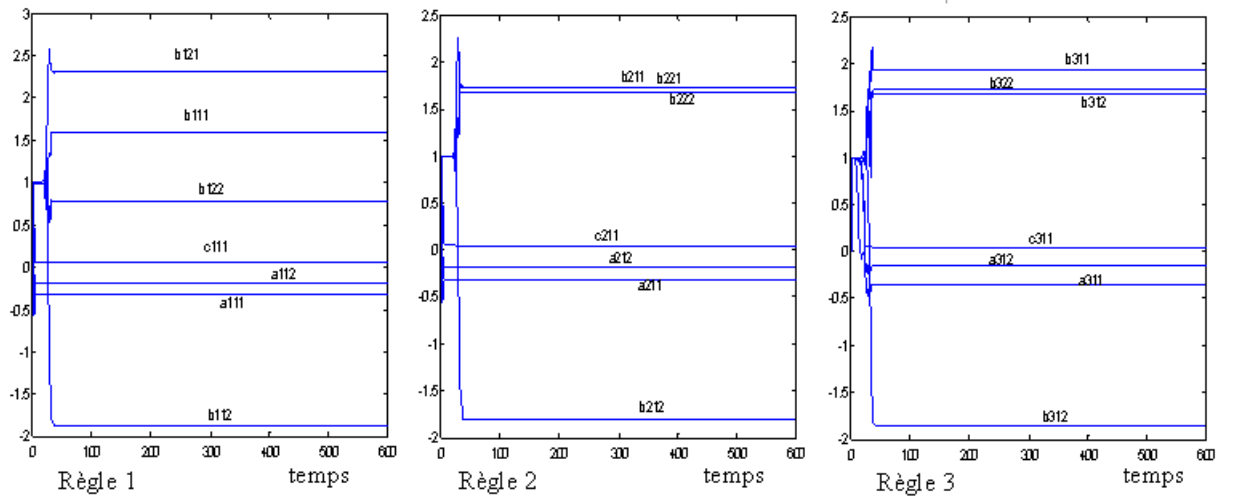
Les matrices de centres et de covariances sont déterminées par l'algorithme de classification floue GK. On choisi le nombre classes égale à $c = 3$ pour les deux sous systèmes $y_1(t)$ et $y_2(t)$. Les paramètres non linéaires de la fonction d'appartenance gaussienne (centres et variances) sont données par :

$$V_1 = \begin{bmatrix} 0.0235 & 0.0720 & 0.0023 & 0.0022 & 0.0041 & 0.0042 \\ 0.0424 & 0.04422 & 0.0022 & 0.0022 & 0.0040 & 0.0040 \\ 0.0846 & 0.0688 & 0.0022 & 0.0022 & 0.0041 & 0.0039 \end{bmatrix}$$

$$F_1 = \begin{bmatrix} 0.0713 & 0.0269 & 8.0335 & 8.4517 & 4.8071 & 4.9064 & 0.0408 \\ 0.1299 & 0.1224 & 6.4028 & 6.8515 & 4.1058 & 3.7856 & 0.0311 \\ 0.3344 & 0.4947 & 12.9454 & 13.1246 & 7.9306 & 8.1032 & 0.0614 \end{bmatrix}$$

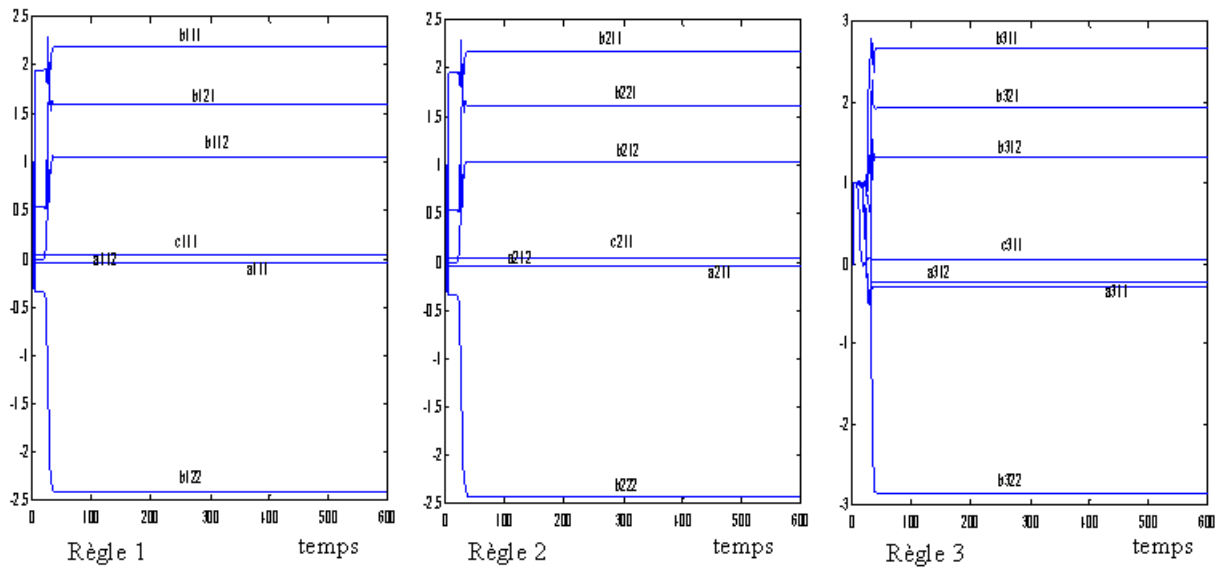
$$V_1 = \begin{bmatrix} 0.0313 & 0.0457 & 0.0023 & 0.0023 & 0.0041 & 0.0042 \\ 0.0378 & 0.0377 & 0.0022 & 0.0022 & 0.0040 & 0.0040 \\ 0.0722 & 0.0883 & 0.0022 & 0.0022 & 0.0040 & 0.0039 \end{bmatrix}$$

$$F_1 = \begin{bmatrix} 0.1878 & 0.1397 & 13.1520 & 11.7753 & 6.1561 & 6.7180 & 0.0669 \\ 0.1956 & 0.1556 & 11.1458 & 10.6673 & 5.6689 & 5.8682 & 0.0579 \\ 0.4412 & 0.2936 & 23.6297 & 25.8100 & 14.0950 & 13.6570 & 0.0970 \end{bmatrix}$$

FIGURE 3.4 – Evolution des paramètres des conséquents de $y_1(t)$ pour les trois règles

Les figures (Fig.3.4) et (Fig.3.5) décrivent respectivement l'évolution des paramètres pour trois règles des sorties estimées $y_1(t)$ et $y_2(t)$. Les paramètres de conclusion de chacune des règles du modèle flou T-S sont ajustés par l'algorithme des moindres carrés récursif avec facteur d'oubli $\lambda = 0.45$.

Les figures (Fig.3.6a) et (Fig.3.6b) représentent le comportement de poursuite des deux systèmes. Les signaux de références y_{1m} et y_{2m} sont carrés de période 100. On remarque qu'une bonne poursuite est obtenue, ce qui prouve l'efficacité de l'approche vis à vis de la compensation des interactions sans aucune hypothèse restrictive sur leur nature.

FIGURE 3.5 – Evolution des paramètres des conséquents de $y_2(t)$ pour les trois règles

Les comportements des deux signaux des excitations riches $u_1(t)$ et $u_2(t)$ sont représentés respectivement sur les figures (Fig.3.7a) et (Fig.3.7b).

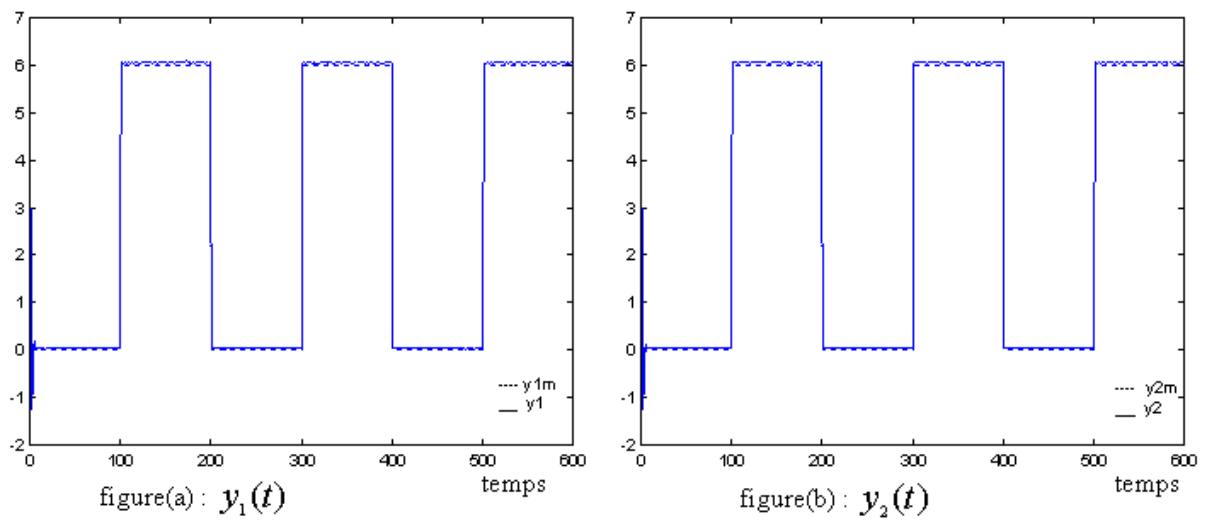


FIGURE 3.6 – Evolution du suivi de trajectoire des signaux des deux systèmes

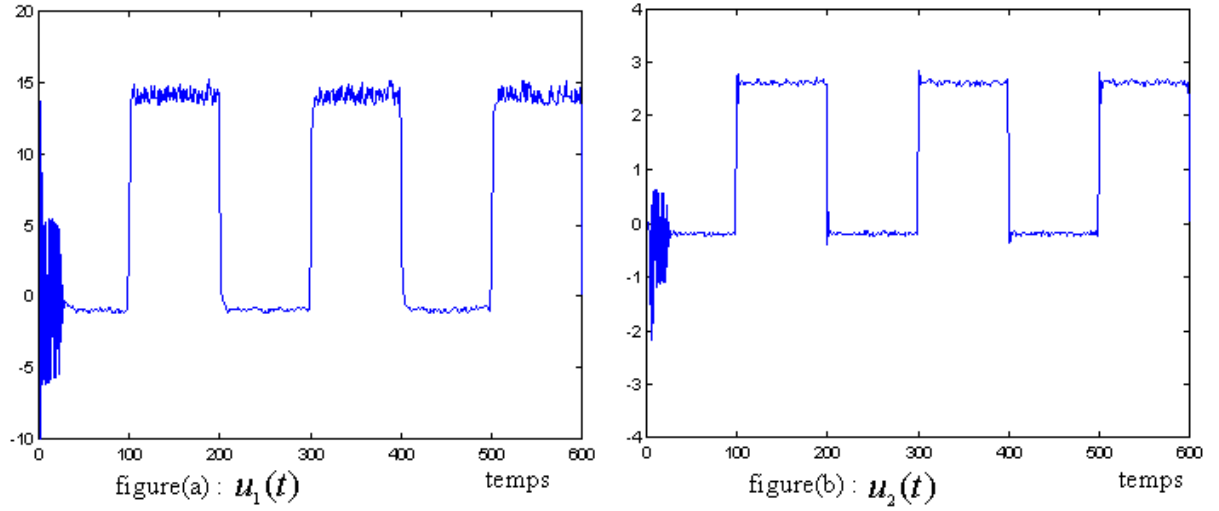


FIGURE 3.7 – Signaux de commande

3.7.3 Cas d'un grand système

Nous allons tester notre approche sur une classe de systèmes non linéaires interconnectés. Pour cela nous considérons deux pendules inversés reliés par un ressort [Spooner-1999]. Chaque pendule peut être positionné par la commande u_i appliquée par un servomoteur à sa base. Il est supposé que les deux mesures ϕ_i et $\dot{\phi}_i$ (position angulaire et sa dérivé) sont disponibles pour le $i^{\text{ème}}$ contrôleur, figure(Fig.3.8).

Les équations qui décrivent les mouvements des pendules sont définies par :

$$\begin{aligned}
 \dot{x}_{11} &= x_{12} \\
 \dot{x}_{12} &= \left(\frac{m_1 g r}{j_1} - \frac{k r^2}{4 j_1} \right) \sin(x_{11}) + \frac{k r}{2 j_1} (l - b) + \frac{u_1}{j_1} + \frac{k r^2}{4 j_1} \sin(x_{21}) \\
 \dot{x}_{211} &= x_{22} \\
 \dot{x}_{22} &= \left(\frac{m_1 g r}{j_2} - \frac{k r^2}{4 j_2} \right) \sin(x_{21}) + \frac{k r}{2 j_1} (l - b) + \frac{u_2}{j_2} + \frac{k r^2}{4 j_2} \sin(x_{12})
 \end{aligned} \tag{3.69}$$

où $x_{11} = \phi_1$ et $x_{21} = \phi_2$ sont les écarts verticaux angulaires des pendules. Les paramètres $m_1 = 2Kg$ et $m_2 = 2.5Kg$ sont les masses des pendules, $j_1 = 0.5Kg$ et $j_2 = 0.625Kg$ sont les moments d'inertie respectivement pour les deux pendules. La constante du ressort reliant les deux pendules est $k = 100N/m$; la taille du pendule est $r = 0.5m$; la longueur normale du ressort est $l = 0.5m$ et la constante de gravité est $g = 9.81m/s^2$. La distance entre les pendules est définie par $b = 0.4m$ (avec $b < 1$ dans cet exemple, de sorte que les pendules se repoussent quand ils se retrouvent tous les deux en position droite).

Ici, nous essayerons de conduire les positions angulaires à zéro, de sorte que $e_i = -\phi_i$ (c.-à-d. $y_{1r} = y_{2r} = 0$) pour $i = 1, 2$. Pour construire l'identificateur flou $u_i(\underline{x}_i, \underline{\theta}_i)$, nous dé-

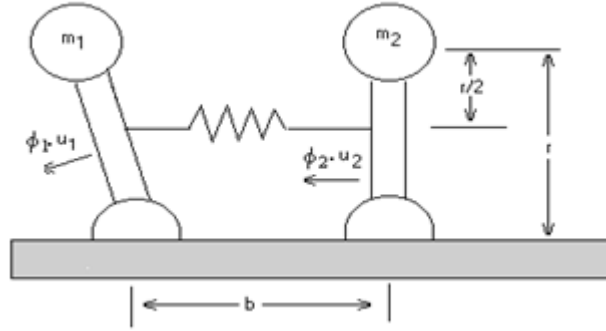


FIGURE 3.8 – Deux pendules inversés connectées par un ressort

finissons cinq sous ensembles flous pour chacune des composantes suivantes : $\underline{x}_1 = (x_{11}, x_{12})$ et $\underline{x}_2 = (x_{21}, x_{22})$, $A_{x_{ij}}^1, \dots, A_{x_{ij}}^5$ représentent les sous ensembles flous, ($j = 1, 2$ et $i = 1, 2$) caractérisés par les fonctions d'appartenance suivantes :

$$\begin{aligned} \mu_{A_{x_{ij}}^1} &= \exp(-(x_{ij} + 0.8)^2) & \mu_{A_{x_{ij}}^2} &= \exp(-(x_{ij} + 0.4)^2) \\ \mu_{A_{x_{ij}}^3} &= \exp(-(x_{ij})^2) & \mu_{A_{x_{ij}}^4} &= \exp(-(x_{ij} - 0.4)^2) \\ \mu_{A_{x_{ij}}^5} &= \exp(-(x_{ij} - 0.8)^2) \end{aligned} \quad (3.70)$$

Du fait qu'on a choisi cinq sous ensembles flous pour chaque composante, nous avons 25 règles floues dont la description linguistique est donnée par :

$$\mathbf{R}_i^l : \text{Si } x_{11} \text{ est } A_{x_{11}}^{k_1} \text{ et } x_{12} \text{ est } A_{x_{12}}^{k_2} \text{ et } x_{21} \text{ est } A_{x_{21}}^{k_1} \text{ et } x_{22} \text{ est } A_{x_{22}}^{k_2} \text{ Alors } y_i^l \text{ est } C_i^l \quad (3.71)$$

où : $k_i = 1, \dots, 5$, $i = 1, 2$ et $l = 1, 2, \dots, 25$

Les paramètres de commande pour la simulation sont choisis comme suit :

Les gains d'adaptation sont fixés à : $\eta_i = 0.01, \eta_{a_i} = 0.001, \eta_{i1} = 0.001, \eta_{i2} = 0.001, \eta_{a_i} = 0.0001$ et $\sigma_i = 0.15$.

la matrice Λ_i est choisie de sorte que le polynôme $\hat{L}(\xi) = \xi^2 + 4\xi + 4$ ait les racines égales à $-2, 2$.

La solution de l'équation de Lyapunov pour $Q_i = \text{diag}(10, 10)$ est

$$P_i = \begin{bmatrix} 6.5625 & -5 \\ -5 & 6.25 \end{bmatrix} \quad (i = 1, 2)$$

Les conditions initiales sont les mêmes pour les deux cas direct et indirect :

$$(x_{11}, x_{12}, x_{21}, x_{22})^T = (1, 1, 1, 1)^T, \theta_{i1} = \theta_{i2} = 0.25 \text{ et } \Gamma_i = [-1 - 0.75 - 0.5 - 0.2500.250.50.751]$$

Les performances de la loi de commande décentralisées indirecte et directe sont respective-

ment représentées sur les figures (Fig.3.9) et (Fig.3.10).

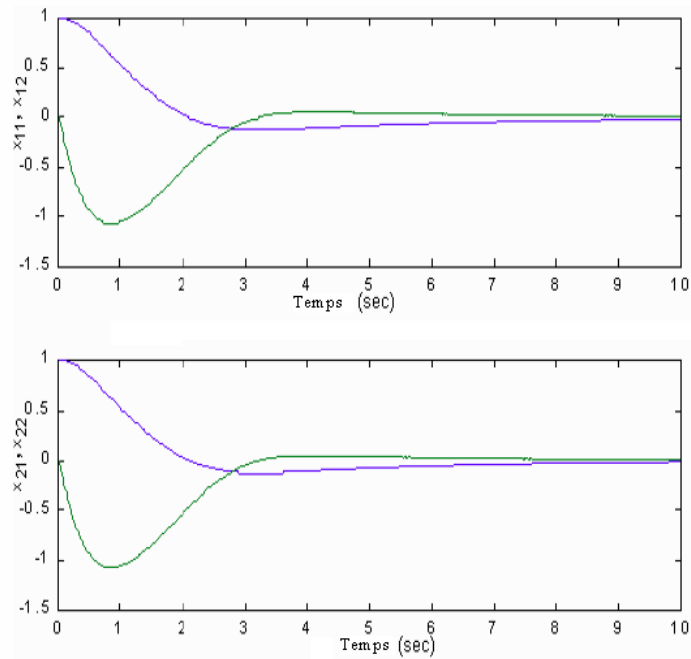


FIGURE 3.9 – Performance de la commande décentralisée utilisant l’approche adaptative directe

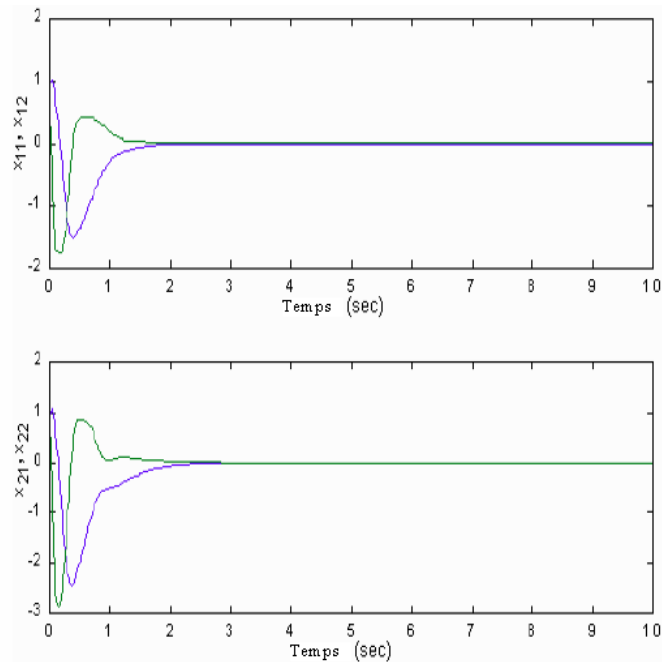


FIGURE 3.10 – Performance de la commande décentralisée utilisant l’approche adaptative indirecte

Les simulations ont montré, dans les deux cas direct et indirect, que l’approche proposée

est très efficace en terme de poursuite. Elle a pu répondre facilement au problème des interconnexions, des non linéarités et des perturbations externes. Cependant la convergence de l'approche indirecte est relativement lente par rapport à celle directe, ce qui est naturellement justifié la faisabilité en deux étapes de l'approche indirecte.

3.7.4 Cas du régulateur PI combiné à l'approche mode glissant

Soit le pendule inversé décrit par le système suivant :

$$x_{(n)} = f(x, t) + g(x, t)u + d(t) \quad (3.72)$$

où

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

$$f(\underline{x}, t) = \begin{bmatrix} 0 \\ \frac{9.8 \sin(x_1) - 0.05 x_2 \cos(x_1) \sin(x_1)}{2/3 - 0.1 \cos^2(x_1)} \end{bmatrix}$$

$$g(\underline{x}, t) = \begin{bmatrix} 0 \\ \frac{-0.1 \cos(x_1)}{2/3 - 0.1 \cos^2(x_1)} \end{bmatrix}$$

$$d(t) = \begin{bmatrix} 0 \\ 0.2 \text{rand}(1, 200) \end{bmatrix}$$

L'objectif est de forcer la sortie x_1 à suivre le signal de référence $y_r = 0.5 \sin(0.015t)$ pour une surface de glissement de la forme $\sigma = 0.3e + \dot{e}$.

Les conditions initiales du système sont $x_1(1) = 1$ et $x_2(1) = 0$

Les paramètres de la loi proposée : $k_i = 2$, $k_p = 3$ et $k_d = 5.918$

La matrice P est obtenue par la résolution de l'équation (3.56) avec :

$$Q = \begin{bmatrix} 2 & 0 \\ 0 & 200 \end{bmatrix} \quad \text{et} \quad P = \begin{bmatrix} 10.166 & 1 \\ 1 & 17.393 \end{bmatrix}$$

La valeur de $\delta_{rb} = 0.1$ implique que $\sigma_2 = 1.1 > 11\delta_{rb}$ et $\sigma_1 = 0.173$. Sur la figure (Fig.3.11) est donnée l'allure de poursuite. Nous remarquons qu'en présence de réticence de fortes amplitudes, la performance est légèrement dégradée.

Les figures (Fig.3.12) et (Fig.3.14) montrent respectivement les signaux de commande par mode de glissement classique et par l'approche proposée.

Une bonne poursuite est obtenue par l'approche proposée comme le montre la figure (Fig.3.13).

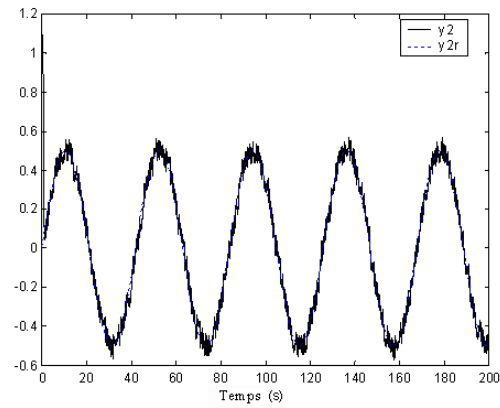


FIGURE 3.11 – Comportement de $y(t)$ et $y_r(t)$ avec le contrôleur de mode de glissement classique

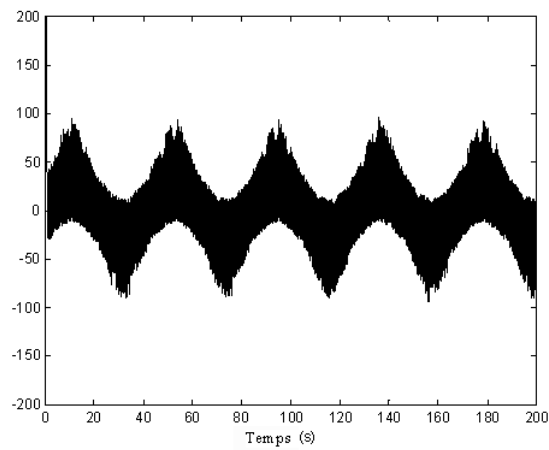


FIGURE 3.12 – Signal de commande correspondant

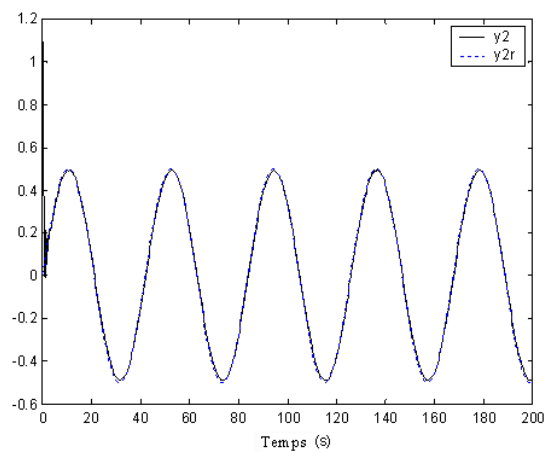


FIGURE 3.13 – Comportement de $y(t)$ et $y_r(t)$ par le régulateur PI et mode de glissement

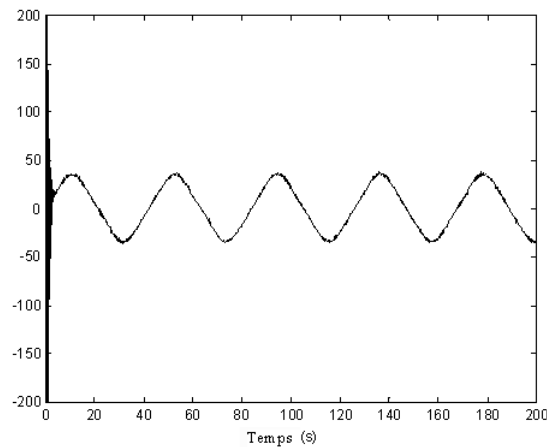


FIGURE 3.14 – Signal de commande correspondant

3.8 Conclusion

Dans ce chapitre, nous avons présenté différentes structures de commandes floues stables et robustes. Ces lois de commande sont destinées aussi bien aux systèmes monovariables qu'aux systèmes multivariables. Notons toutefois, qu'aucune condition restrictive n'est imposée, ni sur la nature des non linéarités ni sur les interactions entre les différentes boucles quand il s'agit des systèmes multivariables et décentralisés, ni sur les perturbations externes. La technique de la logique floue, à chaque fois utilisée, nous a permis de transformer l'étude du problème non linéaire en celui linéaire et d'excellents résultats de poursuite ont été obtenus et justifiés par les différentes simulations effectuées. Les approches proposées peuvent être appliquées à une large classe de systèmes non linéaires.

Conclusion générale

Les travaux présentés dans cette thèse s'intéressent principalement à la commande des systèmes non linéaires, multivariables et décentralisés. La commande de ce type de systèmes soulève un certain nombre d'hypothèses restrictives, sur lesquelles les chercheurs se focalisent actuellement. Parmi ces hypothèses, on cite :

- les non linéarités,
- le couplage croisé dans le cas des systèmes multivariables,
- les interconnexions entre les différents sous systèmes, dans le cas des grands systèmes,
- les perturbations externes qui peuvent affectées le système.

La synthèse d'un contrôleur robuste doit tenir compte de toutes ces conditions restrictives. Les approches proposées dans cette thèse ont prouvé leur efficacité en traitant ces différentes hypothèses restrictives.

La stratégie globale de commande proposée est basée sur deux étapes fondamentales :

- la première concerne la classification des données, elle utilise les mécanismes de la logique floue afin de partitionner les données en classes puis en modèles localement linéaires. Une nouvelle méthode est proposée pour optimiser le nombre de classe, ce qui réduit automatiquement le nombre de modèles locaux et par conséquent le nombre de paramètres à estimer. Cette réduction de paramètres est très bénéfique dans le sens qu'elle réduit la charge des calculs et facilite la mise en oeuvre de l'algorithme. Cette approche fait appel par la suite aux critères de validation puis d'estimation paramétrique afin de déterminer le modèle.
- la deuxième étape concerne le choix de la commande à appliquer en fonction de l'objectif à réaliser. Cette commande prend en considération en dépit des facteurs perturbateurs cités ci-dessus, les dynamiques négligées, les erreurs de modélisation et les perturbations non modélisables.

Notons que les approches proposées sont valables aussi bien pour le cas monovariable que celui multivariable. L'extension au cas multivariable que nous avons adopté est passée par une

décomposition du système MIMO en un ensemble de systèmes MISO. Cette décomposition dont la réalisation est toujours possible a permis de résoudre facilement le problème de l'identification paramétrique, du couplage croisé et des non linéarités.

La stratégie proposée a été élargie aux grands systèmes, chaque sous système est supposé inconnu. Sa commande tient compte du résultat de son approximation et permet à chaque sous système de compenser facilement les termes qui lui sont interconnectés.

Les résultats de simulation obtenus montrent une performance en terme de poursuite et assurent la stabilité et la convergence des schémas de commande proposés, ce qui est d'ailleurs justifié par les améliorations constatées en comparaison avec celles existantes dans la littérature. La simplicité et la facilité de mise en oeuvre des algorithmes proposés laissent prétendre une grande réussite sur le plan pratique.

En perspectives nous proposons :

- D'étendre l'approche proposée pour la détermination du nombre optimal de classes pour des systèmes plus complexes.
- Du fait que le degré d'appartenance n'est qu'une estimation métrique quantifiant la distance d'un point au centre de la classe, nous proposons d'améliorer la précision et le concept de l'appartenance en construisant des fonctions d'appartenance qui tiennent compte de la distance d'un point au centre de la classe et de la distance au modèle linéaire. Cette approche améliorera aussi le concept formulé par les fonctions de type 2 qui considèrent que le degré d'appartenance est imprécis et par suite lui-même flou.
- Les deux points soulevés ci-dessus pourront être utilisés pour développer et perfectionner les algorithmes de classification et d'identification déjà existant afin d'améliorer leur performance et minimiser davantage l'erreur de modélisation floue des systèmes non linéaires.

Bibliographie

- ABONY, J. (2003). *Fuzzy Modeling Identification for Control*. Birkhäuser, Boston.
- ABONYI, F., SZEIFERT, F. et BABUSKA, R. (2002). Modified gath geva fuzzy clustering for identification of takagi-sugeno fuzzy models. *IEEE Transactions on Systems, Man, and Cybernetics*, 32(5):612–621.
- ALEXIUK, M. et PIZZI, N. (2004). Cluster validation indices for fmri data : fuzzy cmeans with feature partitions versus cluster merging strategies. *IEEE NAFIPS conference*.
- ANGELOV, P. et FILEV, D. (2004). An approach to online identification of takagi-sugeno fuzzy models. *IEEE Trans. Syst., Man, Cybern. Part B* :, 34(1):484–498.
- BABUSKA, R. (1998). *Fuzzy Modeling for Control*. Kluwer Academic Publishers, Mass, USA.
- BABUSKA, R., ROUBOIS, J. et VERBRUGGEN, H. (1998). Identification of mimo systems by input-output ts models. *IEEE International Conference on Fuzzy Systems, Anchorage, USA*, 1:657–662.
- BABUSKA, R. et VERBRUGGEN, H. (1995). Identification of composite linear models via fuzzy clustering. *Proc. European Control Conference, Rome, Italy*, pages 207–1212.
- BABUSKA, R. et VERBRUGGEN, H. (1997). Constructing fuzzy models by product space clustering, in fuzzy model identification, selected approaches. *H. Hellendoorn and D. Driankov, Eds, Berlin, Germany : Springer-Verlag*, pages 53–90.
- BENSAID, A., HALL, L., BEZDEK, J., CLARKE, L., SILBIGER, M., ARRINGTON, J. et MURTAGH, R. (1996). Validity-guided (re)clustering with applications to image segmentation. *IEEE Transactions on Fuzzy Systems*, 4:112–123.
- BEZDEK, J. (1974a). Cluster validity with fuzzy sets. *J. Cybernet*, 3:58–73.
- BEZDEK, J. (1974b). Numerical taxonomy with fuzzy sets. *Journal Math. Biology*, 1:57–71.
- BEZDEK, J. (1981). Pattern recognition with fuzzy objective function algorithms. *Plenum Press, New York*.

-
- BEZDEK, J. et DUNN, J. (1975). Optimal fuzzy partitions : A heuristic for estimating the parameters in a mixture of normal distribution. *IEEE Trans. Computers*, C-24, pages 835–838.
- BEZDEK, J., HATHAWAY, R., HOWARD, R., WILSON, C. et WINDHAM, M. (1987). Local convergence analysis of a grouped version of coordinate descent. *Journal of Optimization Theory and Application*, 54(3):471–477.
- BORTOLET, P. et PALM, R. (1997). Identification, modeling and control by means of takagi-sugeno fuzzy systems. *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Spain*, pages 515–520.
- BUCKLEY, J. (1992). Universal fuzzy controllers. *Automatica*, 28.
- CASTRO, J. (1995). Fuzzy logic controllers are universal approximator. *IEEE Transactions on Systems Man and Cybernetics-part C*, 25:629–635.
- C.FRANCO, SILVA, L. et OLIVEIRA, A. (2002). A validity measure for hard an fuzzy clustering derived from fisher’s linear discriminant. *International Conference on Fuzzy Systems*.
- CHAOUI, F., GIRI, F., DION, J., SAAD, M. et DUGARD, L. (2000). Direct adaptive control subject to input amplitude constraint. *IEEE Trans. on Automatic Control*, pages 485–490.
- CHAOUI, F. Z., GIRI, F. et M’SAAD, M. (2001). Adaptive control of input-constrained type-1 plants stabilization and tracking. *Automatica*, pages 197–203.
- DUNN, J. (1974). A fuzzy relative to isodata process and its use in detecting compact well separated clusters. *Journal of Cybernetics*, 3(3):32–57.
- ERRAHMANI, A., OUAJKA, H., BENYAKHLEF, M. et BOUMHIDI, I. (2007). Decentralized adaptive fuzzy control for a class of nonlinear systems. *WSEAS Transaction on Systems and Control*, 8(2):411–418.
- FENG, G. et CHEN, G. (2005). Adaptive control of discrete-time chaotic systems : a fuzzy control approach, chaos. *Solutions and Fractals*, 23:459–467.
- FORNBERG, B. et ZUEV, J. (2007). The runge phenomenon and spatially variable shape parameters in rbf interpolation. *Computers and Mathematics with Applications*, 54(3):379–398.
- FRIGUI, H. et KRISHNAPURAM, R. (1996). A robust clustering algorithm based on competitive agglomeration and soft rejection of outliers. *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition-CVPR 96*, pages 550–555.
- GASSO, K. (2000). *Identification des systèmes dynamiques non linéaires : approche multi-modèles*. Thèse de doctorat de l’Institut National Polytechnique de Lorraine, Vandoeuvre-lès-Nancy.
- GHAT, I. et GEVA, A. (1989). Unsupervised optimal fuzzy clustering. *IEEE Trans. Pattern Anal. Machine Intell*, 7:773–781.

-
- GUESMI, K., ESSOUNBOULI, N., MANAMANNI, N., HAMZAOUI, A. et ZAYTOON, J. (2005). A fuzzy logic controller synthesis for a boost converter. *IFAC 05 World Congress, Prague*.
- GUILLÉN, A., GONZÁLEZ, J., ROJAS, I., POMARES, H., HERRERA, L., VALENZUELA, O. et PRIETO, A. (2007). Using fuzzy logic to improve a clustering technique for function approximation. *Neurocomputing*, 70(16-18):2853–2860.
- GUSTAFSON, E. et KESSEL, W. (1978). Fuzzy clustering with a fuzzy covariance matrix. *IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes, San Diego*, pages 761–766.
- HADAD, A., GHIDARY, S., BAHRAMI, S. et SHAHBAZI, S. (2006). A modification of sugeno yasukawa modeler to improve structure identification phase. *ICGST ACSE*, 6(3):33–40.
- JAIN, A. et DUBES, R. (1988). *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, NJ.
- JANG, J. (1993). Anfis : adaptive network-based fuzzy inference systems. *IEEE Transaction on Systems, Man, Cybern*, 23(3):665–685.
- JINGXIN, Z. et SHIJUN, L. (1983). Adaptive control weighted suboptimal control for linear dynamic systems having polynomial input. *IEEE Trans. Automatic Control*, 28.
- KAYMAK, U. et BABUSKA, R. (1995). Compatible cluster merging for fuzzy modelling. *In Proceedings of the FUZZ IEEE IFES 95, Yokohama, Japan*, pages 897–904.
- KIM, E., PARK, P., JI, S. et PARK, M. (1997). A new approach to fuzzy modelling. *IEEE Trans on Fuzzy Systems*, 5(3):328–337.
- KUKOLJ, D. et LEVI, E. (2004). Identification of complex systems based on neural and takagi-sugeno fuzzy model. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(1):272–282.
- LAGRAT, I., OUAJKA, H. et BOUMHIDI, I. (2006). Fuzzy sliding mode pi controller for nonlinear systems. *WSEAS Transaction on Signal Processing*, 9(2):1137–1143.
- LAGRAT, I., OUAJKA, H. et BOUMHIDI, I. (2007a). Adaptive control of a class of nonlinear systems based on takagi-sugeno fuzzy models. *Proceeding ICTIS 07, Fez, Morocco*, pages 594–597.
- LAGRAT, I., OUAJKA, H. et BOUMHIDI, I. (2007b). Fuzzy clustering for identification of takagi-sugeno fuzzy models of a class of nonlinear multivariable systems. *Proceeding System Electronique-Informatique et de l'information, SETTI 07, Mohammedia, Maroc*, pages 210–215.
- LAGRAT, I., OUAJKA, H. et BOUMHIDI, I. (2008). Fuzzy adaptive control of a class of MISO nonlinear systems. *Control and cybernetics*, 37(1):59–73.
- LANDAU, I. et KARIMI, A. (1997). An output error recursive algorithm for unbiased identification in closed loop. *Automatica*, 33(5):933–938.

-
- MAMDANI, E. H. et ASSILIAN, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of Man-Machine Studies*, 7:1–13.
- MAULIK, U. et BANDYOPADHAY, S. (2002). Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12).
- MENDEL, J. (2000). Uncertainty, fuzzy logic, and signal processing. *Signal processing*, 80(6): 913–933.
- MEUNIER, B. (1995). *La logique floue et ses applications*. Addison.
- MORÈRE, Y. (2000). Mise en oeuvre de lois de commandes pour les modèles flous de type takagi-sugeno. *Doctorat de l'université de Valenciennes et du Haut Cambrésis, Valenciennes*.
- MURRAY-SMITH, R. et JOHANSEN, T. (1999). *Multiple model Approaches to Modelling and Control*. Taylor & Francis, London.
- OUAKKA, H. et BOUMHIDI, I. (2009). Optimal TS models for identification of nonlinear systems from input-output data. *ICGST International Journal on Automatic Control and Systems Engineering, ACSE*, 09(I):1–7.
- PAGÈS, O. et HAJJAJI, A. E. (2005). Two fuzzy multiple reference model tracking control designs with an application to vehicle lateral dynamics control. *IEEE Conf. on Decision and Control, and the European Control Conference*, pages 3267–3272.
- PRALY, L. (1983). Robustness of indirect adaptive control based on pole placement design. *Solutions and Proc. Of the 1st IFAC Workshop on adaptive systems in control and signal processing, San Francisco, USA*.
- RIID, A., ISOTAMM, R. et RUSTERN, E. (2001). Transparency analysis of first-order takagi-sugeno systems. *Proceedings of 10th International Conference on System Modelling Control*, pages 165–170.
- RIID, A. et RÜSTERN, E. (2004). Takagi-sugeno systems with improved interpretability and interpolation using a complementary interpolation model. *Proceedings of 4th Intelligent Systems Design and Applications*, pages 259–264.
- RUPSINI, E. (1969). A new approach to clustering. *Information and Control*, 15:22–32.
- SLOTINE, J. et LI, W. (1991). Applied nonlinear control. *Prentice Hall*.
- SUGENO, M. et KANG, G. (1988). Structure identification of fuzzy model. *Fuzzy Sets and systems*, 28.
- SUN, S., WANG, S. et JIANG, Q. (2004). Fcm-based model selection algorithms for determining the number of clusters. *Pattern recognition*, 37(10):2027–2037.
- TAFAZOLI, S., LEDUC, M. et BAFAS, X. (2006). Modeling using fuzzy subtractive clustering. *Computational cognition*, 4(3):15–27.

-
- TAKAGI, T. et SUGENO, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on System Man and Cybernetics*, 15(1):116–132.
- TRAUWAERT, E. (1988). On the meaning of dunn's partition coefficient for fuzzy clusters. *Fuzzy Sets Systems*, 25:217–242.
- TSEKOURAS, G. (2005). On the use of the weighted fuzzy c-means in fuzzy modelling. *Advances in Engineering Software*, 36(5):287–300.
- TSEKOURAS, G., SARIMVEIS, H., KAVAKLI, E. et BAFAS, G. (2005). A hierarchical fuzzy clustering approach to fuzzy modelling. *Fuzzy Sets and Systems*, 150(23):245–266.
- UTKIN, V. (1992). Sliding modes in control optimization. *Springer Verlag, USA*.
- WANG, L. X. et MENDEL, J. (1992). Fuzzy basis functions, universal approximation, and orthogonal least squares learning. *IEEE Transactions on Neural Networks*, 3:807–814.
- WENG, F. et LANG, S. (1990). Globally convergent direct adaptive control algorithm for multivariable systems with general time delay structure. *International Journal of Control*, 51(2):301–314.
- YOUNG, L., DAE, W., LEE, L. D. et KWANG, H. (2004). A cluster validation index for gk cluster analysis based on relative degree of sharing. *Informatics and Computer Science*, 168(4):225–242.
- ZADEH, L. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. on Systems, Man, and Cybernetics*, 1:28–44.
- ZENG, X. et SINGH, M. (1995). Approximation theory of fuzzy systems mimo case. *IEEE Transactions on Fuzzy Systems*, 3(2):219–235.

Annexe A

Fuzzy Sliding Mode PI Controller for Nonlinear Systems

I. Lagrat, H. Ouakka et I. Boumhidi

***WESAS TRANSACTION ON SIGNAL PROCESSING
Issue 9, Vol 2, pp.1137-1143, September 2006.***

Paru aussi dans:

***Proceedings of the 6th WSEAS International Conference on Simulation,
Modelling and Optimization, pp. 534-539, Lisbon, Portugal, 2006.***

Fuzzy Sliding Mode PI Controller for Nonlinear Systems

I. LAGRAT, H. OUAKKA, I BOUMHIDI

Département de physique
LESSI, Faculté Des Sciences
B. P 1796 Atlas, 30000 Fez
MOROCCO

E-mail : i.lagrat@caramail.com

Abstract: - in this paper, we present a fuzzy logic controller to combine sliding mode controller and PI controller for nonlinear systems with external disturbances. A sliding mode controller can give good transient performance and PI offer zero steady state error. However, the sliding mode controller suffers from a well known problem chattering due to high gain and high-speed switching control. The undesirable chattering may excite previously unmodelled system dynamics causes the poor performance of the steady state. Hence, combining these two controllers by a fuzzy logic can combine their advantages and remove their disadvantages. The proposed method can efficiently to eliminate the chattering in sliding phase so that high performance can be achieved. Some simulation results prove the validity of the proposed method.

Key-words: - Fuzzy logic, PI control, sliding mode control, chattering, nonlinear system.

1 Introduction

The control of nonlinear systems has been an important research topic and many approaches have been proposed [1], [2]. The sliding mode control theory of the variable structure system provides a method to design a system in such a way that the controller system should be insensitive to parameter variations and external disturbances [3]. Essentially, the sliding mode control uses discontinuous control action to drive the state trajectory toward a specific hyper plane in the state space, and then the state trajectory is maintained to slide on the specific hyper plane until the origin of the state space is reached. In the sliding mode control, the hitting time of the system state reaches the switching plane will affect the speed of the system with the desired dynamic behaviour. Sliding mode control (SMC) is well known for handling matched uncertainties [3], [4]. A sliding-mode control law is formulated using a Lyapunov approach to guarantee that the system state first reaches the prescribed sliding mode in finite time from any initial state, and then remains on it there after by a discontinuous control. However, SMC suffers from a well known problem chattering due to the high gain and high-speed switching control. The undesirable chattering may excite previously unmodelled system dynamics and

damage actuators, resulting in unpredictable instability. One method to alleviate this drawback is to introduce a boundary layer about the sliding plane [5]. This method can lead to stable close loop system without the chattering problem, but there exists a finite steady state error due the finite steady gain of the control algorithm. Another class of techniques is based on the use of an observer [9]. However, state observer can cause loss robustness. The high-order sliding mode approach, known as r-sliding mode is also used [10], [11]. However, the discontinuity set of controllers is stratified union of manifolds with codimension varying in the rang from 1 to the relative degree r . unfortunately, the complicated structure of the controller discontinuity set causes certain redundant transient chattering. In the past several years, active research has been carried out in controller design based on universal approximators, such as fuzzy control and neural network control [12], [13]. Due to the robustness property as well as wide applications of SMC technique. The research of fuzzy model under SMC has attracted many attentions in recent years [14], [15]. The apparent similarities between sliding mode control and fuzzy control motivate considerable research efforts in combining the

two approaches for achieving more superior performances such as overcoming some limitations of the traditional sliding mode control. As a model free design method, fuzzy systems have been as a model free design method [7], fuzzy systems have been successfully applied to control complex or ill-defined processes whose mathematical models are difficult to obtain [6], [7]. The ability of converting linguistic descriptions into automatic control strategy makes it a practical and promising alternative to the classical control scheme for achieving control of complex nonlinear systems. To eliminate steady-state error, a PI controller should be employed. This paper proposes a fuzzy logic controller (FLC) to combine an SMC and a PI controller. As the SMC and PI controllers can give good transient and steady-state performance respectively, the role of the FLC is to schedule them under different operation conditions [7], [8]. The remaining of this paper is organised as follow. In section 2, problem statements. The proposed fuzzy SMC and PI control will be developed in Section 3. Section 4, a plant is used to test the proposed FLC method and some compared results demonstrate its feasibility. Finally, Section 5 concludes the paper.

2 Problem statements

In this section, the variable structure system with sliding mode control is briefly reviewed.

Consider a general class of SISO n -th order nonlinear systems as follow:

$$\begin{aligned} \dot{x}^{(n)} &= f(\underline{x}, t) + g(\underline{x}, t)u + d(t) \\ y &= x \end{aligned} \quad (1)$$

where f and g are nonlinear functions, $\underline{x} = [x, \dot{x}, \dots, x^{(n-1)}]^T = [x_1, x_2, \dots, x_n]^T \in R^n$ is the state vector of the systems which is assumed to be available for measurement, $u \in R, y \in R$ are the input and the output of the system, respectively, and $d(t)$ is the unknown external disturbance. We have to make an assumption that $d(t)$ have upper bound D , that is, $|d(t)| \leq D$. We require the system (1), to be controllable, the input gain $g(\underline{x}, t) \neq 0$ is necessary. Hence, without loss of generality, we are assumed $g(\underline{x}, t) > 0$. The control problem is to obtain the state $\underline{x}$ for

tracking a desired state $\underline{x}_d$ in the presence of model uncertainties and external disturbance with the tracking error:

$$\underline{e} = \underline{x} - \underline{x}_d = [e, \dot{e}, \dots, e^{(n-1)}]^T \in R^n \quad (2)$$

Define a sliding surface in the space of the error state as:

$$\begin{aligned} \sigma &= c_1 e + c_2 \dot{e} + \dots + c_{n-1} e^{(n-2)} + e^{(n-1)} \\ &= \underline{c}^T \underline{e} \end{aligned} \quad (3)$$

Where $\underline{c} = [c_1, c_2, \dots, c_{n-1}, 1]^T$ are the coefficients of the Hurwitz polynomial $H(z) = z^{n-1} + c_{n-1}z^{n-2} + \dots + c_1$, i.e., all the roots are in the open left half-plane and z are a Laplace operator. If the initial condition $\underline{e}(0) = 0$, the tracking problem

$\underline{x} = \underline{x}_d$ can be considered as the state error vector remaining on the sliding surface $\sigma = 0$ for all $t > 0$.

A sufficient condition to achieve this behaviour is to select the control strategy such that.

$$\frac{1}{2} \frac{d}{dt} (\sigma^2) \leq -k_d |\sigma|, \quad k_d \geq 0 \quad (4)$$

The system is controlled in such a way that the state always moves towards the sliding surface and hits it. The sign of the control value must change at the intersection between the state trajectory and sliding surface.

Consider the control problem of nonlinear systems (1), if $f(\underline{x}, t)$ and $g(\underline{x}, t)$, are known. The SMC input u .

$$u = \frac{1}{g(\underline{x}, t)} \left[- \sum_{i=1}^{n-1} c_i e^{(i)} - f(\underline{x}, t) + \dot{x}_d^{(n)} - k_d \operatorname{sgn}(\sigma) \right] \quad (5)$$

Let the Lyapunov function candidate defined

$$\text{as: } V_1 = \frac{1}{2} \sigma^2 \quad (6)$$

Differentiating (6) with respect to time, $\dot{V}_1$ along the system trajectory as:

$$\begin{aligned} \dot{V}_1 &= \sigma \dot{\sigma} \\ &= \sigma \cdot (c_1 \dot{e} + c_2 \ddot{e} + \dots + c_{n-1} e^{(n-1)} + \dot{x}^{(n)} - \dot{x}_d^{(n)}) \\ &= \sigma \cdot \left(\sum_{i=1}^{n-1} c_i e^{(i)} + f(\underline{x}, t) + g(\underline{x}, t)u + d(t) - \dot{x}_d^{(n)} \right) \\ &\leq -k_d |\sigma| \end{aligned} \quad (7)$$

Hence the SMC input u guarantees the sliding condition of (4). It is obvious that in order to satisfy the sliding condition, a hitting control term u_{sw} must be added i.e. $u = u_{eq} - u_{sw}$.

Where

$$u_{eq} = g(\underline{x}, t)^{-1} \left[- \sum_{i=1}^{n-1} c_i e^{(i)} - f(\underline{x}, t) + \dot{x}_d^{(n)} \right] \quad (8)$$

$$u_{sw} = g(\underline{x}, t)^{-1} \cdot k_d \operatorname{sgn}(s) \quad (9)$$

The result in the control law (5) for a nonlinear plant, the switching-type control term u_{sw} will cause chattering problem. To solve these problems, we propose the fuzzy sliding mode control algorithm using the fuzzy logic system and the PI control law in section 3.

3 Design fuzzy sliding mode PI control

A SMC and a PI controller are combined into a single FLC to control a nonlinear system (1).

We employ PI control term in order to avoid chattering problem. The input and output of the continuous time PI controller is in the form of:

$$u_{PI} = k_p \sigma + k_i \int \sigma dt \quad (10)$$

We define a state δ which will be used on analysing the system with the PI controller as:

$$\begin{cases} \delta = \int \sigma dt & \text{when the PI controller is active} \\ \delta \text{ is constant} & \text{when the PI controller is inactive} \end{cases} \quad (11)$$

δ_r is the reference value of δ . It is constant to cancel out the effect of the unknown disturbance d when the sliding plane is hit. Hence we have

$$\delta_r = \frac{d}{k_i} \quad (12)$$

Where k_i is a gain to be designed later. In practice, due to integral action as given by (11), the state δ will automatically become δ_r under proper designer of the controller when the sliding plane is hit. The maximum bound δ_{rb} can be evaluated as:

$$\delta_{rb} = \frac{\max(d)}{k_i} \quad (13)$$

with $\delta_{rb} > 0$. To carry out the stability analysis, we choose an upper bound for δ :

$$|\delta| < 10 \delta_{rb} \quad (14)$$

The state error is defined as follows:

$$e_\delta = \delta_{rb} - \delta \quad (15)$$

Then from (11).

$$\dot{e}_\delta = \begin{cases} -\sigma & \text{when the PI controller is active} \\ 0 & \text{when the PI controller is inactive} \end{cases} \quad (16)$$

$$\text{Hence } |e_\delta| < 11 \delta_{rb} \quad (17)$$

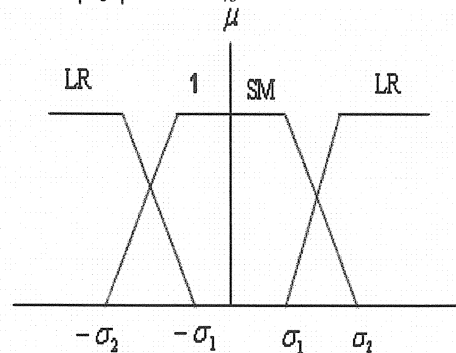


Fig.1 Membership functions

A SMC and a PI controller are combined into FLC, and then the fuzzy rules are givens as:

Rule1: if σ is SM then

$$u = u_1 = g(\underline{x}, t)^{-1} \left[- \sum_{i=1}^{n-1} c_i e^{(i)} - f(\underline{x}, t) + \dot{x}_d^{(n)} - k_i \delta - k_p \sigma \right] \quad (18)$$

Rule2: if σ is LR then

$$u = u_2 = g(\underline{x}, t)^{-1} \left[- \sum_{i=1}^{n-1} c_i e^{(i)} - f(\underline{x}, t) + \dot{x}_d^{(n)} - k_d \operatorname{sgn}(\sigma) \right] \quad (19)$$

where SM and LR are membership functions as shown in figure.1. k_i , k_p and k_d are gains to be designed.

3.1 PI sub-system analysis

From Rule 1, (1) and (18) we have

$$\dot{x}^{(n)} = f(\underline{x}, t) + g(\underline{x}, t) [g(\underline{x}, t)^{-1} (- \sum_{i=1}^{n-1} c_i e^{(i-1)} - f(\underline{x}, t) + \dot{x}_d^{(n)} - k_i \delta - k_p \sigma)] + d(t)$$

$$\dot{\sigma} = k_i \delta - k_p \sigma + d(t)$$

$$= k_i (e_\delta - \delta_{rb}) - k_p \sigma + d(t) \quad (20)$$

Hence, from (15),

$$\dot{\sigma} = k_i e_\delta - k_p \sigma \quad (21)$$

$$\begin{bmatrix} \dot{e}_\delta \\ \dot{\sigma} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ k_i & -k_p \end{bmatrix} \begin{bmatrix} e_\delta \\ \sigma \end{bmatrix} = A \begin{bmatrix} e_\delta \\ \sigma \end{bmatrix} \quad (22)$$

We can define a symmetric positive definite matrix Q and if the real parts of all eigenvalues of A are negative, we can be found a symmetric positive definite matrix P , satisfying the following equation:

$$A^T P + P A = -Q \quad (23)$$

Also from (23), we have define a common Lyapunov function V such that

$$V = [e_\delta \quad \sigma] \begin{bmatrix} P_1 & P_2 \\ P_2 & P_4 \end{bmatrix} \begin{bmatrix} e_\delta \\ \sigma \end{bmatrix} \quad \text{Where}$$

$$P = \begin{bmatrix} P_1 & P_2 \\ P_2 & P_4 \end{bmatrix} \quad (24)$$

Obviously from (23) and (24), $\dot{V} \leq 0$ such that this PI sub-system is stable.

3.2 SMC sub-system analysis

With reference to Fig.1, there are two sub-region defined by $|\sigma| \geq \sigma_2$ and $\sigma_1 \leq |\sigma| \leq \sigma_2$. From (24), we have

$$\dot{V} = P_1 e_\delta \dot{e}_\delta + P_2 \sigma \dot{e}_\delta + P_2 e_\delta \dot{\sigma} + P_4 \sigma \dot{\sigma} \quad (25)$$

To ensure the system stability of SMC sub-system under the Lyapunov function of (24) that $\dot{V} \leq 0$ in this sub-system, we divide study in two sub-regions.

Case1: $|\sigma| \geq \sigma_2$

From (16), $\dot{e}_\delta = 0$. Then (25) can be reduced to $\dot{V} = P_2 e_\delta \dot{\sigma} + P_4 \sigma \dot{\sigma}$ (26)

It can be proved that $\dot{V} \leq 0$ if kd, k_i and k_p satisfied the following three conditions:

$$k_d > k_i \delta_{rb} \quad (27)$$

$$\sigma_2 > 11 \delta_{rb} \quad (28)$$

$$|P_2| < P_4 \quad (29)$$

Proof 1:

Form Rule 2, (1) and (3), we have

$$\sigma \dot{\sigma} = \sigma (d - kd \operatorname{sign}(\sigma)) \quad (30)$$

$$\leq |\sigma| |d| - kd |\sigma|$$

From (27) and (14),

$$kd > k_i \delta_{rb} = \max(d)$$

Hence $\sigma \dot{\sigma} < 0$ (31)

Also, since $|\sigma| \geq \sigma_2$. From (28) and (17),

$$\text{then } |\sigma| > |e_\delta|.$$

Consider (26),

$$\begin{aligned} \dot{V} &= P_2 e_\delta \dot{\sigma} + P_4 \sigma \dot{\sigma} \\ &\leq |P_2 e_\delta \dot{\sigma}| + P_4 \sigma \dot{\sigma} \\ &= |P_2| |e_\delta| |\dot{\sigma}| + P_4 \sigma \dot{\sigma} \\ &< |P_2| |\sigma \dot{\sigma}| - P_4 \sigma \dot{\sigma} \\ &= |P_2| |\sigma \dot{\sigma}| - P_4 |\sigma \dot{\sigma}| \\ &< 0 \end{aligned}$$

Case2: $\sigma_1 \leq |\sigma| \leq \sigma_2$

From (16), $\dot{e}_\delta = -\sigma$. then (25) becomes

$$\dot{V} = -P_1 e_\delta \sigma - P_2 \sigma^2 + P_2 e_\delta \dot{\sigma} + P_4 \sigma \dot{\sigma} \quad (32)$$

To ensure (32), it can be negative if the following conditions

$$kd = \frac{2m}{P_4} \quad (33)$$

where, $m = 11 P_1 \delta_{rb} + |P_2| \sigma_2 + P_4 k_i \delta_{rb}$.

$$\sigma_1 = 11 \delta_{rb} |P_2| \left(\frac{k_i}{m} + \frac{2}{P_4} \right) \quad (34)$$

Proof 2:

From $\sigma_1 \leq |\sigma| \leq \sigma_2$, $\dot{e}_\delta = -\sigma$. then (25)

becomes:

$$\begin{aligned} \dot{V} &= -P_1 e_\delta \sigma - P_2 \sigma^2 + P_2 e_\delta \dot{\sigma} + P_4 \sigma \dot{\sigma} \\ &= -P_1 e_\delta \sigma - P_2 \sigma^2 + P_2 e_\delta \dot{\sigma} + P_4 \sigma d - P_4 kd |\sigma| \quad \text{Let} \end{aligned}$$

$$\leq |\sigma| \max(-P_1 e_\delta \sigma - P_2 \sigma - P_4 d) - P_4 kd |\sigma| + P_2 e_\delta \dot{\sigma}$$

$$m = 11 P_1 \delta_{rb} + |P_2| \sigma_2 + P_4 k_i \delta_{rb} \quad \text{and}$$

$$kd = \frac{2m}{P_4}$$

From (13) and (17),

$$11 P_1 \delta_{rb} > P_1 |e_\delta|$$

$$P_4 k_i \delta_{rb} = \max(P_4 d)$$

$$\sigma_2 > |\sigma|$$

$$m = 11 P_1 \delta_{rb} + |P_2| \sigma_2 + P_4 k_i \delta_{rb}$$

$$> P_1 |e_\delta| + |P_2| \sigma + \max(P_4 d)$$

$$\geq \max(-P_1 e_\delta - P_1 \sigma + P_4 d)$$

Then (32) becomes

$$\dot{V} \leq m |\sigma| - 2m |\sigma| + P_2 e_\delta \dot{\sigma}$$

$$= -m |\sigma| + P_2 e_\delta \dot{\sigma}$$

A sufficient condition for $\dot{V} \leq 0$ is

$$|\sigma| > \frac{\max(P_2 e_\delta \dot{\sigma})}{m} \quad (35)$$

Since $|\sigma| > \sigma_2$, (35) can be satisfied by letting

$$\begin{aligned} \sigma_1 &= \frac{\max(P_2 e_\delta \dot{\sigma})}{m} \\ &= 11 \delta_{rb} |P_2| \frac{\max(\dot{\sigma})}{kd} \quad (\text{from (17)}) \\ &= 11 \delta_{rb} |P_2| \frac{\max(d - \frac{2}{P_4} kd \operatorname{sign}(\sigma))}{kd} \\ &= 11 \delta_{rb} |P_2| \left(\max\left(\frac{d}{kd}\right) + \frac{2}{P_4} \right) \\ &= 11 \delta_{rb} |P_2| \left(\frac{k_i \delta_{rb}}{kd} + \frac{2}{P_4} \right) \quad (\text{from (13)}) \end{aligned}$$

which gives condition (34). In conclusion, conditions (33) and (34) ensure that $\dot{V} \leq 0$.

Theorem 1:

For the dynamic model of (1), if the fuzzy control law is chosen as (18) and (19), and we firstly need to select ki, kp, Q to satisfy (23) and the parameters kd, σ_1 and σ_2 are chosen as in (27), (28) and (34). Then, both SMC and PI sub-system control system is stable in the sense lyapunov approach. The convergence of the tracking error which can still guarantee and the chattering problem of control is eliminated.

4 Simulations examples

In this section, we apply our proposed controller for two examples. The first example is a regulation problem to let the output of a duffing forced-oscillation system. The second example is to let the inverted pendulum to track a desired trajectory.

Example 1: In this example, we verify the validity of the proposed approach on the following system [16]:

$$\begin{aligned} \dot{x}^{(n)} &= f_1(\underline{x}, t) + g_1(\underline{x}, t)u_1 + d(t) \\ y_1 &= x_1 \end{aligned}$$

where

$$\begin{aligned} \underline{x} &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \\ f_1(\underline{x}, t) &= \begin{bmatrix} -0.1x_2 - x_2^3 + 12 \cos(t) \\ 0 \end{bmatrix}, \\ g_1(\underline{x}, t) &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}, d(t) = \begin{bmatrix} 0 \\ d_2 \end{bmatrix}, \\ d_2 &= 0.1 \text{rand}(1, 20) \end{aligned}$$

The control objective is to maintain the system to track the desired trajectory y_{1r} with $y_{1r}(t) = 2 \sin(10^{-2}t)$. Consider the fuzzy control law as described in (18) and (19). Let $k^1 i = 2, k^1 p = 5$.

and $Q_1 = \begin{bmatrix} 2 & 0 \\ 0 & 200 \end{bmatrix}$ from (23) we have $P1 = 12.7$, $P2 = 1$ and $P4 = 20.4$. The magnitude of $\max(d_2) = 0.1$.

Also from (18) and (35), $\delta^1_{rb} = 0.05$. Besides, let $\sigma_1^2 = 0.6 > 11\delta^1_{rb}$ which satisfies (28). Then from (33) and (34), $kd^1 = 7.7916$ and $\sigma_1^1 = 0.06$. Define $\underline{c}^1 = [1 \ 0.1]$.

The initial state $x_1(1) = 1$ and $x_2(1) = 0$.

From Fig.5, it can be seen that the using of fuzzy sliding mode PI controller overcomes the chattering problem of control and is

eliminated. However, as shown in Fig.3, chattering exists of traditional SMC.

It can be seen from Fig. 4, that a good tracking is obtained; the stability convergence and robustness are guaranteed.

Example 2: we test the validity of our approach on the inverted pendulum. Let $x_1 = \theta$ be the angle of the pendulum with respect to the vertical line and $x_2 = \dot{\theta}$. The dynamic equations of such system are given by [17].

$$\begin{aligned} \dot{x}^{(n)} &= f_2(\underline{x}, t) + g_2(\underline{x}, t)u_2 + w(t) \\ y_2 &= x_1 \end{aligned}$$

Where

$$\begin{aligned} \underline{x} &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \\ f_2(\underline{x}, t) &= \begin{bmatrix} \frac{9.8 \sin(x_1) - 0.05 x_2^2 \sin(2x_1)}{2/3 - 0.1 \cos^2(x_1)} \\ 0 \end{bmatrix}, \\ g_2(\underline{x}, t) &= \begin{bmatrix} -0.1 \cos(x_1) \\ \frac{0}{2/3 - 0.1 \cos^2(x_1)} \end{bmatrix}, \\ w(t) &= \begin{bmatrix} 0 \\ w_2 = 0.2 \text{rand}(1, 200) \end{bmatrix}. \end{aligned}$$

The control objective is to maintain the system to track the desired angle trajectory $y_{2r} = 0.5 \sin(0.015t)$. The system parameters Choose the sliding surface as: $s = c_1 e + \dot{e}$, $c_1 = 0.3$. The initial values of $x_1(1) = 1$ and $x_2(1) = 0$.

Let $k^2 i = 2, k^2 p = 3$.

$$\text{Choose } Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 50 \end{bmatrix}$$

$$\text{hence } p = \begin{bmatrix} 10.166 & 1 \\ 1 & 17.393 \end{bmatrix}, \delta^2_{rb} = 0.1.$$

$$\sigma_2^2 = 1.1 > 11\delta^2_{rb}, \sigma_2^1 = 0.173$$

$$kd^2 = 5.918$$

Fig. 6 and Fig. 8 show the simulation results for conventional SMC and proposed controller, respectively. It can be see that, white the conventional SMC the tracking is obtained (Fig. 7) but the chattering is very important. However, from Fig.8, and Fig.9 chattering disappeared and a high performance is achieved.

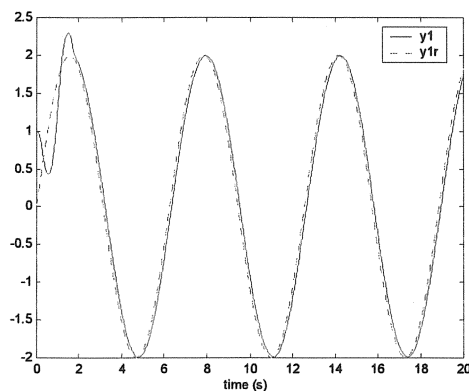


Fig.2 The behaviour of $y(t)$ and $y_r(t)$ with traditional SMC

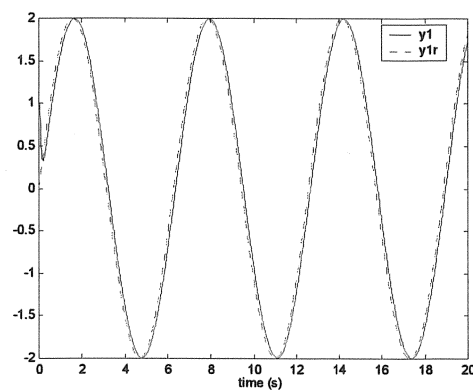


Fig.4 the behaviour of $y(t)$ and $y_r(t)$ with fuzzy SM and PI control

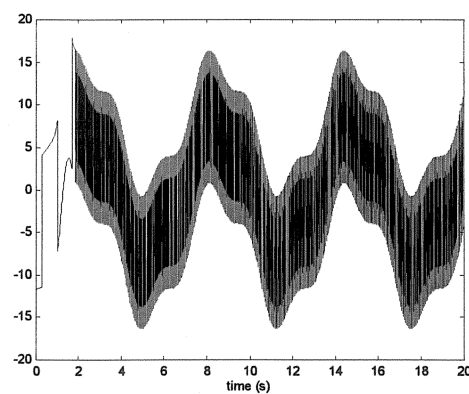


Fig.3 Sliding mode control signal u_1

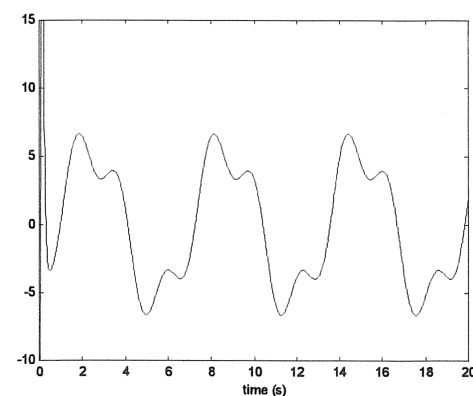


Fig.5 Fuzzy SM and PI control signal u_1

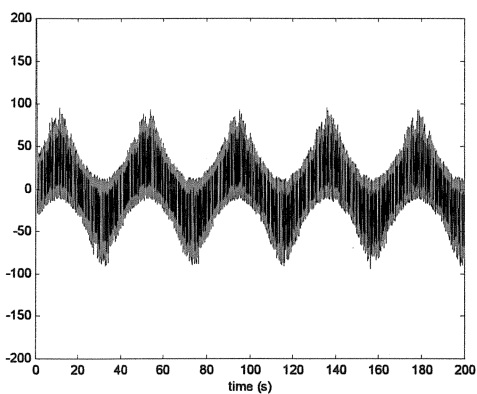


Fig.6 sliding mode control u_2

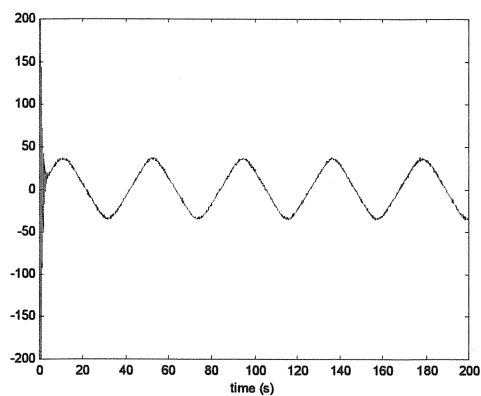


Fig.8 Fuzzy SM and PI control signal u_2

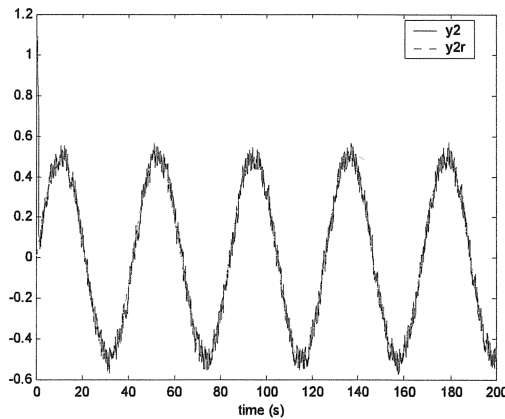


Fig.7 The behaviour of $y_2(t)$ and $y_{2r}(t)$ with traditional SMC

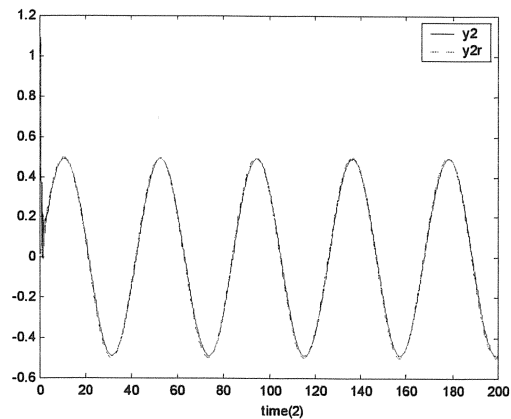


Fig.9 The behaviour of $y_2(t)$ and $y_{2r}(t)$ with fuzzy SM and PI control.

5 Conclusion

A combined a sliding mode and PI controller using a fuzzy logic is proposed to control for nonlinear systems with external disturbances. The uses of the conventional SMC technique introduce automatically the chattering phenomenon. The proposed approach is combined a SMC and a PI controller overcomes the chattering problem and high performance can be achieved. The controller does not need the accurate system mathematical model, so it is relatively easy to design. The simulation results verify the validity of the proposed fuzzy logic controller and can be deal with a large class of nonlinear systems.

References:

- [1] A. Isidori, *Nonlinear Control Systems*, 2nd ed. Berlin, Germany: Springer-Verlag, 1989.
- [2] J.E. Slotine, W.P. Li, *Applied Nonlinear Control*, Prentice Hall, 1991.
- [3] V.I. Utkin, Variable structure systems with sliding modes: A survey, *IEEE Trans. Automatic Contr*, vol.22, 1977, pp.212-222.
- [4] Edwards and S. K. Spurgeon, *Sliding Mode Control—Theory and Applications*, London, U.K.: Taylor & Francis, 1998.
- [5] H.Lee, E.Kim, H.J.Kang and M.Park, A new sliding-mode control with fuzzy boundary layer, *Fuzzy Sets and Systems*, vol. 120, no 1, May, 2001, pp. 135-143.
- [6] Efe M. O., O. Kaynak and B. M. Wilamowski, Stable Training of Computationally Intelligent Systems By Using Variable Structure Systems Technique, *IEEE Trans. on Industrial Electronics*, v.47, no.2, 2000, pp.487-496.
- [7] F.Qiao, M.Zhu, A.Winfield and C.melhuish, Fuzzy sliding mode control for discrete nonlinear systems, *transactions of china automation society*, Vol. 22, No. 2 (sum No.86), June 2003.
- [8] Wang, J.Rad A.B. and Chan, P. T., Indirect adaptive fuzzy sliding mode control: Part I: fuzzy switching, *Fuzzy Sets and Systems*. 122, 2001, pp. 21-30.
- [9] J. J. Slotine, Sliding controller design for nonlinear systems, *International Journal of Control*, 40(2), 1989, pp. 421-434.
- [10] A. Levant, Higher-order sliding modes, differentiation and output-feedback control, *International Journal of Control*, 76(9/10), 2003, pp. 924-941.
- [11] A. Levant, Universal SISO controllers with finite time convergence, *IEEE Transactions on Automatic Control*, 46(9) 2001, pp. 1447-1451.
- [12] K. Neokleous, C. Neoclaus, C. Schiras, A comparison of classical, neural and fuzzy control for an underwater vehicle, *Proceedings of the 7th WSEAS International Conference on Neural Network, cavtat, Croatia*, June 12-14, 2006, pp. 61-66.
- [13] A. Ghaffari, M. N. Bahrani, H. Porsa, Drum type fossil fuelled power plant control based on fuzzy inverse MIMO model, *Proceedings of the 7th WSEAS International Conference on Fuzzy Systems, cavtat, Croatia*, June 12-14, 2006, pp. 26-32.
- [14] C. Chih-chiang, C. Shih-hsiang, Adaptive sliding mode controller design based on TS fuzzy system models, *Automatica* vol. 42, 2006, pp. 1005-1010.
- [15] G. Haijun, Z. Tianping, S. Qikun, Decentralised model reference adaptive sliding mode control based on fuzzy model, *Journal of Systems Engineering and Electronics*, vol.17, No. 1, 2006, pp. 182-186.
- [16] L. X. Wang, *Adaptive fuzzy systems and control: designing and stability analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1994.
- [17] L. X. Wang, *A course in fuzzy systems and control*, Prentice Hall, 1997.

Annexe B

***Fuzzy Clustering for Identification of Takagi-Sugeno Fuzzy Models
of a Class of Nonlinear Multivariable Systems***

I. Lagrat, H. Ouakka et I. Boumhidi

***Conférence Système Electronique - Informatique &
Traitement de l'Information SEITI'07, pp.210-215,
Mohammedia, 25 et 26 janvier 2007.***

FUZZY CLUSTERING FOR IDENTIFICATION OF TAKAGI-SUGENO FUZZY MODELS OF A CLASS OF NONLINEAR MULTIVARIABLE SYSTEMS

I. Lagrat, H. Ouakka and I. Boumhidi

LESSI, Faculté Des Sciences, B. P 1796 Atlas, 30000 Fez- MOROCCO

E-mail : i.lagrat@caramail.com

Abstract: This paper presents the fuzzy identification of nonlinear and multivariable systems. The MIMO systems are decomposed into MISO subsystems. By this decomposition, the MISO systems are represented by compact Takagi-Sugeno fuzzy models. Thus, this approximation permits to convert a nonlinear problem into a linear one. The fuzzy identification method based on fuzzy clustering. In fact, this approach utilised to determine Takagi-Sugeno fuzzy rules and its antecedent parameters. After determination of the consequent parameters, these are adapted by a recursive least squares algorithm. The validity and the performance of the proposed fuzzy identification are tested numerically.

Key-words: Identification, Fuzzy model, Takagi-Sugeno, Fuzzy clustering, Nonlinear system, MIMO systems.

1. INTRODUCTION

The identification of nonlinear multivariable processes is an important and challenging problem. For nonlinear static and dynamic system the conventional techniques of modelling and identification are difficult to implement and sometimes impracticable. However, others techniques based on fuzzy logic are more can be used for modelling the complex nonlinear processes [1] [2]. Fuzzy modelling and identification from input-output process data proved to be effective for the approximation of uncertain nonlinear dynamic system [4]. The Takagi-Sugeno model (TS) has attracted the attention of many searchers. In fact, this model consists of if-then rules with fuzzy antecedent and mathematical functions in the consequent part [6]. The antecedent fuzzy sets divide the input space into a number of fuzzy regions, while the consequent functions describe the systems behaviour in these regions. The task of fuzzy model construction is to determine both the nonlinear parameters of the membership functions and linear parameters of the local models. Possibly, human expert are able to formulate their process knowledge in fuzzy rules. Unfortunately, this usually delivers only a rough idea of the plant behaviour, as humans cannot sense all the detail and might not be able to quantitatively express the observations. Therefore, heuristic approaches, like fuzzy clustering method is applied [1] which compute nonlinear dynamic fuzzy models from input-output measurement data, e.g., local linear model tree method (LOLIMOT), tree construction algorithm or neuro-fuzzy. There is a different algorithm to construct the fuzzy clusters such-as: the C-means algorithm, the Gath-Geva algorithm [3] and Gustafson-kessel [5] which will be used in our contribution.

In our paper, we present a fuzzy identification method based on fuzzy clustering [1] who allows automatic generation of TS models. This approach has been applied to simulation model of MIMO nonlinear process [7].

The paper is organized as follows: in section 2, the applied TS fuzzy model is presented. Section 3 describes fuzzy identification a MIMO nonlinear system method based a fuzzy clustering. In section 4, we present the application example. Conclusions are given in section 5.

2. Takagi-Sugeno fuzzy model of a MIMO process

We consider a MIMO system with n_i inputs and n_o outputs. This system will be approximated by a collection of coupled MISO discrete-time fuzzy models. The MISO models are input-output NARX (Nonlinear Auto regressive with exogenous input) defined by:

$$y_l(k+1) = f_l(x_l(k)), \quad l=1,2,...,n_o \quad (1)$$

With the regression vector represented by:

$$x_i(k) = \left[\{y_1(k)\}_{n_y}^{n_{y1}}, \dots, \{y_{n_y}(k)\}_{n_y}^{n_{yn_y}}, \{u_1(k)\}_{n_u}^{n_{u1}}, \dots, \{u_{n_u}(k)\}_{n_u}^{n_{un_u}} \right] \quad (2)$$

Here, k denotes the discrete time, and n_y and n_u define the number of delayed outputs and inputs respectively. n_d is the number of pure delays. n_y is $n_0 \times n_0$ matrix and n_u , n_d are $n_0 \times n_i$ matrices. The MISO models are estimated independently.

Through this contribution, the unknown function $f_i(\cdot)$ is approximated by a TS fuzzy model which is charities by rule consequents that are linear function of the input variables [6]. The rule base comprises M_i rules of the form:

R_i : if $x_{i1}(k)$ is Ω_{i1} and....and $x_{ip}(k)$ is Ω_{ip} then

$$y_{li}(k+1) = A_i y(k) + B_i u(k) + C_{li} \quad , \quad i=1,2,\dots,M_i \quad (3)$$

Where M_i is the number of rules in i th model. Here Ω_{li} are the antecedent fuzzy sets of i th rule,

A_i and B_i are vectors of polynomials in q^{-1} e.g.,

$$A_i = a_{i0} + a_{i1}q^{-1} + \dots + a_{iAl}q^{-nAl}, \quad B_i = b_{i0} + b_{i1}q^{-1} + \dots + b_{iBl}q^{-nBl} \quad (4)$$

With C_{li} is the offset.

The MIMO TS rules are estimated from input-output system data [1].

$$\text{When} \quad p = \sum_{j=1}^{n_o} n_{y_j} + \sum_{j=1}^{n_i} n_{u_j} + 1 \quad (5)$$

The output of TS model is computed:

$$y_l(k+1) = \frac{\sum_{i=1}^{M_i} \mu_{li}(x_l(k)) \cdot y_{li}(k+1)}{\sum_{i=1}^{M_i} \mu_{li}(x_l(k))} \quad (6)$$

Or

$$y_l(k+1) = \sum_{i=1}^{M_i} y_{li}(k+1) \cdot \Phi_{li}(x_l(k), v_{li}, \sigma_{li}) \quad (7)$$

Where Φ_{li} denote the normalized validity function such that $\sum_{i=1}^{M_i} \Phi_{li}(\cdot) = 1$ for all premise input $x_l(\cdot)$.

This normalization is achieved by:

$$\Phi_{li}(x_l(k), v_{li}, \sigma_{li}) = \frac{\mu_{li}(x_l(k))}{\sum_{j=1}^{M_i} \mu_{lj}(x_l(k))} \quad (8)$$

The membership functions μ_{li} are chosen as Gaussian with centres v_{li} and standard deviation σ_{li} .

$$\mu_{lj}(x_l(k)) = \exp\left(-\frac{1}{2} \left(\frac{x_{l1} - v_{lj1}}{\sigma_{lj1}}\right)^2\right) \dots \exp\left(-\frac{1}{2} \left(\frac{x_{lp} - v_{ljp}}{\sigma_{ljp}}\right)^2\right) \quad (9)$$

μ_{lj} is the degree of fulfilment of the rule j . once structure is fixed, the n_o MISO parameters are estimated independently by fuzzy clustering [3].

3. Fuzzy identification

The TS fuzzy model is suitable to model a large class of nonlinear systems [1] [6]. Fuzzy modelling and identification from measured data are effective tools for approximation of uncertain nonlinear systems. The identification of TS model is usually done two steps. In the first steps, off-line identification is to determine both the nonlinear parameters of the Gaussian (v_{li} and σ_{li}) membership function and linear parameters of the local models (A_{li}, B_{li}, C_{li}) are determined by fuzzy clustering

algorithm [1]. In the second steps, on-line of fuzzy model the known, the linear parameters in the consequents are locally adapted by a recursive least-squares algorithm.

3.1 Off-line identification of the fuzzy model

The previous section has showed how the consequent part of TS models can be identified by weighted least-squares method when the antecedent membership functions are given. In system identification, the Gustafson-Kessel algorithm is applied to identify TS model [3]. The available data samples are collected in matrix Z , formed by concatenating the regression matrix and the output vector. The clustering obtains the fuzzy partition of the Z data. The fuzzy partition is presented by the $U_l = [u_{lij}]_{nc \times n}$ matrix where $u_{lij} \in [0,1]$ element of the matrix represents the degree of membership of the observation in cluster i, nc and n are the number of clusters and data samples respectively. A prototype matrix $V_{li} = [v_{li1}, \dots, v_{lipi}]$ and set of clusters covariance matrices $F_{li} = [F_{li1}, \dots, F_{lipi}]$. Once triplet (U, V, F) is determined the parameters of the rule's premises (v_{li} and σ_{li}) and the consequents parameters (A_{li}, B_{li}, C_{li}) are computed. For more detail see [1]. After, the antecedent membership functions from the cluster parameters are determined. The Gaussian functions are used to represent the fuzzy sets Ω_{lij} :

$$\Omega_{lij}(x_{li}(k)) = \exp\left(-\frac{1}{2}\left(\frac{(x_{li} - v_{lij})^2}{\sigma_{lij}^2}\right)\right) \quad (10)$$

The consequents parameters as $\theta_{li} = [A_{li}, B_{li}, C_{li}]$ in each rule are estimated separately by weighted least-squares method by minimizing the following criterion [1].

$$\min_{\theta_{li}} \frac{1}{N} [y_l - \xi_l \theta_{li}]^T Q_{li} [y_l - \xi_l \theta_{li}] \quad (11)$$

When $\xi_l = [x_l \ 1]$ is the regression matrix extended by a unitary column and Q_{li} is a matrix containing the values of validity functions Φ_{li} of the i th local model for each data samples. The weighting matrix:

$$Q_{li} = \text{diag}(\Phi_{li}(x_1(1), v_{li}, \sigma_{li}), \dots, \Phi_{li}(x_1(N), v_{li}, \sigma_{li})) \quad (12)$$

by using the weighted least-squares solution of θ_{li} given by:

$$\theta_{li} = [\xi_l^T Q_{li} \xi_l]^{-1} \xi_l^T Q_{li} y_l \quad (13)$$

3.2 On-line adaptation of fuzzy model

In the on-line phase the rule premise are kept fixed on only the rule consequents are adapted. The following recursive weighted least-squares (RWLS) algorithm with forgetting factor is utilized to estimate the parameters of each local linear model. For j th local linear model its computes a new parameters estimate $\theta_{lj}(k)$ in (14) at the time instant k as follows:

$$\theta_{lj}(k) = \theta_{lj}(k-1) + \delta_{lj}(k) (y_l(k) - \xi_l^T(k) \theta_{lj}(k-1)) \quad (14)$$

$$\delta_{lj}(k) = \frac{P_{lj}(k-1) \xi_l(k)}{\xi_l^T(k) P_{lj}(k-1) \xi_l(k) + \frac{\lambda}{\Phi_{lj}(x_l(k), v_{lj}, \sigma_{lj})}} \quad (15)$$

$$P_{lj}(k) = \frac{1}{\lambda} [I - \delta_{lj}(k) \xi_l^T(k)] P_{lj}(k-1). \quad (16)$$

λ is a forgetting factor that implements forgetting of the old measurements and $\Phi_{lj}(\cdot)$ is weighting of actual data with the validity function value. P_{lj} is a matrix of adaptation gain.

4. A simulations examples

Consider a MIMO system [17] having a polynomial nonlinearity described as:

$$A(q^{-1})y(t) = B_d(q^{-1})Z(t) + \xi(t) \quad (17)$$

Where q^{-1} is the back shift operator.

$$A(q^{-1}) = \text{diag}[A_i(q^{-1})], A_i(q^{-1}) = \sum_{r=0}^{na(i)} a_{ir} q^{-r}, a_{i0} = 1, B_d(q^{-1}) = [q^{-d_{ij}} B_{ij}(q^{-1})], B_{ij}(q^{-1}) = \sum_{r=0}^{nb(i,j)} b_{ijr} q^{-r}, b_{ij0} \neq 0 \quad (18)$$

d_{ii} is time delay between the input $u_i(t)$ and the output $y_i(t)$. d_{ij} is time delay between the input $u_j(t)$ and the output $y_i(t)$. $y(t) \in R^n$, $Z(t) \in R^n$ and $\xi(t) \in R^n$ are respectively the output, nonlinear input and the disturbance vector. As in [8] the non-linearity is assumed to be represented as follows:

$$Z_i(t) = f_{i0} + f_{i1}u_i(t) + \dots + f_{ip}u_i^{p_i-1}(t) \quad (19)$$

The MIMO system (17) can be decomposed into n multi-input, single-output (MISO) systems. The fuzzy identification, decoupling and control will be conducted at the level of each subsystem. Thus, the computational load is reduced.

$$A(q^{-1})y_i(t) = q^{-d_{ii}}B_{ii}(q^{-1})Z_i(t) + \sum_{\substack{j=1 \\ i \neq j}}^n q^{-d_{ij}}B_{ij}(q^{-1})Z_j(t) + \xi_i(t) \quad (20)$$

The multivariable system has tested in simulation represented by:

$$A_1(q^{-1})y_1(t) = q^{-d_{11}}B_{11}(q^{-1})Z_1(t) + q^{-d_{12}}B_{12}(q^{-1})Z_2(t).$$

$$A_2(q^{-1})y_2(t) = q^{-d_{22}}B_{22}(q^{-1})Z_2(t) + q^{-d_{21}}B_{21}(q^{-1})Z_1(t)$$

With

$$A = [A_1 \quad A_2]^T = \begin{bmatrix} 0.1q^{-1} \\ 0.2q^{-1} \end{bmatrix}, B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} 1 + 0.5q^{-1} & 0.05q^{-1} \\ 0.3q^{-1} & 1 + 0.8q^{-1} \end{bmatrix}$$

$$d_{11} = 1, d_{12} = 0, d_{21} = 0, d_{22} = 1.$$

$$Z_1(t) = 2u_1(t) + u_1^2(t).$$

$$Z_2(t) = u_2^2(t)$$

The identification procedure is carried out with random inputs $u_1(k)$ and $u_2(k)$ uniformly distributed in the interval $[-1 \ 1]$ and three clusters for each output. Once the input-output data is available, we compute the matrix (U, V and F) by Gustafson-Kessel algorithm then we determine the consequent parameters of each rule of TS fuzzy model are linear conclusions of system's inputs.

The Responses of the plant and identification model are shown in Figure 1 and 2. To valid the established model, we apply the input signal are square waves with period 15.

These responses are obtained from local models computed by fuzzy clustering without adaptation of the consequent parameters is shown in Figure 3 and 4. It is noticed that the estimated outputs can't follow the process's output and the error resultant is rather big. To improve quality of the established fuzzy model, the parameters of the consequences of the rules are adapted by a recursive least squares algorithm with forgetting factor ($\lambda = 0.99$) according to (14).

Figure.5 and 6 shows the responses of the plant and the identification model with adaptation.

We define a function VAF which computes the percentile variance accounted for between two

Signals as follows:

$$VAF = 100\% \left[1 - \frac{\text{var}(y - y_m)}{\text{var}(y)} \right]$$

y is the output of the process and y_m is the output of the model. The VAF of two equal signals is 100%. If the signals differ, VAF is lower. Table 1 gives the VAF performance indices for the responses of the plant and the identification model in the identification phase, the validation phase without adaptation and in the validation phase with adaptation. We can see that the adapted fuzzy TS model is more accurate than a non-adapted fuzzy TS model.

The rules are linear conclusions of system's inputs, for example for the rule i :

$$y_1(k) = a_{11}^1(k)y_1(k-1) + b_{11}^1(k)u_1(k-1) + b_{12}^1(k)u_1(k-2) + b_{22}^1(k)u_2(k-1) + C_{11}^1(k)$$

$$y_2(k) = a_{21}^2(k)y_2(k-1) + b_{21}^2(k)u_1(k-1) + b_{12}^2(k)u_2(k-1) + b_{22}^2(k)u_2(k-2) + C_{21}^2(k)$$

Figure.7 and 8 shows the evolution of parameters of the output y_1 and y_2 for three clusters.

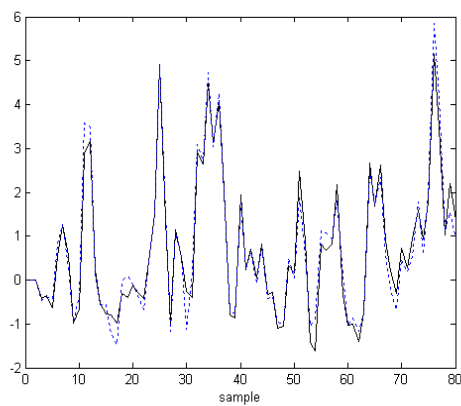


Figure.1 comparison of the process output (solid line) y_1 with fuzzy model output (dashed dotted line) for identification.

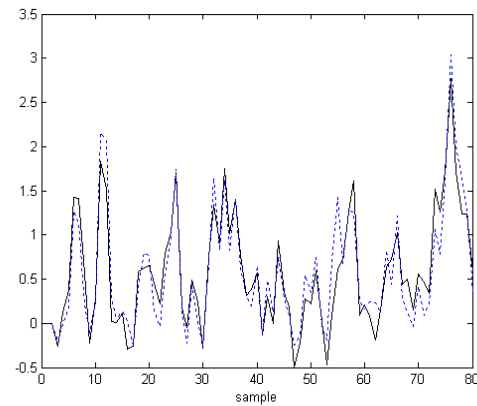


Figure.2 comparison of the process output (solid line) y_2 with fuzzy model output (dashed dotted line) for identification.

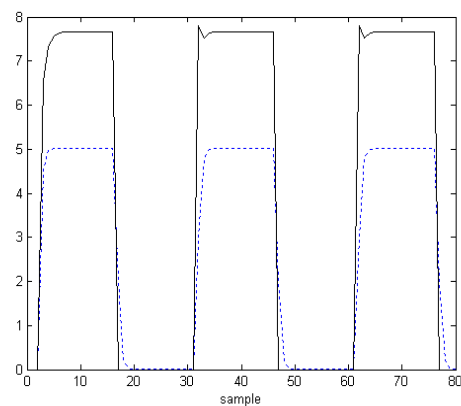


Figure.3 comparison of the process output (dashed dotted line) y_1 with fuzzy model output (solid line) for validation.

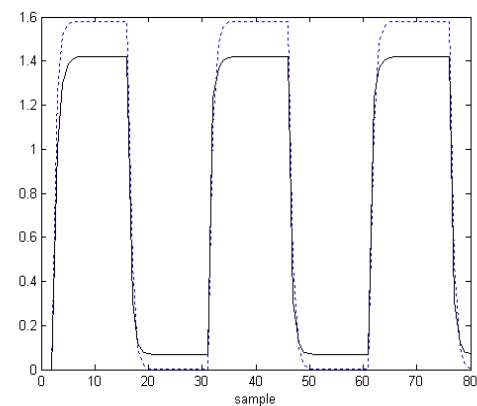


Figure.4 comparison of the process output (dashed dotted line) y_2 with fuzzy model output (solid line) for validation.

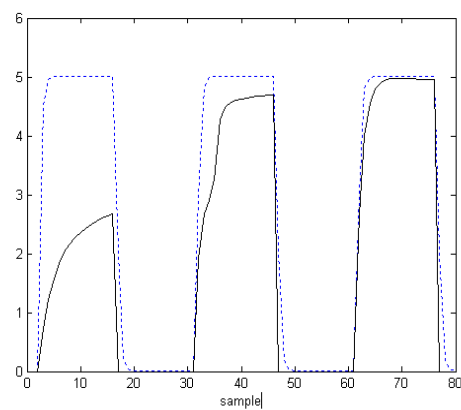


Figure.5 comparison of the process output (dashed dotted line) y_1 And the fuzzy model output (solid line) with parameters s adaptation .

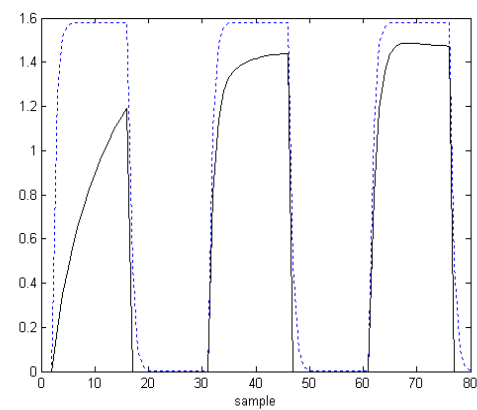
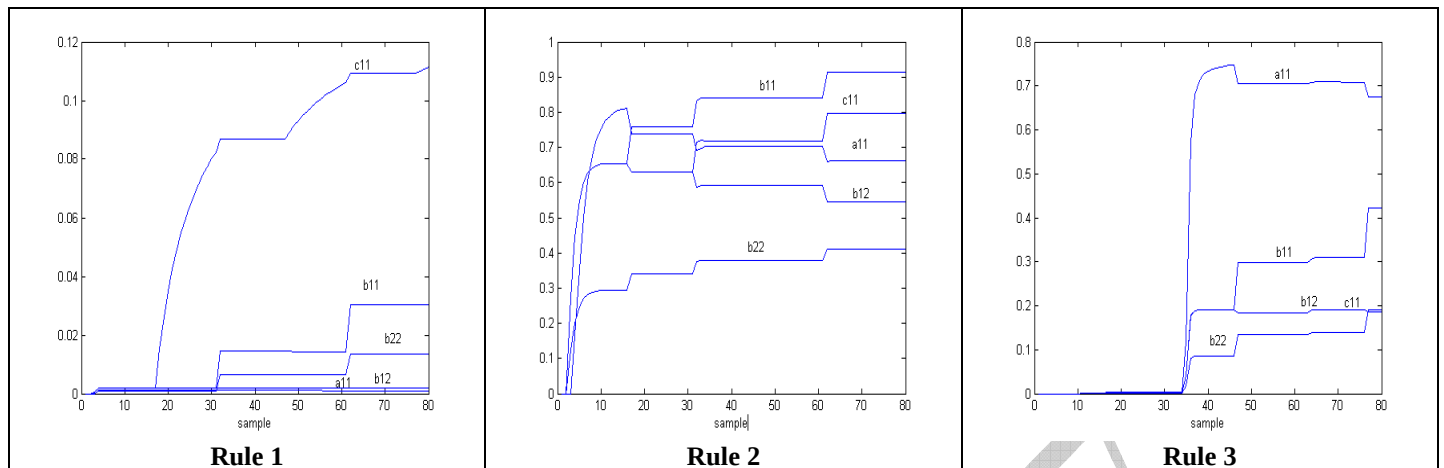
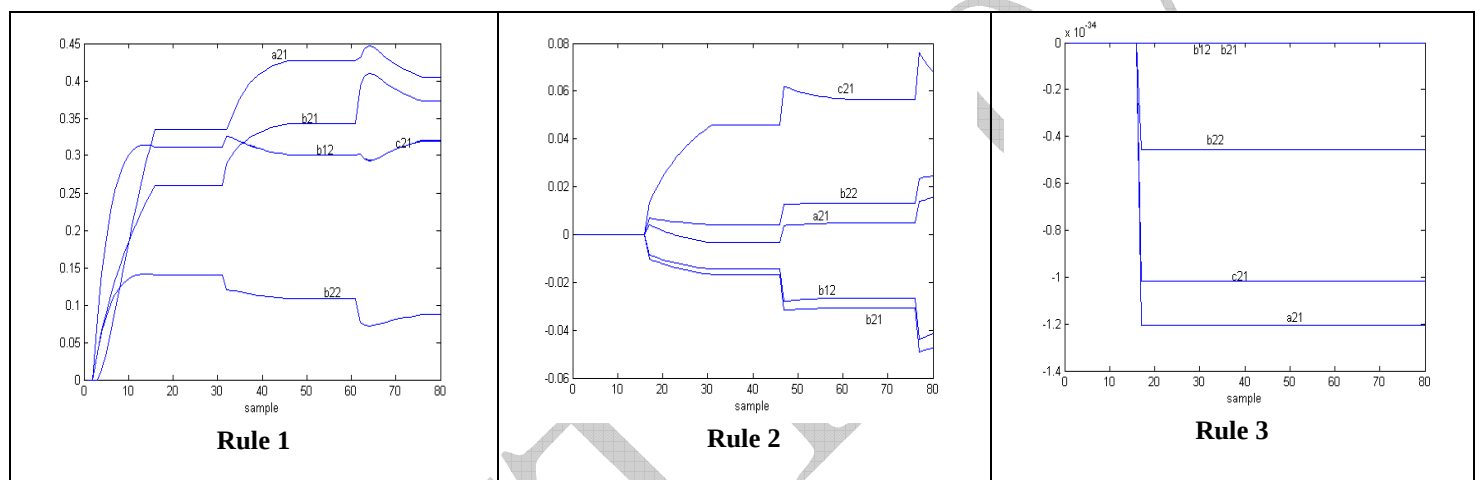


Figure.6 comparison of the process output (dashed dotted line) y_2 And the fuzzy model output (solid line) with parameters s adaptation .

	Identification	Validation without adaptation	Validation with adaptation
VAF- y_1 (%)	92.67	58.79	78.17
VAF- y_2 (%)	97.85	67.68	85.79

Table.1 Comparison of the prediction accuracy of TS fuzzy model in three phases.

Figure.7 Evolution of parameters of output $y_1(t)$ Figure.8 Evolution of parameters of output $y_2(t)$

5. Conclusions

In this paper, we are interesting on the identification of MIMO TS fuzzy models. The local models and the antecedent part of the fuzzy model are determined by fuzzy clustering. Fuzzy modelling and on-line adaptation of the fuzzy model have been successful applied to obtain compact and accurate models for MIMO processes. This approach is demonstrated by means of the identification of MIMO systems having input nonlinearities and able to describe the process in large operating points to be use in schema of adaptive control this will be the object of our forth coming work.

References:

- [1] R. Babuska. Fuzzy Modeling for Control. Kluwer Academic Publishers, Boston, 1998.
- [2] I.Lagrat, H. Ouakka, I.Boumhidi, Fuzzy Sliding Mode PI Controller for Nonlinear Systems, WSEAS Transaction on Signal Processing vol. 2 September 2006, pp 1137-1144.
- [3] I. Gath, A. B. Geva, unsupervised optimal fuzzy, IEEE Transactions on Pattern Analysis and Machine Intelligent vol. 7, pp 773-781, 1989.
- [5] D. E. Gustafson, V. C. Kessel, Fuzzy Clustering, with fuzzy covariance matrix, In Proceeding IEEE. CDC, San Diego, pp 761-766, 1979.
- [4] A. Trabelsi, F. Lafont, M. Kamoun, G. Ensa, Identification of Nonlinear Multivariable Systems by Adaptive Fuzzy Takagi-Sugeno Model, IJCC, vol. 2 (3), pp 137-153, 2004.
- [6] T. Takagi, M. Sugeno, Fuzzy Identification of Systems and its Application to Modeling and Control, IEEE Transactions on Systems, Man and Cybernetics, vol. 15 (1), 116-132, 1985.
- [7] F. Weng and Lang S, Globally convergent direct adaptive control algorithm fore multivariable systems with general time delay structure. International Journal of Control, vol 51 (2), pp 301-314, 1990.
- [8] Z Jinxing and Shijium L. Explicit-self-tuning control for a class of nonlinear systems, vol. 25(4), 593-596 1989.

Annexe C

***Adaptive Control of a Class of Nonlinear Systems Based on
Takagi-Sugeno Fuzzy Model***

I. Lagrat, H. Ouakka et I. Boumhidi

***Information and Communication Technologies International
Symposium, ICTIS'07, pp.594-597,
Fez-Morocco, 3-5 April, 2007.***

Adaptive control of a class of nonlinear systems based on Takagi-Sugeno fuzzy model

I. Lagrat, H. Ouakka, and I. Boumhidi

Abstract—This paper presents an adaptive control for a class of nonlinear systems. The nonlinear system is represented by compact Takagi-Sugeno fuzzy models. Thus, this approximation permits to convert a nonlinear problem into a linear one. Hence, a method of designing a controller based some conventional adaptive control technique. The validity and the performance of the proposed control algorithm are tested on the chaotic Henon model.

Index Terms—Fuzzy logic, Takagi-Sugeno model, Adaptive control, Nonlinear system.

I. INTRODUCTION

The control of nonlinear systems has been an important research topic and many approaches have been proposed [1], the identification of nonlinear processes is an important and challenging problem. Recently, the Takagi-Sugeno model (T-S) fuzzy approach [4], [8] has been used to model nonlinear systems.

In fact, this model consists of if-then rules with fuzzy antecedent and mathematical functions in the consequent part [3]. The antecedent fuzzy sets divide the input space into a number of fuzzy regions, while the consequent functions describe the systems behaviour in these regions. Using this technique, the dynamic of nonlinear system is easily obtained by linearization near different operation points by input-output identification around these points. Once these linear models obtained, local linear controller can be designed, the overall controller would be a fuzzy blending of these controller.

I. Lagrat Laboratoire d'Electronique, Signaux-Systèmes et d'Informatique. Faculté Des Sciences Dhar El Mehraz, B P 1796 Atlas 30000 Fez .MOROC. i.lagrat@caramail.com.

H. Ouakka Laboratoire d'Electronique, Signaux-Systèmes et d'Informatique. Faculté Des Sciences Dhar El Mehraz, B P 1796 Atlas 30000 Fez .MOROC.

I. Boumhidi Laboratoire d'Electronique, Signaux-Systèmes et d'Informatique. Faculté Des Sciences Dhar El Mehraz, B P 1796 Atlas 30000 Fez .MOROC.

This contribution deals with nonlinear adaptive controllers based on local linear fuzzy models. Nonlinear dynamic systems are described by means of T-S fuzzy models. The output of these fuzzy systems is calculated as an interpolation of locally valid linear models. On the one hand, this allows a linguistic interpolation of the fuzzy rules. On the other hand, classical linear control concepts can be applied to the local linear models [4]. The process information from the fuzzy models can be utilised by a great variety of control methodologies [2], [5].

The paper is organized as follows: section II formulates the T-S fuzzy model of a class of nonlinear systems, and section III presents the fuzzy adaptive control design. In section IV, simulation results are given of Henon chaos model.

II. T-S FUZZY MODEL OF NONLINEAR SYSTEM

We consider a class of nonlinear systems defined by:

$$y(t+1) = f(x(t)) \quad (1)$$

With the regression vector represented by:

$$x(t) = [y(t), \dots, y(t-n), u(t), \dots, u(t-n)] \quad (2)$$

Here, t denotes the discrete time, and n define the number of delayed output. Through this contribution, the unknown function $f(\cdot)$ is approximated by a T-S fuzzy model which is charities by rule consequents that are linear function of the input variables [3]. The rule base comprises m rules of the form:

R^i : if x_1 is F_1^i and x_s is F_s^i

Then

$$y(t+1) = a_{i1}y(t) + \dots + a_{in}y(t-n) + b_{i1}u(t) + \dots + b_{in}u(t-n) \quad (3)$$

$$i = 1, 2, \dots, m$$

Where R^i denotes the i th fuzzy inference rule, m is the number of inference rules, $F_j^i (j=1, 2, \dots, s)$ are fuzzy sets, $u(t)$ is the system input variable, $y(t)$ is the system output, $a_{i1}, \dots, a_{in}, b_{i1}, \dots, b_{in}$, are coefficient of the i th subsystem, and $x(t) = [x_1, \dots, x_s]$ are some measurable system variables. Let $\mu_i(x(t))$ be the normalized

membership function of the inferred fuzzy set F^i ,

where $F^i = \bigcap_{j=1}^s F_j^i$ and $\sum_{i=1}^m \mu_i = 1$

The output of T-S fuzzy model is computed:

$$y(t+1) = \sum_{i=1}^m \mu_i [a_{i1}y(t) + \dots + a_{in}y(t-n) + b_{i1}u(t) + \dots + b_{in}u(t-n)] \quad (4)$$

III. FUZZY ADAPTIVE CONTROL DESIGN

In this section, the objective of the fuzzy adaptive control under consideration is to find an adaptive control law such that output of the nonlinear system, with unknown parameters can track a given bounded reference signal with guaranteed stability.

We define

$$\begin{aligned} \phi_i(t-1) &= [a_{i1}y(t-1) \dots a_{in}y(t-n) \mu_i u(t-1) \dots \mu_i u(t-n) \mu_i]^T \\ \theta_i &= [a_{i1} \dots a_{in} \ b_{i1} \dots b_{in} \ c_i]^T \\ \phi(t-1) &= [\phi_1^T(t-1) \ \phi_2^T(t-1) \ \dots \ \phi_m^T(t-1)]^T \\ \theta &= [\theta_1^T \ \theta_2^T \ \dots \ \theta_m^T]^T \end{aligned}$$

The system described by (4) can also be rewritten as:

$$y(t) = \theta^T \phi(t-1) \quad (5)$$

This is a regression form, with θ being a system parameter vector and ϕ a regression vector. It should be noted that the system (5) is in general nonlinear but it is linear with respect to its unknown parameter vectors. Based on parameterization (5), the identification algorithm giving estimates $\hat{\theta}(t)$ of $\theta(t)$ can be obtained using the normalized least-squares algorithm [6].

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{p(t-1)\bar{\phi}(t-1)\bar{e}_1(t)}{\lambda(t) + \bar{\phi}^T(t-1)p(t-1)\bar{\phi}(t-1)} \quad (6)$$

$$\bar{e}_1(t) = \bar{y}(t) - \bar{\phi}^T(t-1)\theta(t-1)$$

$$p(t) = p(t-1) - \frac{p(t-1)\bar{\phi}(t)\bar{\phi}^T(t)p(t-1)}{\lambda(t) + \bar{\phi}^T(t-1)p(t-1)\bar{\phi}(t-1)}$$

$$\lambda(t) = \lambda_0 \lambda(t-1) + (1 - \lambda_0), \quad \text{with} \quad \lambda_0 = 0.95$$

$$\text{and } \lambda(0) = 0.96, \ p(0) = \varepsilon I, \ 0 < \varepsilon < \infty.$$

$$\bar{y}(t) = \frac{y(t)}{\eta(t-1)}, \ \bar{\phi}(t) = \frac{\phi(t-1)}{\eta(t-1)}, \quad \eta(t) = \max\{1, \|\phi(t)\|\},$$

$\eta(t)$ is a normalizing signal.

We will employ the certainty equivalence principle to design the adaptive controller [5]. In doing so, we first consider the control design for the case where the parameters of the system model are known a priori. Consider the following local control law:

$$R^i : \text{if } x_1 \text{ is } F_1^i \text{ and } x_s \text{ is } F_s^i$$

then

$$u_i(t) = \frac{1}{b_{i1}} [-h_i(t) + y_r(t+1) - k_1 e(t-1) - \dots - k_n e(t-n+1)] \quad (7)$$

$$i = 1, 2, \dots, m$$

where

$$\hat{h}_i(t) = a_{i1}y(t) + \dots + a_{in}y(t-n+1) + b_{i2}u(t-1) + \dots + b_{in}u(t-n+1) + c_i$$

The global control law can then be obtained as:

$$u(t) = \sum_{i=1}^m \mu_i u_i(t) = \sum_{i=1}^m \mu_i \frac{1}{b_i} [h_i(t) + y_r(t+1) - k_1 e(t) - k_2 e(t-1) - \dots - k_n e(t-n+1)] \quad (8)$$

(8)

and $\{k_i\}$ are coefficients of a stable polynomial defined by:

$$\Omega(z) = z^n + k_1 z^{n-1} + \dots + k_{n-1} z + k_n$$

This specifies the desired output tracking error dynamics.

Substituting the control law (8) into the system model (4) leads to the following closed-loop system:

$$e(t+1) + k_1 e(t) + \dots + k_n e(t-n+1) = 0 \quad (9)$$

This, in turn, can guarantee the tracking error to approach zero as time goes to infinity.

Based on the certainty equivalence principle, we now choose the following local adaptive control law:

$$R^i : \text{if } x_1 \text{ is } F_1^i \text{ and } x_s \text{ is } F_s^i$$

then

$$u_i(t) = \frac{1}{\hat{b}_i} [\hat{h}_i(t) + y_r(t+1) - k_1 e(t) - k_2 e(t-1) - \dots - k_n e(t-n+1)] \quad (10)$$

$$i = 1, 2, \dots, m$$

With

$$\hat{h}_i(t) = \hat{a}_{i1}y(t) + \dots + \hat{a}_{in}y(t-n+1) + \hat{b}_{i2}u(t-1) + \dots + \hat{b}_{in}u(t-n+1) + \hat{c}_i$$

The global control law can then be obtained as:

$$u(t) = \sum_{i=1}^m \mu_i u_i(t) = \sum_{i=1}^m \mu_i \frac{1}{\hat{b}_i} [\hat{h}_i(t) + y_r(t+1) - k_1 e(t) - k_2 e(t-1) - \dots - k_n e(t-n+1)] \quad (11)$$

Remark1. In order to guarantee the non-singularity of the control law $\hat{b}_i \neq 0$ has to be ensured. There are a number of methods that can be used to achieve this in adaptive control systems design, such as the projection method proposed in [7].

Next, substituting the control law (11) into the fuzzy dynamic model (4) leads the following closed-loop system:

$$e(t+1) + k_1 e(t) + \dots + k_n e(t-n+1) = \sum_{i=1}^m \mu_i [h_i(t) - \hat{h}_i(t)] + \sum_{i=1}^m \mu_i [b_{i1} - \hat{b}_{i1}] u(t) \quad (12)$$

(12)

By defining

$$x_e(t) = [e(t-n+1) \ e(t-n+2) \ \dots \ e(t)]^T \quad (13)$$

The above closed-loop system can be expressed in state-space form, as:

$$x_e(t+1) = Ax_e(t) + Be_1(t+1) \quad (14)$$

Where

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & & 0 \\ \vdots & \vdots & 0 & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ -k_n & -k_{n-1} & \cdots & -k_2 & -k_1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

Theorem1. For the fuzzy system (4), if the adaptive control law (11) is used, then the resulting closed-loop system is stable in sense that all the signals are bounded. Furthermore, the output tracking error will approach zeros as time goes to infinity.

IV. SIMULATION RESULTS

This section presents a simulation example to shown an application of the proposed adaptive control algorithm and its satisfactory performance. The dynamic equation of Henon model [5] is given by:

$$y(t) = -y(t-1)^2 + 0.3y(t-2) + 1.4 + u(t-1) \quad (15)$$

With the following two membership functions [5]:

$$\mu_1(y(t-1)) = 0.5(1 - \frac{y(t-1)}{d}); \quad \mu_2(y(t-1)) = 0.5(1 + \frac{y(t-1)}{d}) \quad (16)$$

Where $d = 0.7$ is a constant, large enough to cover the range of the output, the Henon model can be expressed as:

Rule 1: if $y(t-1)$ is μ_1 then

$$y(t) = dy(t-1) + 0.3y(t-2) + u(t-1) + 1.4$$

Rule 2: if $y(t-1)$ is μ_2 then

$$y(t) = -dy(t-1) + 0.3y(t-2) + u(t-1) + 1.4$$

The initial position $[y(0) \ y(-1)] = [0 \ 1]^T$

The initial parameter is chosen to be $\hat{\theta}(0) = [d \ 1 \ 1 \ 1.4 \ -d \ 1 \ 1 \ 1.4]^T$

The response of the non-adaptive control system is shown in Fig.1, where a worse performance of regulation is observed but a good is obtained in Fig.2, with the adaptive control law proposed.

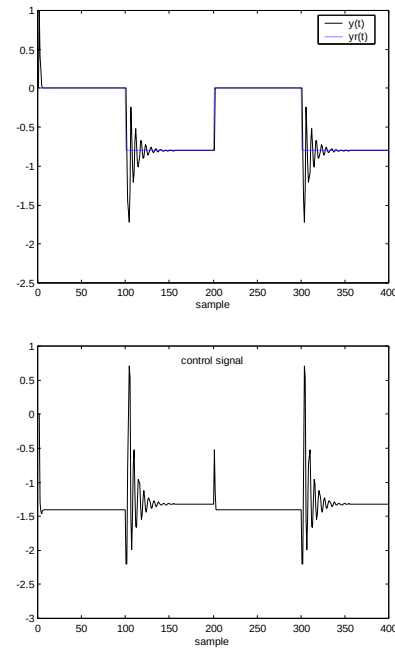


Fig.1. Responses with non adaptive control.

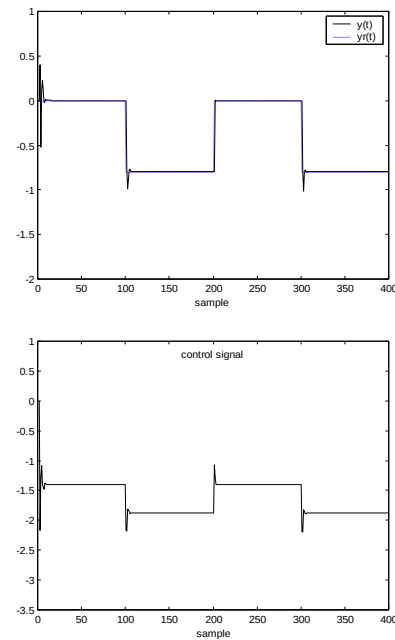


Fig.2. Responses with adaptive control.

V. CONCLUSIONS

In this paper has presented an adaptive control for a class of nonlinear systems. Firstly, the system nonlinear is represented by T-S fuzzy model. This approximation permits to convert a nonlinear problem into a linear one. A design a local linear adaptive controller in each local region, using the normalized least-squares algorithm. Then reassemble a global adaptive controller over the entire domain, that the stability of the closed-loop is guaranteed. The simulation result demonstrates the performance of the proposed control strategy.

REFERENCES

- [1] J.E. Slotine, W.P. Li, *Applied Nonlinear Control*, Prentice Hall, 1991.
- [2] I. Lagrat, H. Ouakka, I. Boumhidi, Fuzzy Sliding Mode PI Controller for Nonlinear Systems, *WSEAS Transactions on Signal Processing*, vol (2): 1137-1143, September 2006.
- [3] T. Takagi, M. Sugeno, Fuzzy Identification of Systems and its Application to Modeling and Control, *IEEE Transactions on Systems, Man and Cybernetics*, 15 (1): 116-132, 1985.
- [4] R. Babuska, *Fuzzy Modeling for Control*, Kluwer Academic Publishers, Boston, 1998.
- [5] G. Feng, G. Chen, Adaptive control of discret-time chaotic systems: a fuzzy control approach, *Chaos. Solutions and Fractals* 23, 459-467. 2005
- [6] L. Praly, Robustness of indirect adaptive control based on pole placement design. *Proc. Of the 1st IFAC Workshop on adaptive systems in control and signal processing*, San Francisco, USA.1983.
- [7] G. Feng, Analysis of new algorithm for continuous time robust adaptive control, *IEEE Trans Automat Contr* 1999; 44:1764-8.
- [8] I. Lagrat, H. Ouakka, I. Boumhidi, *Fuzzy Clustering For Identification of Takagi-Sugeno Fuzzy Models of a Class of Nonlinear Multivariable Systems*, Systems Electronique-Informatique & Traitement de l'Information Mohammédia les 25 et 26 Janvier 2007, pp: 210-215.

Annexe D

***Decentralized Adaptive Fuzzy Control
for a Class of Nonlinear Systems***

A.Errahmani, H.Ouakka, M.Benyakhlef et I.Boumhidi

***WSEAS TRANSACTIONS on SYSTEMS and CONTROL
Issue 8, Vol 2, pp.411-418, August, 2007.***

Decentralized Adaptive Fuzzy Control for a class of Nonlinear Systems

A.ERRAHMANI, H.OUAKKA, M.BENYAKHLEF AND I.BOUMHIDI

LESSI. Département de physique
Université Sidi Mohamed Ben Abdelah
Faculté des Sciences Dhar Elmahraz
BP 1796 Fez Atlas 30 000
MOROCCO

aerrahmani@menara.ma, mbenyakhlef@netcourrier.com

Abstract: - In this paper, stable direct and indirect decentralized adaptive fuzzy controls are proposed for a class of large-scale nonlinear systems with the strong interconnected. The feedback and adaptive mechanisms for each subsystem depend only upon local measurements to provide asymptotic tracking of a reference trajectory. In both approaches, the proposed controllers are used to approximate the unknown subsystems. In addition, each subsystem is able to adaptively compensate for interconnections without known bounds. Simulation results are given to illustrate the tracking performance of the proposed methods.

Keywords: Decentralized Control, Fuzzy Controller, Interconnected Nonlinear System.

1 Introduction

Decentralized adaptive control systems often arise from various complex situations where there exist physical limitations on information exchange among several systems for which there is insufficient capability to have a single control controller, and due to the physical configuration and high dimensionality of interconnected systems a centralized control is neither economically feasible nor even necessary. Therefore, the decentralized scheme is preferred in control design of large-scale interconnected system [1], [2], [9]. To control a large-scale system, one essential problem is how to handle the interactions among different systems. Intensive research has been devoted to the observer design for large-scale systems. Uncertainties in a large-scale system require the adaptive decentralized technique, for which many decentralized adaptive schemes have been developed, including the model reference adaptive control [1],[4], and nonlinear control with a special class of interconnections [7]. These approaches focus on stabilisation, where the dynamics of subsystems are assumed to be known or to be linear with a set of unknown parameters. However, in practice, large-scale systems may contain significant uncertainties, and/or with unknown parameters in nonlinear forms and unknown structures.

Fuzzy logic control as one of the most useful approaches for utilizing expert knowledge, has been an active field of research the past decade [8],[11]. Fuzzy logic control is generally applicable to plants

that are mathematically poorly modelled and where experienced operators are available for providing qualitative guidance. The most important advantage of fuzzy-logic-control schemes lies in the fact that the developed controllers can deal with increasingly complex systems and controllers without precise knowledge of the model structure of the underlying system dynamic. Recently there have been significant research efforts on these issues in fuzzy control system [8],[11],[12] but these approaches work only for large-scale systems with a known or linear dynamics with a set of unknown parameters and bounds interconnections. In practice, however, not all states are usually available.

This paper presents two approaches which can easily tackle the output tracking control problem of a class of large-scale nonlinear system with unknown interconnections bounds. A direct adaptive approach approximates unknown control laws required to stabilize each subsystem, while an indirect adaptive is provided which identifies the isolated subsystem dynamics to produce a stabilizing controller. Both approaches ensure asymptotic tracking using only local measurement.

The organization of this paper is as follows: section 2 describes the problem under investigation; section 3, the direct adaptive decentralized control; while in section 4, we introduce the indirect approach. Experimental results are then used to demonstrate the effectiveness of the proposed approaches is presented in section 5, with a conclusion given in section 6.

2 Problem Formulation

Consider a class of nonlinear interconnected SISO subsystems S_i ($i = 1, 2, \dots, N$) described as follows:

$$\begin{cases} \dot{x}_{i,1} = x_{i,2} \\ \vdots \\ \dot{x}_{i,n_i} = f_i(x_i) + g_i(x_i)u_i + \Delta_i(\underline{x}) \\ y_i = x_{i,1} \end{cases} \quad (1)$$

where $\underline{x} = [x_1^T, x_2^T, \dots, x_N^T]^T$, $x_i \in \mathbb{R}^{n_i}$, is the global state vector, $u_i(t) \in \mathbb{R}$ is the control signal input and $y_i \in \mathbb{R}$ is the output of the plant for the subsystem S_i . The functions $f_i(\cdot)$ and $g_i(\cdot)$ are unknown and nonlinear, and $\Delta_i(\underline{x}) \in \mathbb{R}$, are interconnection among subsystems unknown ($i = 1, 2, \dots, N$).

The tracking error for S_i is defined by $e_{i0} = y_{ir} - y_i$. Our objective is to design an adaptive control for each subsystem which will cause the output y_i to track a desired output trajectory y_{ir} ($i.e., e_{i0} \rightarrow 0$) in the presence of the strong interconnections using only local measurements.

Assumption 1: Let the scalars q_{ij} quantify the strength of the interconnections and the output vector for the i^{th} subsystem be defined by $e_i = [e_{i0}, \dot{e}_{i0}, \dots, e_{i0}^{(d_i-1)}]^T$; it is assumed that the interconnections satisfy:

$$|\Delta_i(\underline{x})| \leq \sum_{j=1}^N q_{ij} \|e_j\|_2 \quad (2)$$

where $\|\cdot\|_2$ is the Euclidean vector norm. This assumption on the interconnections can be satisfied by a variety of decentralized nonlinear systems. For instance, in [10] it is shown to be satisfied for an intervehicle spacing regulation problem in a platoon of an automated highway system.

Subsystem (1) can be expressed as:

$$\dot{\varsigma}_i = \Lambda_{i0}\varsigma_i + b_i[f_i(\underline{x}_i) + g_i(\underline{x}_i)u_i + \Delta_i(\underline{x})] \quad (3)$$

where

$$\Lambda_{i0} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \quad b_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (4)$$

and

$$\varsigma_i = [\varsigma_{i,1}, \varsigma_{i,2}, \dots, \varsigma_{i,d_i}]^T, \quad \varsigma_{i,1} = y_i, \quad \varsigma_{i,2} = \dot{y}_i, \dots, \quad \varsigma_{i,d_i} = y_i^{(d_i-1)}$$

Assume that the given reference y_{ir} is bounded and have up to $d_i - 1$ bounded derivatives. The reference vector is denoted as $Y_{ir} = [y_{ir}, \dot{y}_{ir}, \ddot{y}_{ir}, \dots, y_{ir}^{(d_i-1)}]^T$. Define the tracking error of the i^{th} subsystem as $e_{i0} = y_{ir} - y_i$. Then the error vector of the i^{th} subsystem is given by $e_i = [e_{i0}, \dot{e}_{i0}, \dots, e_{i0}^{(d_i-1)}]^T$.

It is desired that the output error of the i^{th} subsystem follow $e_i^{(d_i)} + k_{i,d_i-1}e_i^{(d_i-1)} + \dots + k_{i,0}e_{i0} = 0$, here the coefficients are picked so that each:

$$\hat{L} = \varsigma_i^{d_i} + k_{i,d_i-1}\varsigma_i^{d_i-1} + \dots + k_{i,0} \quad (5)$$

has its roots in the open left-half complex plane (in Hurwitz).

If the subsystem S_i is well known ($g_i(\underline{x}_i) \neq 0$) and free of external disturbances; ($\Delta_i(\underline{x}_1, \underline{x}_2, \dots, \underline{x}_N) = 0$), then the primary control should be designed to have the following idealized control law:

$$u_i^* = \frac{1}{g_i(\underline{x}_i)} (-f_i(\underline{x}_i) + K_i^T e_i + y_{ir}^{(d_i)}) \quad (6)$$

In spite of the primary control (6) which mathematically cancels the given system and then places it in a stabilizing part, so as to guarantee $\lim_{t \rightarrow \infty} e_i = 0$, it is clear that, in practice, an exact

cancellation of the given system nonlinearity is theoretically unrealizable and physically impossible. Thus, in this study, the direct adaptive approach implements an adaptive fuzzy system to approximate the idealized control action, and with the indirect approach we approximate the unknown dynamics for each subsystem ($f_i(\underline{x}_i)$ and $g_i(\underline{x}_i)$).

3 Direct adaptive fuzzy decentralized control

In this section, a direct adaptive output-feedback fuzzy decentralized controller is designed, with guaranteed stability of the integrated closed loop system.

Assume that in subsystem (1) $g_i(\underline{x}_i) \neq 0$. The direct adaptive controller is designed as:

$$u_i = u_i(\underline{x}_i, \underline{\theta}_i) + g_i^{-1}(\underline{x}_i)(a_i(t)e_i^T p_i b_i + u_{ih}) \quad (7)$$

where $u_i(\underline{x}_i, \underline{\theta}_i) = \underline{\theta}_i^T \varphi_i(\underline{x}_i)$, $\underline{\theta}_i = [\theta_i^1, \theta_i^2, \dots, \theta_i^{m_i}]^T$ are parameter vectors and $\varphi_i(\cdot) = [\varphi_{i,1}, \varphi_{i,2}, \dots, \varphi_{i,m_i}]^T$ is a regressive vector with regressor φ_i^l ($1 \leq l \leq m_i$, where m_i is the number of rules), which is defined as a fuzzy basis function [5]. The term $a_i(t)e_i^T p_i b_i$ is used to compensate unknown effects from the

interconnections ($p_i \in \mathfrak{R}^{d_i \times d_i}$ is a positive definite matrix defined by a Lyapunov matrix equation and $b_i \in \mathfrak{R}^{d_i}$ is a vector), and u_{ih} is auxiliary control compensation.

Substituting (7) to (1), and adding $g_i(x_i)u_i^*$ and then subtracting $g_i(x_i)u_i^*$ on the right-hand side of (1), we obtain,

$$y_i^{(d_i)} = g_i(x_i)(u_i(x_i, \theta_i) - u_i^*) + a_i(t)e_i^T p_i b_i + u_{ih} + K_i^T e_i + y_{ir}^{(d_i)} + \Delta_i(x) \quad (8)$$

The error dynamics for the i^{th} subsystem may be expressed as:

$$e_i^{(d_i)} = y_{ir}^{(d_i)} + g_i(x_i)(u_i^* - u_i(x_i, \theta_i)) - a_i(t)e_i^T p_i b_i - u_{ih} - K_i^T e_i - y_{ir}^{(d_i)} - \Delta_i(x) \quad (9)$$

Define the optimal parameter vectors and fuzzy approximation error as:

$$\theta_i^* = \arg \min_{\theta_i \in \Omega_i} \left\{ \sup_{x_i \in \mathfrak{R}^{n_i}} |u_i^* - u_i(x_i, \theta_i)| \right\} \quad (10)$$

where $\Omega_i = \{\theta_i / \|\theta_i\| \leq M_i\}$ is the convex compact set which contains feasible parameter sets for θ_i^* .

Define the parameter error as $\Phi_i = \theta_i^* - \theta_i$.

In the analysis to follow, we will use the fact that $u_i^* - u_i(x_i, \theta_i) = \Phi_i^T \varphi_i(x_i) - w_i$ where

$\Phi_i = [\tilde{\Phi}_{i,1}, \tilde{\Phi}_{i,2}, \dots, \tilde{\Phi}_{i,m_i}]^T$ are parameter vectors, $\varphi_i(x_i) = [\varphi_{i,1}, \varphi_{i,2}, \dots, \varphi_{i,m_i}]^T$ are defined above which is defined as a fuzzy basis function [5].

The dynamics equations of the i^{th} subsystem can be written as:

$$\dot{e}_i = \Lambda_i e_i + b_i [g_i(x_i) \Phi_i^T \varphi_i(x_i) + \beta_i(x_i) w_i - a_i(t)e_i^T p_i b_i - u_{ih} - \Delta_i(x)] \quad (11)$$

and w_i represent the fuzzy approximation error of the i^{th} subsystem.

Assumption 2: We assume that, there exists a function $T_{w_i}(x_i) > 0$ such that:

$$|\beta_i(x_i) w_i| < T_{w_i}(x_i) \quad \forall 1 \leq i \leq N \quad (12)$$

The direct adaptive fuzzy decentralized control that we have proposed in (7) can be classified as:

$$\dot{\theta}_i = \eta_i \text{Proj}[\cdot] \quad (13)$$

where $\text{Proj}[\cdot]$ is the projection operator [5]

$$u_{ih} = T_{w_i}(x_i) \text{sign}(e_i^T p_i b_i) \quad (14)$$

$$\dot{a}_i = \eta_{a_i} (e_i^T p_i b_i)^2 \quad (15)$$

where $\eta_i > 0$, and $\eta_{a_i} > 0$ are fixed adaptive gains.

Theorem 1:

Consider the nonlinear subsystem (1), suppose that assumptions 1-2 are satisfied. If there exists a matrix

$p_i = p_i^T > 0$ satisfying the Lyapunov

equation: $\Lambda_i^T p_i + p_i \Lambda_i + Q_i = 0$ where $Q_i = Q_i^T > 0$.

The adaptive fuzzy decentralized controller law is chosen as (7) with parameter adaptation law (13)-(15). Then the proposed fuzzy decentralized control scheme can guarantee that (i) all the variables of the closed-loop system are bounded and (ii) performance tracking is achieved.

Proof: consider the following Lyapunov function for the i^{th} subsystem:

$$v_i = e_i^T p_i e_i + \frac{1}{\eta_i} \Phi_i^T \Phi_i + \frac{1}{2\eta_{a_i}} \Phi_{a_i}^T \Phi_{a_i} \quad (16)$$

where $\Phi_i = \theta_i^* - \theta_i$, $\Phi_{a_i} = a_i - \tau_i^*$, τ_i^* well be defined shortly, and each $p_i \in \mathfrak{R}^{d_i \times d_i}$ is a positive definite and symmetric matrix.

Taking the time derivative of v_i , yields

$$\dot{v}_i = (\dot{e}_i^T p_i e_i + e_i^T p_i \dot{e}_i) + \frac{2}{\eta_i} \Phi_i^T \dot{\Phi}_i + \frac{1}{\eta_{a_i}} \Phi_{a_i}^T \dot{\Phi}_{a_i} \quad (17)$$

Substituting (11) into (17), applying (12)-(14) and choosing $\zeta_i(t) = e_i^T p_i b_i$ yields

$$\begin{aligned} \dot{v}_i = & -e_i^T Q_i e_i - \tau_i^* [(e_i^T p_i b_i)^2 \\ & + 2e_i^T p_i b_i \frac{\Delta_i(x)}{\tau_i^*} + (\frac{\Delta_i(x)}{\tau_i^*})^2 \\ & - (\frac{\Delta_i(x)}{\tau_i^*})^2] \\ = & -e_i^T Q_i e_i - \tau_i^* (e_i^T p_i b_i + \frac{\Delta_i(x)}{\tau_i^*})^2 \\ & + \frac{1}{\tau_i^*} (\Delta_i(x))^2 \end{aligned} \quad (18)$$

so that if each $\tau_i^* > 0$, we simply obtain

$$\dot{v}_i \leq -e_i^T Q_i e_i + \frac{1}{\tau_i^*} (\Delta_i(x))^2 \quad (19)$$

Now consider the composite system Lyapunov

candidate $V = \sum_{i=1}^N \varepsilon_i v_i$ where each $\varepsilon_i > 0$, yields

$$\dot{V} \leq \sum_{i=1}^N \varepsilon_i [-e_i^T Q_i e_i + \frac{1}{\tau_i^*} (\sum_{j=1}^N q_{ij} \|e_j\|_2)^2] \quad (20)$$

Since $\sum_{j=1}^N q_{ij} \|e_j\|_2 = \psi^T \chi_i$, where

$\psi = [\|e_1\|_2, \|e_2\|_2, \dots, \|e_N\|_2]^T$ and

$\chi_i = [q_{i1}, q_{i2}, \dots, q_{iN}]^T$, let λ_i the real part of the eigenvalue of Q_i , (20) may be written as

$$\dot{V} \leq \sum_{i=1}^N \varepsilon_i [-\lambda_i \|e_i\|_2^2 + \frac{1}{\tau_i^*} \psi^T \chi_i \chi_i^T \psi] \quad (21)$$

Define $K^* = [\tau_1^*, \tau_2^*, \dots, \tau_N^*]$. Let $\tau_i^* = \tau^*, i = 1, 2, \dots, N$ for some $0 < \tau^*$. Define

$D = \text{diag}\{\varepsilon_1 \lambda_1, \varepsilon_2 \lambda_2, \dots, \varepsilon_N \lambda_N\}$ and

$M = \sum_{i=1}^N \varepsilon_i \chi_i \chi_i^T$ so that $\dot{V} \leq -\psi^T A \psi$, where

$A = D - \frac{1}{\tau^*} M$. Then for some sufficiently large

$\tau^* > 0$ the matrix A is positive definite. The diagonal dominance property may be established using Gershgorin's Theorem [3]. Now define $K^* = [\tau^*, \tau^*, \dots, \tau^*]^T \in \mathbb{R}^N$ as

$$K^* = \arg \min_{\substack{K^* \in \mathbb{R}^N \\ 0 < \tau^*}} \left\{ K^{*T} K^* : A = D - \frac{1}{\tau^*} M \text{ is positive definite} \right\} \quad (22)$$

There exists sufficiently large τ^* such that A , defined by (22), is positive definite, which implies that $V \in \ell_\infty$, and thus $\|\psi\|_2 \in \ell_\infty$. Also

$$\int_0^\infty \psi^T A \psi dt \leq -\int_0^\infty \dot{V} dt + \text{const} \quad (23)$$

so that $\|\psi\|_2 \in \ell_2$. Since all of the signals are well defined, we also have $\dot{e}_i \in \ell_\infty^{d_i}$ so that $d/dt \|e_i\|_2 = e_i^T \dot{e}_i / \|e_i\|_2 \leq \|e_i\|_2 \in \ell_\infty$. Using

Barbalat's Lemma, we thus establish that $\lim_{t \rightarrow \infty} \|\psi\|_2 = 0$, thus we are guaranteed asymptotically stable tracking for each of the subsystems.

4 Indirect adaptive fuzzy decentralized control

In this section, it is assumed that the function $f_i(\underline{x}_i)$ and $g_i(\underline{x}_i)$ are unknown. Take a universal fuzzy system $\hat{f}_i(\underline{x}_i / \underline{\theta}_i)$ with $\underline{x}_i \in U_{x_i}$ for some compact set U_{x_i} to approximate the uncertain term $f_i(\underline{x}_i)$ where $\underline{\theta}_i$ contains the tunable parameters. Here the linearly parameterized fuzzy model [8] is employed in the approximation procedure. Then we replace $f_i(\underline{x}_i)$ and $g_i(\underline{x}_i)$ by the fuzzy system $\hat{f}_i(\underline{x}_i / \underline{\theta}_{i1})$ and $\hat{g}_i(\underline{x}_i / \underline{\theta}_{i2})$ respectively, with singleton fuzzifier, center average defuzzifier, and

product inference. The fuzzy system $\hat{f}_i(\underline{x}_i / \underline{\theta}_{i1})$ and $\hat{g}_i(\underline{x}_i / \underline{\theta}_{i2})$ can be expressed as:

$$\begin{aligned} \hat{f}_i(\underline{x}_i / \underline{\theta}_{i1}) &= \underline{\theta}_{i1}^T \varphi_i(\underline{x}_i) \text{ and} \\ \hat{g}_i(\underline{x}_i / \underline{\theta}_{i2}) &= \underline{\theta}_{i2}^T \varphi_i(\underline{x}_i) \text{ for } i = 1, 2, \dots, N \end{aligned} \quad (24)$$

where $\underline{\theta}_{ik} = [\theta_{ik}^1, \theta_{ik}^2, \dots, \theta_{ik}^{m_i}] \in \mathbb{R}^{m_i}$ is a parameter vector ($k = 1, 2$) and $\varphi_i(\underline{x}_i) = [\varphi_i^1(\underline{x}_i), \varphi_i^2(\underline{x}_i), \dots, \varphi_i^{m_i}(\underline{x}_i)] \in \mathbb{R}^{m_i}$ is a regressive vector with the regressor $\varphi_i^l(\underline{x}_i)$ defined as,

$$\varphi_i^l(\underline{x}_i) = \frac{\prod_{j=1}^{n_i} \mu_{F_{i,j}^l}(\underline{x}_i^j)}{\sum_l \prod_{j=1}^{n_i} \mu_{F_{i,j}^l}(\underline{x}_i^j)} \quad (25)$$

where $\mu_{F_{i,j}^l}(\cdot)$ is the membership functions for

$1 \leq l \leq m_i$ (m_i is the number of rules) and $1 \leq j \leq n_i$

in this paper, we present the decentralized adaptive fuzzy controller defined as

$$u_i = u_{if} + \frac{1}{\hat{g}_i(\underline{x}_i / \underline{\theta}_{i2})} (-\hat{f}_i(\underline{x}_i / \underline{\theta}_{i1}) + \quad (26)$$

$$K_i^T e_i + y_{ir}^{(d_i)} + a_i(t) e_i^T p_i b_i / 2 + u_{ih})$$

where u_{if} is the fuzzy controller, introduced to perform the main control action, which is given by synthesizing fuzzy control rules from human experts and/or by trial and error designing tools.

The decentralized fuzzy controller u_{if} is constructed

from the following $\prod_{k=1}^{d_i} n_{i,k}$ rules:

$$\begin{aligned} R_i^{l_1, \dots, l_{d_i}} : & \text{ if } e_{i,1} \text{ is } A_{i,1}^{l_1} \text{ and } e_{i,2} \text{ is } A_{i,2}^{l_2} \text{ and } \dots \\ & \text{ and } e_{i,d_i} \text{ is } A_{i,d_i}^{l_{d_i}} \text{ then } L_i^{l_1, \dots, l_{d_i}} \text{ is } C_i^{l_1, \dots, l_{d_i}} \end{aligned} \quad (27)$$

where $n_{i,k}$ define the number of fuzzy sets $A_{i,k}^{l_k}$ in $U_{i,k}$ ($1 \leq l_k \leq n_{i,k}$ and $1 \leq k \leq d_i$) such that for any

$e_{i,k} \in U_{i,k}$, there exists a fuzzy set $A_{i,j}^{l_j}$ so that the memberships function $\mu_{A_{i,j}^{l_j}}(e_{i,k}) \neq 0$. The centres of

these fuzzy sets are adapted by the proposed law which will be defined below.

According to the universal approximation theorem [6], there exist optimal approximation parameters θ_{i1}^* and θ_{i2}^* such that $\hat{f}_i(\underline{x}_i / \theta_{i1}^*)$ and $\hat{g}_i(\underline{x}_i / \theta_{i2}^*)$ can, respectively, approximate $f_i(\underline{x}_i)$ and $g_i(\underline{x}_i)$ as best as possible. Define the optimal parameter vectors and fuzzy approximation errors

$$\begin{aligned}\theta_{i1}^* &= \arg \min \left\{ \sup_{x_i \in \mathcal{R}^{n_i}} |f_i(x_i) - \hat{f}_i(x_i / \theta_{i1})| \right\} \\ \theta_{i2}^* &= \arg \min \left\{ \sup_{x_i \in \mathcal{R}^{n_i}} |g_i(x_i) - \hat{g}_i(x_i / \theta_{i2})| \right\}\end{aligned}\quad (28)$$

where Ω_{i1} and Ω_{i2} are the convex compact sets, which contain feasible parameter sets for θ_{i1}^* and θ_{i2}^* respectively, and

$$\begin{aligned}\Omega_{i1} &= \{\theta_{i1} : \theta_{i1} \theta_{i1}^T \leq M_{i1}\} \\ \Omega_{i2} &= \{\theta_{i2} : \text{tr}(\theta_{i2} \theta_{i2}^T) \leq M_{i2}\}\end{aligned}\quad (29)$$

define

$$\Delta f_i(x_i) = f_i(x_i) - \hat{f}_i(x_i / \theta_{i1}^*)$$

and

$$\Delta g_i(x_i) = g_i(x_i) - \hat{g}_i(x_i / \theta_{i2}^*) \quad (30)$$

which denote the minimum approximation errors.

Throughout this section we need the following assumption:

Assumption 3: there exists a positive function $0 < M_{w_i}(x_i)$ such that

$$|\Delta f_i(x_i) + \Delta g_i(x_i)u_i| \leq M_{w_i}(x_i) \quad \forall 1 \leq i \leq N \quad (31)$$

Substituting (26.) to (1), the tracking error dynamic equation can be written as

$$\begin{aligned}\dot{e}_i &= \Lambda_i e_i + b_i [f_i(x_i) - \hat{f}_i(x_i / \theta_{i1}) \\ &\quad + (g_i(x_i) - \hat{g}_i(x_i, \theta_{i2}))u_i \\ &\quad - a_i(t)e_i^T p_i b_i / 2 - u_{ih} - \Delta_i(x)]\end{aligned}\quad (32)$$

where

$$\Lambda_i = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -K_{i0} & K_{i1} & \cdots & K_{id_i} \end{bmatrix}, \quad b_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (33)$$

from (24) and (30), (31) can be written as

$$\begin{aligned}\dot{e}_i &= \Lambda_i e_i + b_i [\Phi_{i1}^T \varphi_i(x_i) + \\ &\quad \Phi_{i2}^T \varphi_i(x_i)u_i + \Delta f_i(x_i) + \Delta g_i(x_i) \\ &\quad - a_i(t)e_i^T p_i b_i / 2 - u_{ih} - \Delta_i(x)]\end{aligned}\quad (34)$$

where the parameter error vectors are defined as

$$\Phi_{i1} = \theta_{i1}^* - \theta_{i1}, \quad \Phi_{i2} = \theta_{i2}^* - \theta_{i2}$$

The following update laws are now defined for the decentralized indirect adaptive controller:

$$\dot{\theta}_{i1} = \eta_{i1} e_i^T p_i b_i \varphi_i(x_i) \quad (35)$$

$$\dot{\theta}_{i2} = \eta_{i2} e_i^T p_i b_i \varphi_i(x_i)u_i \quad (36)$$

$$u_{ih} = -M_{w_i}(x_i) \text{sign}(e_i^T p_i b_i) \quad (37)$$

$$\dot{a}_i = \eta_{a_i} (e_i^T p_i b_i)^2 \quad (38)$$

$$u_{if} = \sigma_i \Gamma_i^T \varphi_i(x_i) \quad (39)$$

where η_{i1}, η_{i2} and η_{a_i} are fixed adaptive gains. The parameters update law for the isolated system identifier (35)-(36) are used to estimate the dynamics of the subsystem under control. The update law (37) is designed to compensate for the effects of the representation error, where (38) is used to stabilize the subsystem by estimating the effects of the interconnections. The vector

$\Gamma_i^{l_1, \dots, l_{d_i}} = (\Gamma_i^{l_1, 1}, \Gamma_i^{l_1, 2}, \dots, \Gamma_i^{l_1, n_{i,j}})$ is the centre average

of $C_i^{l_1, \dots, l_{d_i}}$. We define $\Gamma_i = (\Gamma_i^{l_1, 1}, \Gamma_i^{l_1, 2}, \dots, \Gamma_i^{l_1, n_{i,j}})$ as

the collection of $\Gamma_i^{l_1, \dots, l_{d_i}}$'s for

$l_1 = 1, 2, \dots, M_{i,1}; \dots; l_{d_i} = 1, 2, \dots, n_{i,d_i}$. The centre

of the l^{th} set C_i^l ($l = (l_1, l_2, \dots, l_{d_i})$) is given by the proposed equation

$$\Gamma_i^{l_1, \dots, l_{d_i}} = \begin{cases} \Gamma_i^{l_1, \dots, l_{d_i}} & \text{if } \max_{j=1,2,\dots,n_i} \mu_{A_{ij}^{l_j}}(e_{ij}) = 0 \\ \frac{\sum_{j=1}^{d_i} e_{ij} (\mu_{A_{ij}^{l_j}}(e_{ij}))}{\max_{j=1,2,\dots,d_i} (\mu_{A_{ij}^{l_j}}(e_{ij}))} & \text{otherwise} \end{cases} \quad (40)$$

σ_i is positive constant.

Theorem 2:

Consider the nonlinear subsystem (1), with the assumptions 1 and 3 are satisfied. If there exists a matrix $p_i = p_i^T > 0$ satisfying the Lyapunov equation:

$\Lambda_i^T p_i + p_i \Lambda_i + Q_i = 0$ where $Q_i = Q_i^T > 0$. Then the proposed control (26) with adaptation laws (35-38) will ensure that, for $i = 1, 2, \dots, N$.

(i) all the variables of the closed-loop system are bounded and (ii) performance tracking is achieved.

Proof: take the error dynamic equation (34), and consider the Lyapunov function candidate

$$\begin{aligned}v_i &= e_i^T p_i e_i + \frac{1}{\eta_{i1}} \Phi_{i1}^T \Phi_{i1} \\ &\quad + \frac{1}{\eta_{i2}} \Phi_{i2}^T \Phi_{i2} + \frac{1}{2\eta_{a_i}} \Phi_{a_i}^T \Phi_{a_i}\end{aligned}\quad (41)$$

where $\Phi_{i1} = \theta_{i1}^* - \theta_{i1}, \Phi_{i2} = \theta_{i2}^* - \theta_{i2}, \Phi_{a_i} = a_i - \tau_i^*$,

τ_i^* will be defined shortly, and each $P_i \in \mathcal{R}_i^{d_i \times d_i}$ is a positive definite and symmetric matrix.

The time derivative of v_i along the error trajectory (41)

$$\begin{aligned} \dot{v}_i = & \dot{e}_i^T p_i e_i + e_i^T p_i \dot{e}_i + \frac{2}{\eta_{i1}} \Phi_{i1}^T \dot{\Phi}_{i1} \\ & + \frac{2}{\eta_{i2}} \Phi_{i2}^T \dot{\Phi}_{i2} + \frac{1}{\eta_{a_i}} \Phi_{a_i}^T \dot{\Phi}_{a_i} \end{aligned} \quad (42)$$

Substituting (34) into (42) and choosing

$\zeta_i(t) = e_i^T p_i b_i$ we obtain:

$$\begin{aligned} \dot{v}_i = & e_i^T (\Lambda_i^T p_i + p_i \Lambda_i) e_i \\ & + 2e_i^T p_i b_i [\Phi_{i1}^T \varphi_i(\underline{x}_i) + \Phi_{i2}^T \varphi_i(\underline{x}_i) u_i \\ & + \Delta f_i(\underline{x}) + \Delta g_i(\underline{x}_i) u_i - a_i(t) e_i^T p_i b_i \\ & - u_{ih} - \Delta_i(\underline{x}_i)] + \frac{2}{\eta_{i1}} \Phi_{i1}^T \dot{\Phi}_{i1} \\ & + \frac{2}{\eta_{i2}} \Phi_{i2}^T \dot{\Phi}_{i2} + \frac{1}{\eta_{a_i}} \Phi_{a_i}^T \dot{\Phi}_{a_i} \end{aligned} \quad (43)$$

Applying parameters adaptation laws (35)-(38), yields

$$\begin{aligned} \dot{v}_i = & -\frac{1}{2} e_i^T Q_i e_i - 2e_i^T p_i b_i \Delta_i(\underline{x}_i) \\ & - \tau_i^* (e_i^T p_i b_i)^2 \\ \leq & -\frac{1}{2} e_i^T Q_i e_i + \frac{1}{\tau_i^*} (\Delta_i(\underline{x}_i))^2 \end{aligned} \quad (44)$$

now consider the composite system Lyapunov

candidate $V = \sum_{i=1}^N \varepsilon_i v_i$, where each $\varepsilon_i > 0$. Taking

the derivate of V gives

$$\dot{V} \leq \sum_{i=1}^N \varepsilon_i [-\lambda_i \|e_i\|_2^2 + \frac{1}{\tau_i^*} \psi^T \chi_i \chi_i^T \psi] \quad (45)$$

where λ_i is the real part of the eigenvalue of Q_i .

Define $K^* = [\tau_1^*, \tau_2^*, \dots, \tau_N^*] \in \mathcal{R}^N$.

Let $D = \text{diag}\{\varepsilon_1 \lambda_1(Q_1), \dots, \varepsilon_N \lambda_N(Q_N)\}$ and

$M = \sum_{i=1}^N \varepsilon_i \chi_i \chi_i^T$, so that $\dot{V} \leq -\psi^T A \psi$, where

$A = D - \frac{1}{\tau_i^*} M$. Then for some sufficiently

large $\tau_i^* > 0$, the matrix A is positive define. The remainder of the theorem (2) follows as for the direct adaptive case.

5 Simulation results

a double-inverted pendulum connected by a spring can be considered as the simplified example of the large-scale system. Each pendulum may be positioned by a torque input u_i applied by a

servomotor at its base. It is assumed that both ϕ_i and $\dot{\phi}_i$ (angular position and rate) are available to the i^{th} controller for $i=1,2$. Fig.1.

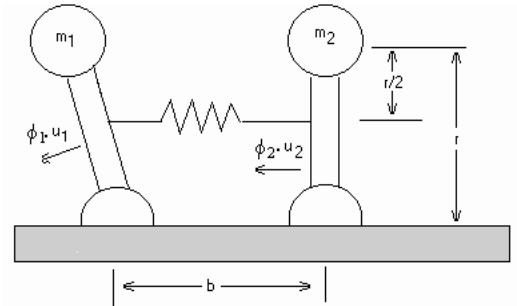


Fig.1. Two inverted pendulums connected by a spring.

Consider a double-inverted pendulum model [10]. The equations which describe the motion of the pendulums are defined by

$$\begin{aligned} \dot{x}_{11} &= x_{12} \\ \dot{x}_{12} &= \left(\frac{m_1 g r}{j_1} - \frac{k r^2}{4 j_1} \right) \sin(x_{11}) + \frac{k r}{2 j_1} (l - b) \\ &+ \frac{u_1}{j_1} + \frac{k r^2}{4 j_1} \sin(x_{21}) \end{aligned} \quad (46)$$

$$\begin{aligned} \dot{x}_{21} &= x_{22} \\ \dot{x}_{22} &= \left(\frac{m_1 g r}{j_2} - \frac{k r^2}{4 j_2} \right) \sin(x_{21}) + \frac{k r}{2 j_2} (l - b) \\ &+ \frac{u_2}{j_2} + \frac{k r^2}{4 j_2} \sin(x_{12}) \end{aligned} \quad (47)$$

where $x_{11} = \phi_1$ and $x_{21} = \phi_2$ are the angular displacements of the pendulums from vertical. The parameters $m_1 = 2\text{kg}$ and $m_2 = 2.5\text{kg}$ are the pendulum end masses, $j_1 = 0.5\text{kg}$ and $j_2 = 0.625\text{kg}$ are the moments of inertia, the constant of connecting spring is $k = 100\text{N/m}$, the pendulum height is $r = 0.5\text{m}$, the natural length of the spring is $l = 0.5\text{m}$ and the gravitational acceleration is $g = 9.81\text{m/s}^2$. The distance between the pendulum hinges is defined as $b = 0.4\text{m}$ (with $b < l$ in this example, so that the pendulum links repels each other when both are in the upright position (Fig 1)).

In (45) and (47)

$$f_1(x_1) = \left(\frac{m_1 g r}{j_1} - \frac{k r^2}{4 j_1} \right) \sin(x_{11}), \quad g_1(x_1) = 1/j_1;$$

$$f_2(x_2) = \left(\frac{m_1 g r}{j_2} - \frac{k r^2}{4 j_2} \right) \sin(x_{21}), \quad g_2(x_2) = 1/j_2$$

$$\Delta_1(x) = \frac{kr}{2j_1}(l-b) + \frac{kr^2}{4j_1}\sin(x_{21}),$$

$$\Delta_2(x) = \frac{kr}{2j_2}(l-b) + \frac{kr^2}{4j_2}\sin(x_{12}),$$

the motion equations fit the format of system (1). Here we will attempt to drive the angular positions to zero, so that $e_i = -\phi_i$ (i.e., $y_{1r} = y_{2r} = 0$) for $i = 1, 2$

To construct the fuzzy approximators $u_i(\underline{x}_i, \underline{\theta}_i)$ in (9) and $\hat{\alpha}_{ij}(\underline{x}_i / \theta_{i1}), \hat{\beta}_{ij}(\underline{x}_i / \theta_{i2})$ in (24), we define three fuzzy sets for component of each $\underline{x}_1 = (x_{11}, x_{12})$ and $\underline{x}_2 = (x_{21}, x_{22})$ with labels $A_{x_{ij}}^1$, $A_{x_{ij}}^2$, $A_{x_{ij}}^3$, $A_{x_{ij}}^4$ and $A_{x_{ij}}^5$ characterized by:

$$\mu_{A_{x_{ij}}^1}(x_{ij}) = \exp(-(x_{ij} + 0.8)^2)$$

$$\mu_{A_{x_{ij}}^2}(x_{ij}) = \exp(-(x_{ij} + 0.4)^2)$$

$$\mu_{A_{x_{ij}}^3}(x_{ij}) = \exp(-(x_{ij})^2)$$

$$\mu_{A_{x_{ij}}^4}(x_{ij}) = \exp(-(x_{ij} - 0.4)^2)$$

$$\mu_{A_{x_{ij}}^5}(x_{ij}) = \exp(-(x_{ij} - 0.8)^2)$$

with $n_{i,j} = 5$, $j = 1, 2$ and $i = 1, 2$.

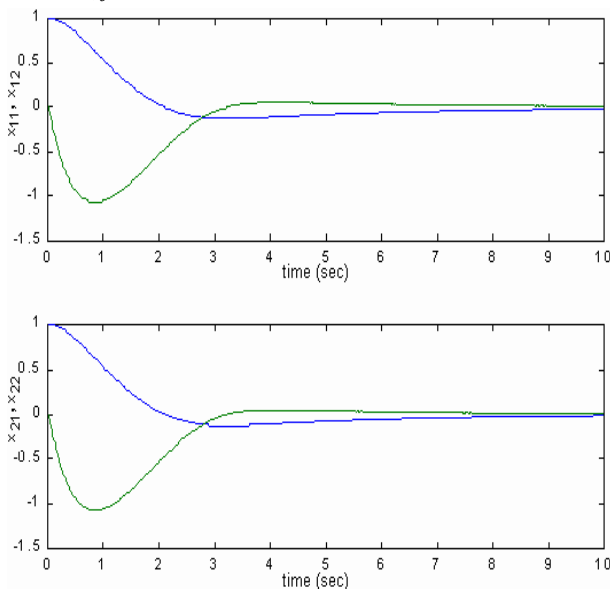


Fig. 2, Control of the pendulums using the proposed direct adaptive decentralized technique.

Conclusion

In the course of this paper, we have presented an adaptive output-feedback fuzzy decentralized control for a class of large-scale nonlinear

Defining 25 fuzzy rules, in the following linguistic description:

$R_i^{(l)} : \text{if } x_{i1} \text{ is } A_{x_{i1}}^{j_1} \text{ and } x_{i2} \text{ is } A_{x_{i2}}^{j_2} \text{ then } y_i^l \text{ is } C_i^l$

Denoting $D_i = \sum_{l=1}^{25} \prod_{k=1}^2 \mu_{A_{x_{ik}}^l}(x_{ik})$,

$$\varphi_i(x_i) = [\varphi_{i,1}(x_i)/D_i, \varphi_{i,2}(x_i)/D_i, \dots, \varphi_{i,25}(x_i)/D_i]^T$$

we can construct the fuzzy system (7) and (24) respectively, as follows: we choose $\eta_i = 0.01$, $\eta_{a_i} = 0.001$ and

$$\eta_{i1} = 0.001, \eta_{i2} = 0.001, \eta_{a_i} = 0.0001, \sigma_i = 0.15,$$

$Q_i = \text{diag}(10, 10)$, and each Λ_i so that

$$\hat{L}(\zeta) = \zeta^2 + 4\zeta + 4 \text{ has roots at } (-2, -2).$$

Choose the initial conditions to be the same for both direct and indirect approaches in the simulations:

$$(x_{11}, x_{12}, x_{21}, x_{22})^T = (1, 1, 1, 1)^T, \quad \theta_{i1} = \theta_{i2} = 0_{25 \times 1} \text{ and}$$

$\Gamma_i = [-1 \ -0.75 \ -0.5 \ -0.25 \ 0 \ 0.25 \ 0.5 \ 0.75 \ 1]$. For the direct approach, the simulation results are shown in Fig 2. The simulation results are given in Fig.3

Both direct and indirect fuzzy decentralized controllers achieve good performance, as can be seen from the simulation results.

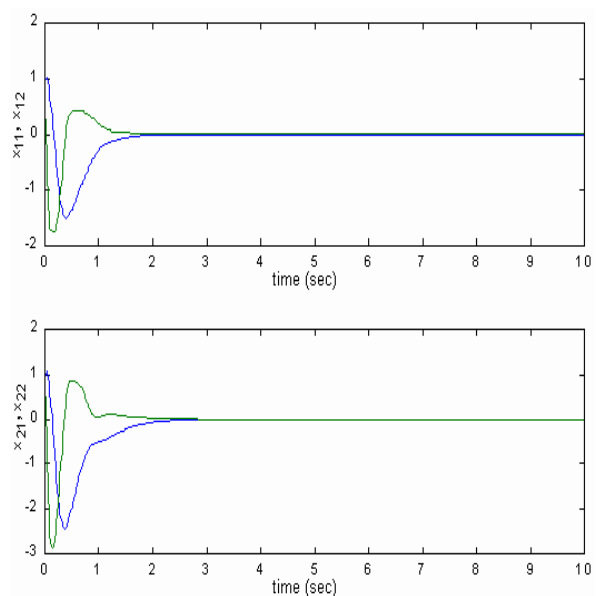


Fig. 3, Control of the pendulums using the proposed indirect adaptive decentralized technique.

systems. In both, direct and indirect adaptive proposed design methods, fuzzy logic systems are used to estimate the part the decentralized adaptive fuzzy controller and unknown nonlinear dynamics

without knowing bounds of interconnections. Furthermore, the stability of nonlinear interconnected systems is also guaranteed and ensures asymptotic tracking using only local measurement. The proposed approaches are simple without complex algorithms. Simulations have shown that the proposed controls methodology is effective, with guaranteed stability and satisfying performance.

References

- [1] A. Hmamed and L. Radouane, Decentralized nonlinear adaptive feedback stabilization of Large-scale interconnected systems, *IEE. Proc. D contr. Theory Appl.*, vol. 130, Mar. 1983, pp. 57-62.
- [2] D. T. Gavel and D. D. Siljak, Decentralized Adaptive Control: Structural Conditions for stability, *IEEE Trans. Automatic Contr.*, vol. 34, Apr. 1989, pp. 413-426.
- [3] P. C. Parks and V. Hahn, *Stability theory*, New York: Prentice-Hall, 1992.
- [4] A. Data, Performance improvement in decentralized adaptive control A modified Model reference scheme, *IEEE Trans. Automatic Contr.*, vol. 38, , Nov. 1993, pp. 1717-1722.
- [5] Wang, L. X., Stable adaptive fuzzy control of nonlinear systems, *IEEE. Transaction On Fuzzy Systems*, vol.1, 1993, pp.146-155.
- [6] X. L. Wang, *Adaptive fuzzy systems and control: design and stability analysis*, Englewood Cliffs, NJ: Prentice-Hall.
- [7] J. T. Spooner, and K.M. Passino, Adaptive control of a decentralized nonlinear, *IEEE. Transaction. On Automatic Control*, vol. 41, 1996 pp.280-284.
- [8] SH.O.Wang, K. Tnaka, and M.F.Griffin, An approach to fuzzy control of nonlinear systems: Stability and design issues, *IEEE. Trans. Fuzzy Syst.*, vol.4, 1996, pp.14-23..
- [9] J. Sandeep and K. Farshad, Decentralized Adaptive control of a class of Large-scale interconnected nonlinear systems, *IEEE. Trans. On Autom. Contr.*, vol.42, N.2, Feb.1997, pp.136-154.
- [10] Spooner, J. T. and Passino, K.M. "Decentralised adaptive control of nonlinear systems using radial basis neural networks,". *IEEE. Transaction. On Automatic Control*, 44, pp.2050-2057, 1999.
- [11] B.S.Chen, C.S.Tseng, and H.J.Uang, "Mixed H_∞ / H_2 fuzzy output feedback control design for nonlinear dynamic systems: An LMI approach," *IEEE Trans. Fuzzy Syst.*, Vol.8, pp.250- 265, 2000.
- [12] E.Kim, and H.Lee, "New approaches to relaxed quadratic stability condition of fuzzy

control systems," *IEEE Trans. Fuzzy Syst.*, Vol.8, pp.523-534, 2000.

Annexe E

Fuzzy Adaptive Control of a Class of MISO Nonlinear Systems

I.Lagrat, H.Ouakka et I.Boumhidi

Control and Cybernetics, Vol 37, No 1, pp.177-190, 2008.

Fuzzy adaptive control of a class of MISO nonlinear systems*

by

I. Lagrat, H. Ouakka and I. Boumhidi

LESSI, Faculté Des Sciences
B. P 1796 Atlas, 30000 Fez, Morocco
e-mail" i.lagrat@caramail.com

Abstract: This paper presents a fuzzy adaptive control of a class of MISO nonlinear systems. The dynamic behaviour of each MISO systems is composed of a nonlinear term, interactions effect between the inputs, and disturbances. In these circumstances, adaptive control becomes very difficult to implement and not always an evident task. Thus, the MISO system is approximated by the Takagi-Sugeno fuzzy model. The advantage of this approximation is beneficial in the sense that it allows for converting the nonlinear problem into a linear one. In this respect, the coupling, nonlinearity and unmodeled dynamics are easily compensated. The identification and the control are conducted at the level of each local linear model based on fuzzy approach. The computational load and the complexity of nonlinear approach are reduced and permit wide applicability. The validity and the performance are tested numerically.

Keywords: MISO systems, identification, fuzzy clustering, Takagi-Sugeno fuzzy model, adaptive control, nonlinear system.

1. Introduction

Identification of nonlinear multi-variable processes is an important and challenging problem. For nonlinear static and dynamic systems, the conventional techniques of modelling and identification are difficult to implement and sometimes impracticable. However, other techniques, based on fuzzy logic system, can be used for modelling of the complex nonlinear processes (Chen, Chen, 1994). Fuzzy modelling and identification from the input-output process data is shown to be effective for the approximation of nonlinear uncertain dynamic system (Johancen, Foss, 1993). The Takagi-Sugeno model has attracted the attention of many researchers. Accordingly, the most important issue for fuzzy logic system is how to obtain a system design with the guarantee of stability

*Submitted: February 2006; Accepted: March 2008

and control performance (Tanaka, Wang, 2001). In fact, this model is based on if-then rules, which are characterized by fuzzy antecedent and mathematical functions in the consequent part (Takagi, Sugeno, 1985). The antecedent fuzzy sets divide the input space into a number of fuzzy regions, while the consequent function describes the system behaviour in these regions. The task of fuzzy model construction is to determine both the nonlinear parameters of the membership functions and the linear parameters of the local models (Johancen, Foss, 1993). The human expert, in this regard, is able to formulate the process knowledge in terms of the fuzzy rules. Unfortunately, this can not provide a clear idea of the plant behaviour, because the human expert cannot come up with all the details and might not be able to quantitatively express the observations. However, heuristic approaches, like fuzzy clustering, are applied (Nelles, Fink, Isermann, 2000) to obtain the fuzzy model of nonlinear dynamic using input-output measurement data, e.g., the local linear model tree method and tree construction algorithm (Sugeno, Kang, 1987), or the neuro-fuzzy inference system (Jang, 1993). There are different algorithms that construct fuzzy clusters, such as the c-means algorithm (Bezdek, 1981), the Gath-Geva algorithm (Gath, Geva, 1989), and ultimately the Gustafson-Kessel algorithm (Gustafson, Kessel, 1979), which is the subject of this paper.

The paper presents a fuzzy adaptive control of a class of MISO systems. Each MISO nonlinear system is coupled in the inputs. The considered system is transformed by Takagi-Sugeno approach and fuzzy clustering into linear models (Takagi, Sugeno, 1985). In this transformation, the parameters of the Gaussian membership function and the local linear model are easily obtained. Moreover, the time-variant behaviour of the plant, which is caused by disturbances or aging of components, should be considered in the system model. Therefore, on-line adaptation of the fuzzy models is required. The local linear parameters of each model in the rule consequents of Takagi-Sugeno fuzzy models are updated by a local recursive weighted least-squares algorithm with a forgetting factor (Trabelsi et al., 2004). The proposed control law is derived from Feng and Chen (2005), and is designed to compensate for the interactions between the inputs. The identification and the control are obtained, independently, for each local model. This strategy reduces the computational burden of the global approach.

The paper is organized as follows: Section 2 contains the presentation of the Takagi-Sugeno fuzzy model. Section 3 describes fuzzy identification of MISO nonlinear system. Section 4 presents the adaptive control law designs. In Section 5, we present a numerical example.

2. Takagi-Sugeno fuzzy model of a MISO process

Consider a class of MISO nonlinear system represented as follows:

$$\begin{aligned} y(k+1) = & f(y(k), \dots, y(k-na+1), u_1(k), \dots, u_1(k-nb+1), u_2(k), \dots, \\ & u_2(k-nb+1), \dots, u_{nu}(k), \dots, u_{nu}(k-nb+1), \eta(k)) \end{aligned} \quad (1)$$

where f represents the unknown nonlinear function and $\eta(k)$ are the disturbances. We assume that the upper bound of the orders na and nb are known and equal to n , and then:

$$y(k+1) = f(y(k), \dots, y(k-n+1), u_1(k), \dots, u_1(k-n+1), u_2(k), \dots, u_2(k-n+1), \dots, u_{nu}(k), \dots, u_{nu}(k-n+1), \eta(k))$$

We define the regression vector $\phi(k)$:

$$\phi(k) = [y(k), \dots, y(k-n+1), u_1(k), \dots, u_1(k-n+1), u_2(k), \dots, u_2(k-n+1), \dots, u_{nu}(k-n+1), \dots, u_{nu}(k-n+1), \eta(k)] \quad (2)$$

Then, equation (1) can be written as:

$$y(k+1) = f(\phi(k)). \quad (3)$$

The function $f(\phi(k))$ is approximated by Takagi-Sugeno fuzzy models, which are characterized by the linear function rule consequents as in Takagi and Sugeno (1985). The Takagi-Sugeno MISO rules are estimated from the system input-output data (Babuska, 1998). The base rule contains M rules of the following form:

$$\begin{aligned} R_j : & \text{ if } y(k) \text{ is } \Omega_{j1} \text{ and } u_{nu}(k-n+1) \text{ is } \Omega_{jnu} \text{ then} \\ y_j(k+1) = & \sum_{r=1}^n a_{jr} y_j(k-r+1) + \sum_{r=1}^n b_{jr} u(k-r+1) \\ & + \sum_{l=1}^{nu} \sum_{r=1}^n b_{jlr} u_l(k-r+1) + c_j \quad j = 1, 2, \dots, M. \end{aligned} \quad (4)$$

The number M of rules is determined by testing many values according to the error criterion as given in Trabelsi et al. (2004). The antecedent fuzzy sets Ω_{ji} are defined on the universe of discourse of input i ; c_j is the offset and the linear parameters θ_j in the rule consequents are given by:

$$\theta_j = [a_{j1}, \dots, a_{jn}, b_{j1}, \dots, b_{jn}, \dots, b_{jnu1}, \dots, b_{jnu n}, c_j]. \quad (5)$$

The membership function is chosen as Gaussian with centre v_j and standard deviation σ_j .

We take the product as the AND operator. The output of the fuzzy system with M rules is aggregated as:

$$y(k+1) = \frac{\sum_{j=1}^M \mu_j(\phi(k)) \cdot y_j(k+1)}{\sum_{j=1}^M \mu_j(\phi(k))} \quad (6)$$

or:

$$y(k+1) = \sum_{j=1}^M y_j(k+1) \cdot \Phi_j(\phi(k), v_j, \sigma_j) \quad (7)$$

where $\Phi_j(\phi(k), v_j, \sigma_j)$ denotes the normalized validity function such that

$$\sum_{j=1}^M \Phi_j(\phi(k), v_j, \sigma_j) = 1 \text{ for all the premise inputs } \phi(k).$$

This normalization is achieved by:

$$\Phi_j(\phi(k), v_j, \sigma_j) = \frac{\mu_j(\phi(k))}{\sum_{j=1}^M \mu_j(\phi(k))} \quad (8)$$

with μ_j being the degree of fulfillment of the rule j :

$$\mu_j(\phi(k)) = \exp\left(-\frac{1}{2}\left(\frac{(y(k) - v_{j1})^2}{\sigma_{j1}^2}\right)\right) \cdots \exp\left(-\frac{1}{2}\left(\frac{(u_{nu}(k - n + 1) - v_{ju_{nu}})^2}{\sigma_{ju_{nu}}^2}\right)\right). \quad (9)$$

Once the structure is fixed, the MISO parameters are estimated independently by fuzzy clustering (Babuska, Verbruggen, 1995).

3. Fuzzy identification

Generally, the approximation by a Takagi-Sugeno fuzzy model enables wide applicability in identification, modelling and control (Babuska, 1998; Takagi, Sugeno, 1985). In fact, fuzzy clustering facilitates automatic generation of Takagi-Sugeno rules and their antecedent parameters. The identification procedure consists of two distinct steps, Trabelsi et al. (2004). The first step is off-line identification, where nonlinear parameters of the Gaussian membership function (v_j, σ_j) and linear parameters of the local models θ_j are determined by the fuzzy clustering algorithm (Babuska, 1998). In the second step, the rule consequents are locally adapted on-line by a recursive least-squares algorithm (Nelles, Fink, Isermann, 2000).

3.1. Off-line identification of the fuzzy model

The previous section has shown how the consequent part of Takagi-Sugeno models can be identified by weighted least-squares method. This would not occur unless the antecedent membership functions are given. The Gustafson-Kessel algorithm is used to identify the Takagi-Sugeno models. The available data samples are collected in matrix $Z = [\phi^T y]$, formed by concatenating the regression matrix and the output vector. Through clustering, the data Z are partitioned into N_c clusters. The result is a fuzzy partition matrix $U = [u_{ij}]_{N_c \times N}$,

whose elements $u_{ij} \in [0, 1]$ represent the degree of membership in cluster i , a prototype matrix $V_j = [v_{j1}, \dots, v_{jnu}]$, and the set of cluster covariance matrices $F_j = [F_{j1}, \dots, F_{jnu}]$. Once the triplet (U_j, V_j, F_j) determined, the parameters of the rule premises (v_{ji}, σ_{ji}) and the consequent parameters θ_j are computed. For more details see Babuska and Verbruggen (1996). Afterwards, the antecedent membership functions of the cluster parameters are determined. The Gaussian functions are used to represent the fuzzy sets, ω_{ji} :

$$\omega_{ji}(\phi(k)) = \exp\left(-\frac{1}{2}\left(\frac{(\phi_i(k) - v_{ji})^2}{\sigma_{ji}^2}\right)\right). \quad (10)$$

The consequent parameters θ_j in each rule are estimated separately by weighted least-squares method and by minimizing the following criterion (Babuska, Verbruggen, 1996):

$$\min_{\theta_j} \frac{1}{N} [y - \xi\theta_j]^T Q_j [y - \xi\theta_j] \quad (11)$$

where $\xi = [\phi \ 1]$ is the regression extended by a unitary column and Q_j is a matrix containing the values of validity functions Φ_j of each j th local model.

The weighting matrix is:

$$Q_j = \text{diag}(\Phi_j(\phi(1), v_j, \sigma_j), \dots, \Phi_j(\phi(N), v_j, \sigma_j)). \quad (12)$$

Then, θ_j is obtained as the weighted least-squares solution:

$$\theta_j = (\xi^T Q_j \xi)^{-1} \xi^T Q_j y. \quad (13)$$

3.2. On-line adaptation of fuzzy model

In the on-line phase, the rule consequents (13) are adapted by the recursive weighted least-squares (RWLS) algorithm with a forgetting factor. This approach is used to estimate the parameters of each local linear model. For the j th local linear model, we can compute new parameter estimates $\hat{\theta}_j(k)$ as in (14):

$$\hat{\theta}_j(k) = \hat{\theta}_j(k-1) + \delta_j(k) \left(y(k) - \xi^T(k) \hat{\theta}_j(k-1) \right) \quad (14)$$

$$\delta_j(k) = \frac{P_j(k-1)\xi(k)}{\xi^T(k)P_j(k-1)\xi(k) + \lambda/\Phi_j(\phi(k), v_j, \sigma_j)} \quad (15)$$

$$P_j(k) = \frac{1}{\lambda} [I - \delta_j(k)\xi^T(k)] P_j(k-1). \quad (16)$$

Here, λ is a forgetting factor and $\Phi_j(\phi(k), v_j, \sigma_j)$ provide the weights of actual data. Then, P_j is the matrix of adaptation gain.

4. Fuzzy adaptive control design

In this section, we will shed light on the adaptive control design for MISO nonlinear system using the Takagi-Sugeno fuzzy models as shown in Feng and Chen (2005). In this respect, we propose a control law, which can easily control a MISO nonlinear system without any restrictive conditions on nonlinearity. The strategy of this control is based on the fuzzy approach, which transforms the nonlinear problem to a linear one, and compensates for the unmodeled dynamics and nonlinearity effects. The structure of the proposed control law takes into account the coupling between the inputs and the disturbances, and permits to ensure stability, performance and wide applicability.

The controller rule has the same antecedents and fuzzy sets as in plant rules. The proposed local adaptive control law is given by:

$$\begin{aligned}
 R_j : & \text{ if } y(k) \text{ is } \Omega_{j1} \text{ and } \dots \text{ and } u_{nu}(k-n+1) \text{ is } \Omega_{jnu} \text{ then} \\
 u_j(k) = & \frac{1}{\hat{b}_{j1}} \left[- \sum_{r=1}^n \hat{a}_{jr} y_j(k-r+1) - \sum_{r=2}^n \hat{b}_{jr} u(k-r+1) \right. \\
 & - \sum_{l=1}^{nu} \sum_{r=1}^n \hat{b}_{jlr} u_l(k-r+1) - \hat{c}_j + y_m(k+1) - \alpha_1 e(k) - \dots \\
 & \left. - \alpha_n e(k-n+1) \right], \quad j = 1, 2, \dots, M.
 \end{aligned} \tag{17}$$

The global control law is obtained as:

$$u(k) = \sum_{j=1}^M \Phi_j(\phi(k), v_j, \sigma_j) u_j(k). \tag{18}$$

More precisely, the objective of the fuzzy adaptive control is to find an adaptive control law, which guarantees that the output of the MISO systems can track a given bounded reference signal. The tracking error is given by:

$$e = y - y_m. \tag{19}$$

With $\{\alpha_i\}$ being the coefficients of the Hurwitz polynomial:

$$\alpha(z) = z^n + \alpha_1 z^{n-1} + \dots + \alpha_{n-1} z + \alpha_n \tag{20}$$

we pass over to

THEOREM 1 *For the fuzzy dynamic model of the system (7), if the adaptive control law is chosen as (17), or (18), using the adaptation algorithm (14), then the closed-loop system is stable in the sense that the output and all inputs are bounded for all the time. The output tracking error $e = y - y_m$ will approach to the zero as time goes to infinity.*

Proof. Substituting the control law (18) into the fuzzy dynamic model (7) leads to the following closed-loop system:

$$\begin{aligned} & e(k+1) + \alpha_1 e(k) + \cdots + \alpha_{n-1} e(k-n+2) + \alpha_n e(k-n+1) \\ & = H(k) - \hat{H}(k) + \sum_{j=1}^M \Phi_j(\phi(k), v_j, \sigma_j) (b_{j1} - \hat{b}_{j1}) u_j(k) \end{aligned} \quad (21)$$

where

$$\begin{aligned} H(k) &= \sum_{r=1}^n a_{jr} y_j(k-r+1) + \sum_{r=2}^n b_{jr} u(k-r+1) + \sum_{l=1}^{nu} \sum_{r=1}^n b_{jlr} u_l(k-r+1) + c_j \\ \hat{H}(k) &= \sum_{r=1}^n \hat{a}_{jr} y_j(k-r+1) + \sum_{r=2}^n \hat{b}_{jr} u(k-r+1) + \sum_{l=1}^{nu} \sum_{r=1}^n \hat{b}_{jlr} u_l(k-r+1) + \hat{c}_j. \end{aligned} \quad (22)$$

By defining:

$$x_e(k) = [e(k-n+1) \ e(k-n+2) \ \cdots \ e(k)]^T \quad (23)$$

the above closed-loop system can be expressed in state-space form, as:

$$x_e(k+1) = Ax_e(k) + B \left\{ H(k) - \hat{H}(k) + \sum_{j=1}^M \Phi_j(\phi(k), v_j, \sigma_j) (b_{j1} - \hat{b}_{j1}) u_j(k) \right\}. \quad (24)$$

That is:

$$\begin{aligned} x_e(k+1) &= Ax_e(k) + B \left\{ (H(k) + \sum_{j=1}^M \Phi_j(\phi(k), v_j, \sigma_j) b_{j1} u_j(k)) \right. \\ &\quad \left. - (\hat{H}(k) + \sum_{j=1}^M \Phi_j(\phi(k), v_j, \sigma_j) \hat{b}_{j1} u_j(k)) \right\} \end{aligned} \quad (25)$$

where

$$\begin{aligned} y(k+1) &= H(k) + \sum_{j=1}^M \Phi_j(\phi(k), v_j, \sigma_j) b_{j1} u_j(k) \\ \hat{y}(k+1) &= \hat{H}(k) + \sum_{j=1}^{M_i} \Phi_j(\phi(k), v_j, \sigma_j) \hat{b}_{j1} u_j(k). \end{aligned} \quad (26)$$

We consider:

$$e(k+1) = y(k+1) - \hat{y}(k+1) \quad (27)$$

and so:

$$x_e(k+1) = Ax_e(k) + Be(k+1) \quad (28)$$

where:

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & & 0 \\ \vdots & \vdots & 0 & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ -\alpha_n & -\alpha_{n-1} & \cdots & -\alpha_2 & -\alpha_1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \quad (29)$$

It should be noted that the matrix A has all its eigenvalues located inside the unit circle of the z -plane, and so stability is ensured. ■

5. Simulation example

Consider a MIMO system (Weng, Lang, 1990) having a polynomial nonlinearity described as:

$$A(q^{-1})y(t) = B_d(q^{-1})Z(t) + \eta(t) \quad (30)$$

where q^{-1} is the back shift operator.

$$A(q^{-1}) = \text{diag}[A_i(q^{-1})], \quad A_i(q^{-1}) = \sum_{r=0}^{na(i)} a_{ir} q^{-r}, \quad a_{i0} = 1$$

$$B_d(q^{-1}) = [q^{-d_{ij}} B_{ij}(q^{-1})], \quad B_{ij}(q^{-1}) = \sum_{r=0}^{nb(i,j)} b_{ijr} q^{-r}, \quad b_{ij0} \neq 0$$

d_{ij} is time delay between the input $u_j(t)$ and the output $y_i(t)$, $y(t) \in R^n$, $Z(t) \in R^n$ and $\eta(t) \in R^n$ are respectively the output, nonlinear input and the disturbances vectors. As in Jinxing and Shijium (1989) the nonlinearity is assumed to be represented as follows:

$$Z_i(t) = f_{i0} + f_{i1}u_i(t) + \dots + f_{ip}u_i^{p_i-1}(t). \quad (31)$$

The MIMO system is represented through two MISO nonlinear systems as follows:

$$A_1(q^{-1})y_1(t) = q^{-d_{11}} B_{11}(q^{-1})Z_1(t) + q^{-d_{12}} B_{12}(q^{-1})Z_2(t) + \eta_1(t)$$

$$A_2(q^{-1})y_2(t) = q^{-d_{22}} B_{22}(q^{-1})Z_2(t) + q^{-d_{21}} B_{21}(q^{-1})Z_1(t) + \eta_2(t)$$

with

$$A = \begin{bmatrix} A_1 & A_2 \end{bmatrix}^T = \begin{bmatrix} 1 + 0.35q^{-1} + 0.15q^{-2} \\ 1 + 0.72q^{-1} + 0.05q^{-2} \end{bmatrix},$$

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} 1 + 0.85q^{-1} & 2 + 1.25q^{-1} \\ 0.12 + 0.65q^{-1} & 1.65 + 0.23q^{-1} \end{bmatrix}$$

$$d_{11} = 1, \quad d_{12} = 1, \quad d_{21} = 1, \quad d_{22} = 1$$

$$Z_1(t) = 0.5u_1(t) + 0.25u_1^2(t), \quad Z_2(t) = u_2(t) + 0.83u_2^2(t)$$

$$\eta_1(t) = 0.1\text{rand}(1, 600), \quad \eta_2(t) = 0.12\text{rand}(1, 600)$$

The inputs are u_1 and u_2 . The outputs are y_1 and y_2 , with η_1 and η_2 representing disturbances. A prototype matrix and covariance matrix are determined by the Gustafson-Kessel algorithm (Babuska, 1998):

$$V_1 = \begin{bmatrix} 0.0235 & 0.0720 & 0.0023 & 0.0022 & 0.0041 & 0.0042 \\ 0.0424 & 0.0422 & 0.0022 & 0.0022 & 0.0040 & 0.0040 \\ 0.0846 & 0.0688 & 0.0022 & 0.0022 & 0.0041 & 0.0039 \end{bmatrix}$$

$$F_1 = \begin{bmatrix} 0.0713 & 0.0269 & 8.0335 & 8.4517 & 4.8071 & 4.9064 & 0.0408 \\ 0.1299 & 0.1224 & 6.4028 & 6.8515 & 4.1058 & 3.7856 & 0.0311 \\ 0.3344 & 0.4947 & 12.9454 & 13.1246 & 7.9306 & 8.1032 & 0.0614 \end{bmatrix}$$

$$V_2 = \begin{bmatrix} 0.0313 & 0.0457 & 0.0023 & 0.0023 & 0.0041 & 0.0042 \\ 0.0378 & 0.0377 & 0.0022 & 0.0022 & 0.0040 & 0.0040 \\ 0.0722 & 0.0883 & 0.0022 & 0.0022 & 0.0040 & 0.0039 \end{bmatrix}$$

$$F_2 = \begin{bmatrix} 0.1878 & 0.1397 & 13.1520 & 11.7753 & 6.1561 & 6.7180 & 0.0669 \\ 0.1956 & 0.1556 & 11.1458 & 10.6673 & 5.6689 & 5.8682 & 0.0579 \\ 0.4412 & 0.2936 & 23.6297 & 25.8100 & 14.0950 & 13.6570 & 0.0970 \end{bmatrix}$$

The consequent parameters of each rule of Takagi-Sugeno fuzzy model are computed from equation (13) and adapted by using RLS algorithm with forgetting factor ($\lambda = 0.45$), Trabelsi et al. (2004).

For the rule i :

$$\begin{aligned} R_i : y_{i1}(k+1) &= a_{i111}(k)y_1(k) + a_{i112}(k)y_1(k-1) + b_{i111}(k)u_1(k) \\ &\quad + b_{i112}(k)u_1(k-1) + b_{i121}(k)u_2(k) + b_{i122}(k)u_2(k-1) + c_{i111} \\ y_{i2}(k+1) &= a_{i211}(k)y_2(k) + a_{i212}(k)y_2(k-1) + b_{i211}(k)u_2(k) \\ &\quad + b_{i212}(k)u_2(k-1) + b_{i221}(k)u_1(k) + b_{i222}(k)u_1(k-1) + c_{i211} \end{aligned}$$

Fig. 1 shows the evolutions of parameters of the first output y_1 for three rules. Fig. 2 shows the evolutions of parameters of the second output y_2 for three rules.

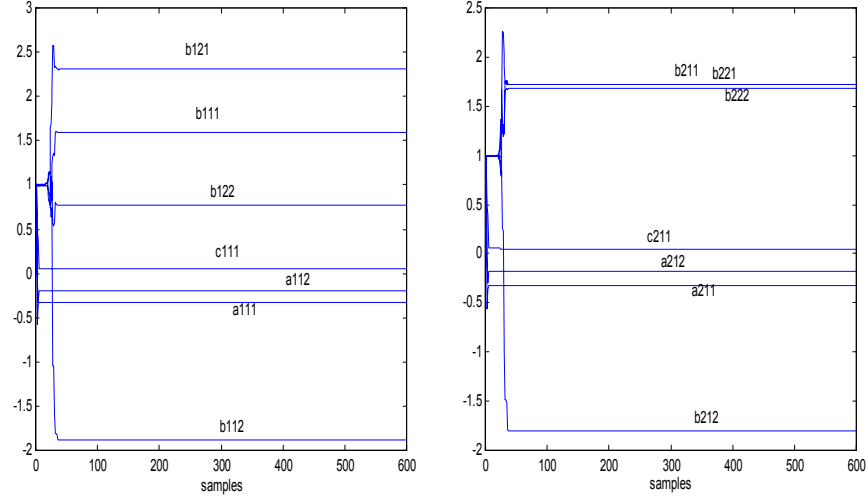
In Figs. 1 and 2 we can notice that the linear parameters vary until the sample 80 and 100, respectively for the output y_1 and y_2 . Afterwards, they are practically constant.

The reference signals $y_{1m}(k)$ and $y_{2m}(k)$ are square waves with period 100.

As shown in Figs. 3 and 4, exact tracking is obtained using the proposed adaptive fuzzy control. The corresponding control laws are presented in Figs. 5 and 6.

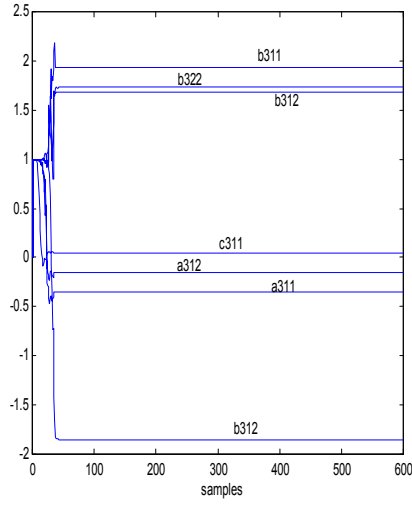
6. Conclusion

The approach presented provides a solution to the problem of robust control of MISO nonlinear systems. For each MISO system a local fuzzy adaptive



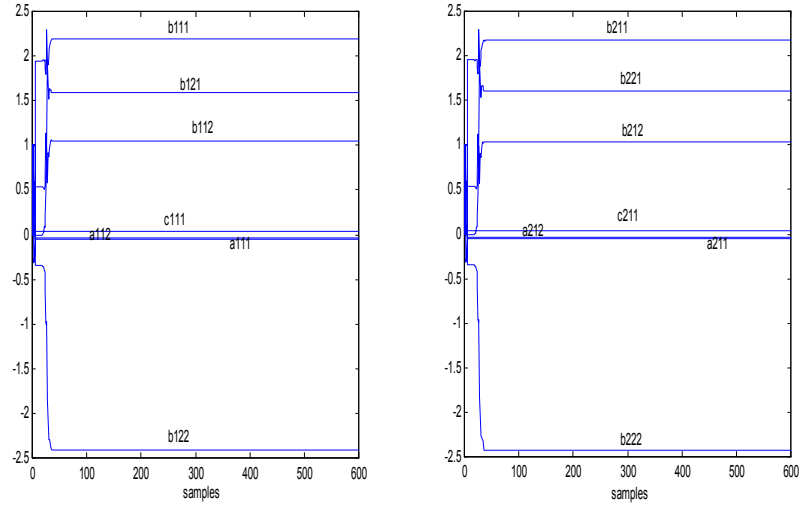
evolution of parameters of rule (1)

evolution of parameters of rule (2)

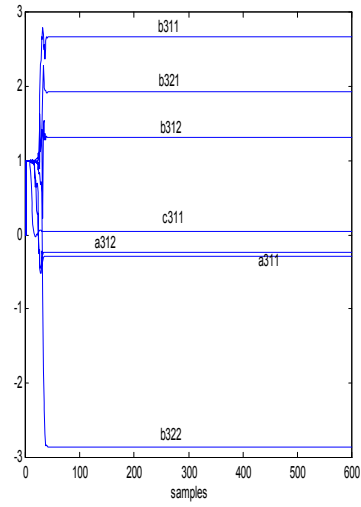


evolution of parameters of rule (3)

Table 1. Evolution of parameters of the first output $y_1(t)$



evolution of parameters of rule (1) evolution of parameters of rule (2)



evolution of parameters of rule (3)

Table 2. Evolution of parameters of the second output $y_2(t)$

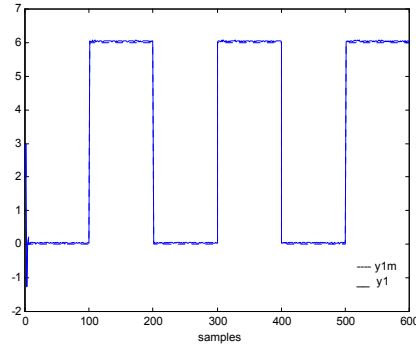


Figure 3. The behaviour of $y_1(k)$ and $y_{1m}(k)$

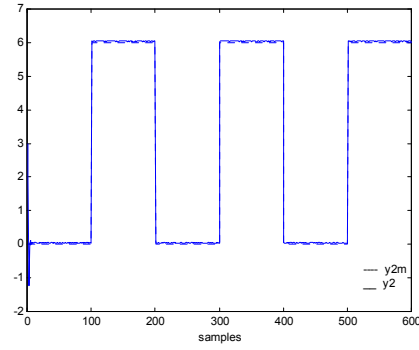


Figure 4. The behaviour of $y_2(k)$ and $y_{2m}(k)$

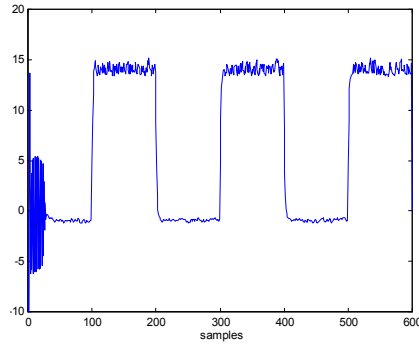


Figure 5. The behaviour of the control $u_1(k)$

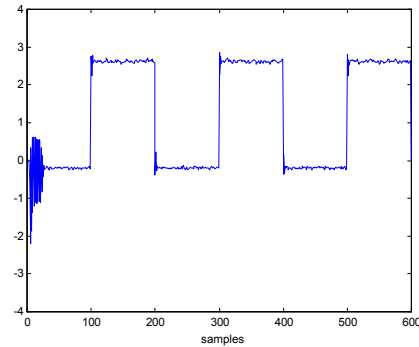


Figure 6. The behaviour of the control $u_2(k)$

control is given. The assumed unknown nonlinearity, coupling between inputs, unmodeled dynamics and disturbances are approximated by the Takagi-Sugeno fuzzy model. The latter allows the conversion of the nonlinear problem into a linear one. This approximation leads to both solving the nonlinearity problem and to releasing the decoupling through the control synthesis. It is shown on a numerical example that the proposed approach is a robust scheme and can deal with a large class of MISO nonlinear systems.

References

- BABUSKA, R. (1998) *Fuzzy Modeling for Control*. Kluwer Academic Publishers, Boston.
- BABUSKA, R., and VEBRUGGEN, H.B. (1995) Identification of composite linear models via fuzzy clustering. In: *Proceedings of European Control*

- Conference 4*, Rome, Italy, 1593-1606.
- BABUSKA, R. and VERBRUGGEN, H.B. (1996) An overview of fuzzy modeling for control. *Control Engineering Practice* **4**, 1593-1606.
- BEZDEK, J.C. (1981) *Pattern Recognition With Fuzzy Objective Function Algorithms*. Plenum Press, New York.
- CHEN, J.Q. and CHEN, J. (1994) An on line identification algorithm for fuzzy systems. *Fuzzy Sets and Systems*, 63-72.
- FENG, G. (1999) Analysis of new algorithm for continuous time robust adaptive control. *IEEE Trans. Automat. Contr.* **44**, 1764-1768.
- FENG, G. and CHEN, G. (2005) Adaptive control of discrete-time chaotic systems: a fuzzy control approach. *Chaos. Solutions and Fractals* **23**, 459-467.
- GATH, I. and GEVA, A.B. (1989) Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **7**, 773-781.
- GLORENNEC, P.Y. (1999) *Algorithmes d'apprentissage pour systèmes d'inférence floue*. Hermes Sciences Publications, Paris.
- GUSTAFSON, D.E. and KESSEL, V.C. (1979) Fuzzy clustering, with fuzzy covariance matrix. In: *Proceedings IEEE. CDC*, San Diego, 761-766.
- HELLENDON, H. and DRIANKOV, D., EDS. (1997) *Fuzzy Model Identification: Selected Approaches*. Springer, Berlin.
- JANG, J.S.R. (1993) ANFIS: Adaptative-network based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics*, **23** (3), 665-685.
- JINXING, Z. and SHIJIUM, L. (1989) Explicit-self-tuning control for a class of nonlinear systems. *Automatica* **25**(4), 593-596.
- JOHANCEN, T.A. and FOSS, B.A. (1993) Constructing NARMAX models using ARMAX models. *International Journal of Control* **58** (5), 1125-1153.
- NARENDRA, K.S. and ANNASWAMY, A.M. (1989) *Stable Adaptive Systems*. Prentice Hall, New Jersey.
- NELLES, O., FINK, A., BABUSKA, R. and SETNES, M. (2000) Comparison of two construction algorithms for Takagi-Sugeno fuzzy models. *International Journal of Applied Mathematics and Computer Science* **10**(4), 835-855.
- NELLES, O., FINK, A. and ISERMANN, R. (2000) Local linear model trees (LOLIMOT) toolbox for nonlinear system identification. 12th *IFAC Symposium on System Identification (SYSID)*, Santa Barbara, USA.
- SUGENO, M. and KANG, G.T. (1987) Structure identification of fuzzy model. *Fuzzy Sets and Systems* **28**, 15-33.
- TAKAGI, T.M. and SUGENO, M. (1985) Fuzzy identification of systems and its application to modelling and control. *IEEE Transactions on Systems, Man and Cybernetics* **15** (1), 116-132.
- TANKA, K. and WANG, H.O. (2001) *Fuzzy Control Systems Design and Analysis. A Linear Matrix Inequality Approach*. John Wiley and Sons, New York.

- TRABELSI, A., LAFONT, F., KAMOUN, M. and ENEA, G. (2004) Identification of nonlinear multivariable systems by adaptive fuzzy Takagi-Sugeno model. *IJCC* **2** (3), 137-153.
- WENG, F. and LANG S. (1990) Globally convergent direct adaptive control algorithm for multivariable systems with general time delay structure. *International Journal of Control* **51**(2), 301-314.

Annexe F

***Optimal T-S Models for Identification of Nonlinear Systems from
Input-Output Data***

H.Ouakka et I.Boumhidi

***ICGST International Journal on Automatic Control and
Systems Engineering, ACSE, Volume 9, Issue I, pp.1-7,
June, 2009.***



Optimal T-S models for identification of nonlinear systems from input-output data

H.Ouakka, I.Boumhidi

L.E.S.S.I, Department of physics, Faculty of Sciences ,Dhar El mahraz

BP 1796 FES Atlas, Morocco

h.ouakka@iam.ma, iboumhidi@hotmail.com

Abstract

Determining optimal structure identification of a nonlinear black box system is one of the most significant steps in Fuzzy modelling based on Takagi-Sugeno models. The one parameter that needs to be determined before performing fuzzy clustering and identification algorithms is the optimal number of clusters. In this paper, we present a new approach for automatically predicting the optimal choice of this parameter simply by detecting the general trend of the data structure. First, an approximation model of the system is built by fitting data to a polynomial function. Second, a preliminary decomposition of the data is realized based on detection of the function turning points. Then, a merging method is adopted to reduce the identified number of clusters. The advantage of the generated solution is that it remains in the horizon of the data; hence there is no need to apply heuristic rules or conventional validation tools. The performance of the proposed method is evaluated for both quality of clustering and fuzzy modeling with 1st order TS systems using GK fuzzy algorithm.

Keywords: *Fuzzy clustering, optimal clusters number, nonlinear system, polynomial regression, Takagi-Sugeno models, GK algorithm.*

1 Introduction

Fuzzy clustering combining with regression analysis are the most tools used to capture sub linear systems in a structure of data, particularly for modelling an unknown black box nonlinear system with Takagi-Sugeno models [3]. There are two main issues in the process of constructing a TS fuzzy model, the first is how to determine the number of rules and the variables involved in the rule antecedents, and the second is how to estimate the parameters of the TS fuzzy model. To construct TS fuzzy model from data, many fuzzy clustering algorithms combined with least

square method have proved to be suitable techniques to partition data space into adequate subspaces and detect linear local models[5].

The Fuzzy C-Means(FCM) and the Gustafson-Kessel(GK) fuzzy clustering algorithms are the most widely used to partition a data set into c homogeneous fuzzy clusters. The FCM detects clusters of a roughly same size. The GK algorithm is an extension of the FCM, which can detect clusters of different orientation and shape in a data set. Both the FCM and the GK algorithm require the number of clusters as an input, and the analysis result can vary greatly depending on the value chosen for this variable. However, in many cases the exact number of clusters in a data set is not known. Both the FCM and the GK algorithm may lead to undesired results if a wrong cluster number is given. The number of clusters needs to be known a priori and the correct number of rules must be estimated in advance[1, 10].

To overcome this problem, many approaches have been proposed: in the first framework, modelling procedure constitutes an iterative process, where the number of rules gradually increases until the model's performance meets a predefined accuracy[7, 12], the second modelling framework is based on using optimal fuzzy clustering, starting with a sufficiently large number of clusters and successively reducing this number by merging clusters that are similar (compatible) with respect to some predefined criteria [11, 13].

In this work we present a new technique to determine the optimal number of clusters, T-S models, that should be used for modelling a given data set of a nonlinear black box system. The basic idea is to fit a polynomial function which follows the general trend in data. The global structure of data space is then recognized by the characteristics points: maxima and minima. Each group of data ranging between maximum and minima, considered a cluster, can be approximated by a linear model. So the total number of cluster, i.e. linear models, can be deduced from the number of turning points. To find the optimal



number, a cluster merging technique is proposed. It consists on fusion of consecutive clusters for which the associated linear model are almost Collinear. The performances of results are examined by using the so called validity clustering criteria and the quality of the approximation is evaluated by comparing fuzzy models obtained by the MATLAB Fuzzy Toolbox and the fuzzy Model identification based on Gustafson-Kessel clustering algorithm.

The basic advantage of the proposed method when compared to other works, is that the optimal number of TS models obtained may be used as an input parameter by any TS identification algorithm to generate an optimal fuzzy model without need to apply heuristic rules or conventional tools to validate the results. In more, the performance of the most indices found in the literature depends on the algorithm used and the structure of the studied system [6, 15].

This paper is organized as follows. Structure of TS fuzzy model is presented in section 2. Section 3 describes the method for construction of the approximation polynomial function, the partitioning method for extraction of the global number of clusters and the merging technique. Section 4 presents simulation results. Section 5 gives some concluding remarks and discuss current and future extensions of this work.

2 Takagi-Sugeno fuzzy models

Consider the identification of an unknown nonlinear Black Box system of the form :

$$y_k = f(x_k)$$

based on specified or measured input data X and measured output data Y of the system, where $k = 1, \dots, N$, denotes the index of the k -th input-output data-pair. In general it may not be easy to find a global nonlinear model that is universally applicable to describe the unknown system $f(\cdot)$. In that case it would certainly be worthwhile to build local linear models for specific region of the data space and combine these into a global model. The relation between input and the output of the process is then described by linguistic if-then rules with fuzzy proposition in the antecedent and crisp functions in the consequents. Hence, it can be seen as a combination of linguistic and mathematical regression modeling in the sense that the antecedents describe fuzzy regions in the input space in which consequent functions are valid. The TS rules have the following form:

$$R_i : \text{if } x \text{ is } A_i \text{ then } y_i = a_i x + b_i \quad i = 1, 2, \dots, c$$

where c is the number of if-then rules, A_i are fuzzy sets, and (a_i, b_i) are the model consequent parameters that have to be identified in a given data set. For a given input crisp vector $x = (x_1, x_2, \dots, x_N)^T$ the inferred global output of the Takagi-Sugeno model is

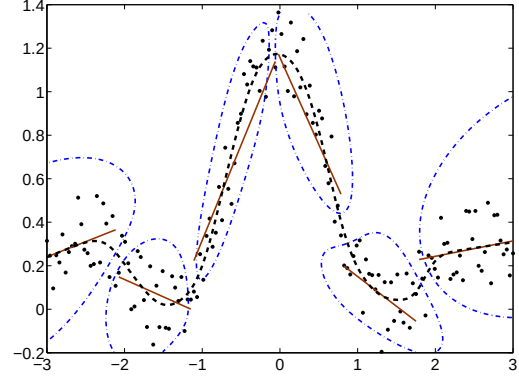


Figure 1: Results of the GK algorithm for fuzzy modelling of a black box nonlinear system : Fuzzy model, TS models and Clusters.

computed by taking the weighted average of the individual rules contributions

$$y = \frac{\sum_{i=1}^c \mu_i(x) \cdot y_i}{\sum_{i=1}^c \mu_i(x)}$$

where $\mu_i(x)$ is the degree of fulfillment of the i^{th} fuzzy rule.

The Gustafson-Kessel (GK) clustering algorithm has often been applied to identify automatically TS models. In general, fuzzy clustering algorithms generate a fuzzy partition given as a fuzzy partition matrix $U = [\mu_{ij}]$, where $\mu_{ij} = \mu_{\tilde{F}_i}(x_j)$ is the membership value of the data x_j belonging to the fuzzy cluster $\tilde{F}_i$. The objective of GK algorithm is to obtain a fuzzy c -partition $\tilde{F} = \{\tilde{F}_1, \dots, \tilde{F}_c\}$ for the given number of clusters c and the given N input-output data (X, Y) by minimizing the evaluation function J_m ,

$$J_m(U, V, A : X) = \sum_{i=1}^c \sum_{j=1}^N (\mu_{ij})^m D_{ijA_i}^2$$

$$D_{ijA_i}^2 = (x_j - v_i)^T A_i (x_j - v_i)$$

where $V = (v_1, \dots, v_c)$ is a vector of the centroids of the fuzzy clusters $(\tilde{F}_1, \dots, \tilde{F}_c)$, m controls the fuzziness of membership of each datum, D is an adaptive norm (Mahalanobis distance) used by the GK algorithm for each cluster to detect different geometrical shapes in data sets. Each i^{th} cluster has its own norm-inducing matrix A_i , where A_i is obtained from fuzzy covariance matrix (F_i) of the i^{th} cluster defined by:

$$F_i = \frac{\sum_{j=1}^N \mu_{ij}^m (x_j - v_i)(x_j - v_i)^T}{\sum_{j=1}^N \mu_{ij}^m}$$

$$A_i = [\rho_i \det(F_i)]^{-1/p} F_i^{-1}$$

with $\rho_i > 0 \quad \forall i$, p is the data space dimension. Without any prior knowledge, the cluster volume ρ_i is simply fixed at 1 for each cluster, which allows the algorithm to find clusters of approximately equal



volumes[2].

It should be noted that for the process of identification of T-S models based on GK fuzzy clustering algorithm, the number of rules coincides with the number of clusters, each cluster is approximated with a local linear model(Figure1).

3 Method formulation

In this section , we present our method in tree steps: identification of the polynomial model, extracting of the global number of clusters from the approximation function and the optimization of T-S models by the merging thechnique (Figure 2) .

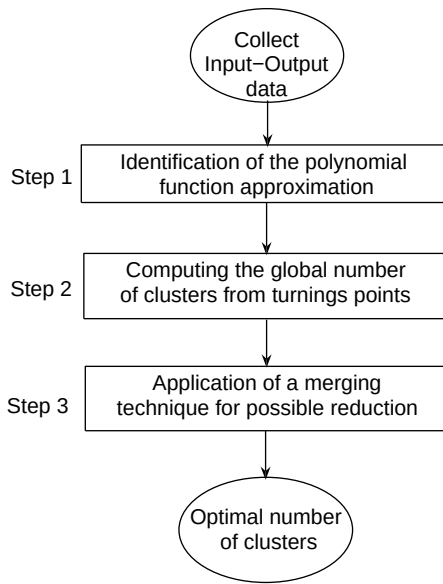


Figure 2: Process for identification of the optimal number of TS models.

3.1 Data model identification

The general problem studied in this section can be stated as follows. On the basis of N input-output samples

$$(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$$

where $x_i \in X$ is the i th input vector and $y_i \in Y$ is the corresponding output, find the model f that best approximates the true underlying function mapping x to y .

In that way, nonlinear regression by polynomial models (using the matlab function polyfit) is used to build a nonlinear function approximation that preserve the general structure of the data . The regression problem stated as a function approximation problem is the following.

From N samples (x_i, y_i) , the polynomial function

that can better predict the output value y on the basis of the regression vector x is given by :

$$y = f(x) = \text{polyfit}(x, y, n). \quad (1)$$

n is the polynomial order or degree.

The function f is defined by:

$$f(x) = a_0 + \sum_{k=1}^n w_k x^k \quad (2)$$

Depending on the degree n , these models can be used to approximate a large class of nonlinear functions. However, the number of parameters w increases with the size of X and the degree n . That clearly reflects that polynomial models suffer from the curse of dimensionality, problem known as Runge phenomenon[4]. In general, the estimation of n can be performed by stepwise regression.

To overcome this problem, an heuristic formula (3), is proposed to compute the adequate polynomial order to build an intermediate mathematical function that describes the overall structure of the data and preserve features of the distribution such as relative maxima, minima and width.

$$n = \sqrt{N \cdot (1 + r^2)} \quad (3)$$

with r = correlation coefficient (X, Y) .

To illustrate this approach, (see figure1), let us consider the following process (4) representing a noisy SISO black box non linear system :

$$y = \text{sinc}(x) + \epsilon \quad (4)$$

ϵ a vector of disturbance.

In this example ,the value of the polynomial degree computed by (3) is $n = 12$, the model is more powerful compared to the others ($n = 5, n = 7$), the overall aspect of the system is guaranteed and the error of modeling is minimized.

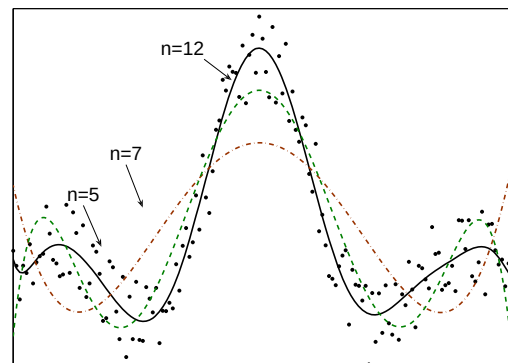


Figure 3: Nonlinear regression by polynomial models



3.2 Extracting clusters from function approximation

Analyzing the behavior of the polynomial function graph (figure 3), we can see that the data structure can be partitioned in seven substructures or clusters. This number is equal to the number of turning points, maximum and minimum, increased by unit. Consequently, we can propose the following result : finding the correct number of cluster involves the determination of the number of turning points of the approximation function described by the polynomial model.

According to the two fundamentals theorems of algebra:

Theorem 1 Any polynomial $P(x) = a_0 + \sum_{k=1}^n w_k x^k$ of degree n , with real coefficients has precisely n zeros in C .

Theorem 2 A polynomial function of degree n , can have at most $(n - 1)$ turning points.

The total number of turning points are the roots of the polynomial function $P(x)$, determined by using the command matlab *polyder*. In our case, the derived polynomial $P'(x)$ of degree $(n - 1)$ has $(n - 1)$ real or complex roots.

Assuming that α is the number of real roots and β the number of complex roots, referring to the first and second theorem, the polynomial degree and roots are bound by the following equation (5):

$$\alpha + \beta = n - 1 \quad (5)$$

Since the system is defined in the data space $(X * Y)$, we are interested thereafter only in the real roots which are included in the data input interval X . Suppose that α^* is the number of total real roots λ_i which define the turning points abscises, such as :

$$\lambda = [\lambda_1, \dots, \lambda_{\alpha^*}] \in X$$

The data input space can be partitioned in (α^*) intervals:

$$X = \underbrace{[x_1, \dots, \lambda_1]}_{X_1} \cup \underbrace{[\lambda_1, \dots, \lambda_2]}_{X_2} \cup \dots \cup \underbrace{[\lambda_{\alpha^*}, \dots, x_N]}_{X_{c^*}}$$

As a result, according to the remark made at the beginning of the paragraph, the number c^* , defined in (6), is the global number of cluster that should be used to partition the data space (7).

$$c^* = \alpha^* + 1 \quad (6)$$

$$X * Y = \bigcup_{i=1}^{c^*} \overbrace{[X_i, Y_i]}^{\text{cluster}} \quad (7)$$

X_i and Y_i are respectively the i th sub input data interval and its corresponding sub output interval.

By applying the least square method for each group of data in a cluster, we can identify for each cluster a local linear model: $y = a_i x + b_i, i = 1, \dots, c^*$ that approximates each sub system as shown in figure 4.

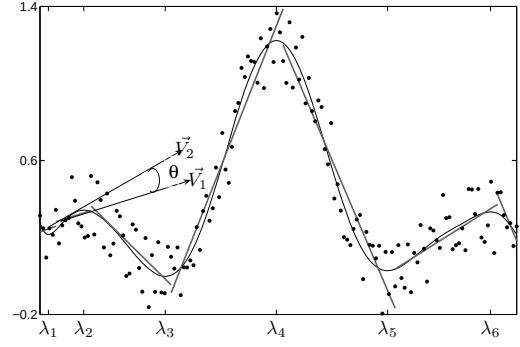


Figure 4: Partition of input data set in 7 subsets based on function approximation turning points.

3.3 Determining the number optimal of cluster

In figure 4, the two first regressed linear models present a strongly correlation. The corresponding groups of data may be merged in one cluster. In that way, for optimizing the number of clusters, we propose a clustering merging criterion based on the fusion of two consecutive line segment i and $i + 1$, given by (8) into a single cluster if their associated vectors are nearly correlated or linear. We assume that each local

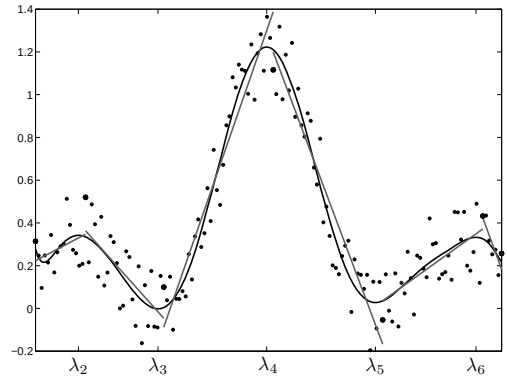


Figure 5: Reduction of the number of local models (clusters) to 6 by the merging technique.

regression model is interpreted as a vector $\vec{V}_i$ defined for $x \in X_i, y \in Y_i$.

$$y = a_i x + b_i, i = 1, \dots, c^*. \quad (8)$$

It is known, trigonometry and geometrical approach, see figure 4, that the co-linearity or correlation of two vectors is measurable by the angle :

$$\theta = \text{angle}(\vec{V}_i, \vec{V}_{i+1}), i = 1, \dots, (c^* - 1)$$

$\vec{V}_i$ is the vector corresponding to the i th local linear model.

In addition there is a relationship between this angle and the correlation of the vectors assumed representing the regressed data in each cluster $\cos \theta = r$.



The angle for which the merging criteria is validated, is taken equal to $\theta_r = 30$, this is equivalent for a correlation coefficient of data equal to $r = 0.82$, considered strongly correlation. The cluster merging algorithm is presented below:

1. Initial parameters

- $k = 0$, *Fusion number counter*.
- $\theta_r = 30$, *Validation merge criterion*.

for $i = 1$ to $(c^* - 1)$

2. Construction of vectors and angle θ .

- $\vec{V}_i \leftarrow y = a_i x + b_i, x \in X_i$.
- $\vec{V}_{i+1} \leftarrow y = a_{i+1} x + b_{i+1}, x \in X_{i+1}$.
- $\theta = \text{angle}(\vec{V}_i, \vec{V}_{i+1})$.

3. Merge criterion control.

- if $\theta \leq \theta_r$.
- *Merging intervals* $X'_i = X_i \cup X_{i+1}$.
- *Resulting local linear model* $y' = a'_i x + b'_i$.
- $k = k + 1$.

The optimal number of clusters corresponding to the optimal partition is then given by:

$$C = c^* - k. \quad (9)$$

The number of clusters determined, in subsection 3.2, for the system (4), is equal to seven clusters; this number is reduced to six by applying the merging method. Figure 5 shows the new partitioning of the X data input, the sub data inputs X_1, X_2 are merged and a new sub local linear model is built to approximate the data in the resulting cluster corresponding to sub input interval $[x_1, \dots, \lambda_2]$:

$$X = \underbrace{[x_1, \dots, \lambda_2]}_{X'_1} \cup \underbrace{[\lambda_2, \dots, \lambda_3]}_{X'_2} \cup \dots \cup \underbrace{[\lambda_6, \dots, x_N]}_{X'_6}$$

X'_1 is the resulting sub input interval from the fusion of X_1 and X_2 .

Comparing the fuzzy model (figure 1) and the polynomial function (figure 3), the two models represent identically the general trend of data, this shows well, the effectiveness of the polynomial function to be used as a basis for partitioning of the data of an unknown nonlinear system.

4 Simulations results

In order to verify the results of the proposed method, we have prepared a set of noisy data generated by the function (10) taken from [9, 8] approximated from 201 points placed in equal intervals in $[-1, 1]$.

$$y = 0.6 \sin(\pi x) + 0.3 \sin(3\pi x) + 0.1 \sin(5\pi x) + \epsilon. \quad (10)$$

ϵ is a distribution added to simulate a black box nonlinear system.

The Gustafson-Kessel (GK) algorithm is applied to cluster these data set at each cluster number c from $c = 4$ to $c = 11$.

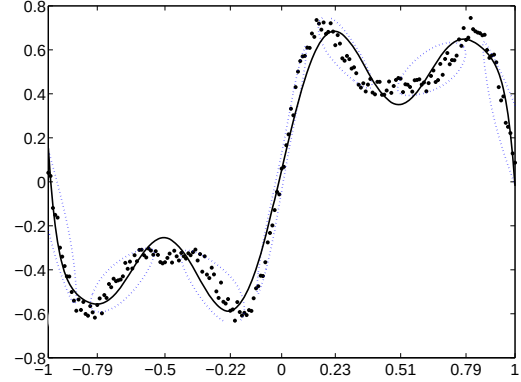


Figure 6: Result of partition of the system (10) data set : polynomial model ($n=11$), Optimal number of clusters ($c=7$).

Two validation approaches will be used to evaluate the results. The quality of clustering is tested using the fuzzy clustering toolbox to compute the so called clustering validity criteria : Partition coefficient (PC) and Classification entropy (CE). Both the two indices measure the amount of overlapping between clusters. The optimal partition can be determined by the point of the extrema of the validation indexes in dependence of the number of clusters. The optimal number of clusters is assumed equal to the local maximum of the (PC) indices and the local minimum for the (CE) [14].

The performance of the fuzzy model is measured by the VAF, which computes the percentile Variance accounted for between two signals as follows:

$$VAF = 100 \cdot \left[1 - \frac{\text{var}(y_1 - y_2)}{\text{var}(y_1)} \right]$$

y_1 is the output of the process and y_2 is the output of the fuzzy model with the same number of clusters obtained by the MATLAB fuzzy toolbox (ANFIS) and the fuzzy model identification (FMID) toolbox based on GK clustering [1]. The VAF of two equal signals is 100. If the signals differ, VAF is lower.

Figure 6 illustrates the result of our proposed method for partitioning the data based on the identified polynomial function. The polynomial degree computed by (3) is $n = 11$, number of turning points detected is 6 and the optimal number of clusters is given by (9) $c = 7$.

The performance of each clustering validity measure is given in figure 7. As seen from the figure, the (PC) and (CE) validity measures find that the optimal cluster number c is at $c = 7$.



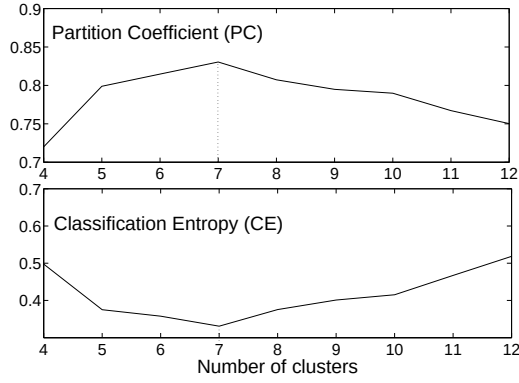


Figure 7: Performance of (PC) and (CE) clustering validation index for different number of clusters.

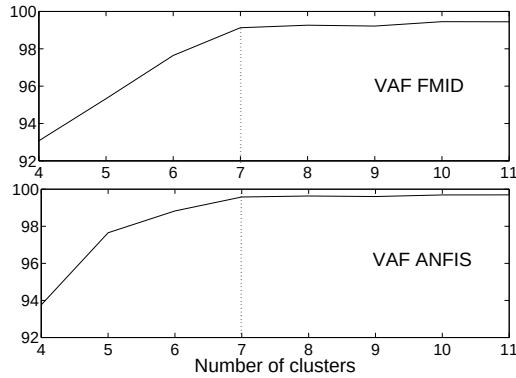


Figure 8: VAF results for different number of clusters.

In figure 8, The VAF predict $c = 7$ as the correct optimal number of clusters since there is no wide variation between the value of the VAF for the number of cluster ranging between 7 and 11. This shows applicability of our method to compute the optimal number of clusters for approximation a nonlinear black box system with a fuzzy model as shown in figure 9.

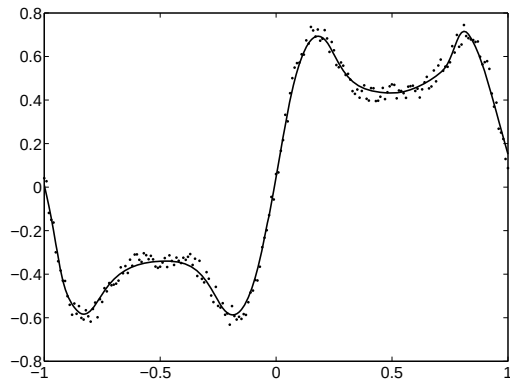


Figure 9: Comparison of the fuzzy model and the original data.

In order to verify the approach for a continuous

nonlinear system, the system (10) is taken without perturbation, our method has computed ($c = 9$) optimal clusters, the algorithm using a complementary interpolation model applied in [9] has validate the same number of interpretable TS models that capture the essence of the process under consideration.

The simulation results are illustrated in table 1.

cluster number	VAF FMID	VAF ANFIS	PC	CE
6	98.208	99.200	0.840	0.309
7	99.613	99.912	0.868	0.264
8	99.705	99.941	0.881	0.234
9	99.821	99.967	0.894	0.207
10	99.760	99.961	0.892	0.212
11	99.974	99.972	0.894	0.207

Table 1: Validation indices for the system (10) without perturbation .

5 Conclusion

The major contribution of this paper is an improved method to compute automatically the optimal number of clusters, needed as input parameter for clustering algorithms, for correctly approximating a nonlinear black box system with TS models. The simulation show that the proposed approach simplifies and makes more effective to extract optimal and legitimate TS local models from data . In addition, the data model identification used in this paper has proved to be suitable for transforming an unknown black box system into a grey box one . The approach can be extended for more complexes systems, since it has proved important result for SISO nonlinear systems.

References

- [1] F. Abonyi, F. Szeifert, and R. Babuska. Modified gath-geva fuzzy clustering for identification of takagi-sugeno fuzzy models. *IEEE Transactions on Systems, Man, and Cybernetics*, 32(5):612–621, 2002.
- [2] R. Babuska. *Fuzzy modeling for control*. Kluwer Academic, 1998.
- [3] P. Bortolet and R. Palm. Identification, modeling and control by means of takagi-sugeno fuzzy systems. In *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*, pages 515–520, Barcelona, Spain, Jul 1997.
- [4] B. Fornberg and J. Zuev. The runge phenomenon and spatially variable shape parameters in rbf interpolation. *Computers and Mathematics with Applications*, 54(3):379–398, 2007.



- [5] E. Gustafson and W.C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In *IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, pages 761–766, San Diego, Jan 1978.
- [6] A.H. Hadad, S.S. Ghidary, S. Bahrami, and S. Shahbazi. A modification of sugeno-yasukawa modeler to improve structure identification phase. *ICGST-ACSE*, 6(3):33–40, 2006.
- [7] D. Kukolj and E. Levi. Identification of complex systems based on neural networks and takagi-sugeno fuzzy model. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(1):272–282, 2004.
- [8] A. Riid, R. Isotamm, and E. Rustern. Transparency analysis of first-order takagi-sugeno systems. In *Proceedings of 10th International Conference on System Modelling Control*, pages 165–170, Zakopane, Poland, 2001.
- [9] A. Riid and E. Rüstern. Takagi-sugeno systems with improved interpretability and interpolation using a complementary interpolation model. In *Proceedings of 4th Intelligent Systems Design and Applications*, pages 259–264, Budapest, 2004.
- [10] S. Sun, S. Wang, and Q. Jiang. Fcm-based model selection algorithms for determining the number of clusters. *Pattern recognition*, 37(10):2027–2037, 2004.
- [11] S. Tafazoli, M. Leduc, and X.S. Bafas. Modeling using fuzzy subtractive clustering. *Computational cognition*, 4(3):15–27, 2006.
- [12] G. Tsekouras. On the use of the weighted fuzzy c-means in fuzzy modelling. *Advances in Engineering Software*, 36(5):287–300, 2005.
- [13] G. Tsekouras, H. Sarimveis, E. Kavakli, and G. Bafas. A hierarchical fuzzy-clustering approach to fuzzy modelling. *Fuzzy Sets and Systems*, 150(2):245–266, 2005.
- [14] J. Valente and W. Pedrycz. *Advanced in Fuzzy Clustering and its Applications*. Jhon Wiley and Sons Ltd, 2007.
- [15] LL.K. Young, W.K. Dae, L. Doheon Lee, and H.L. Kwang. A cluster validation index for gk cluster analysis based on relative degree of sharing. *Informatics and Computer Science*, 168(4):225–242, 2004.

Biography:



Hamid Ouakka, received his Master's degree in 2004 from the University Sidi Mohamed Ben Abdellah, Faculty of Sciences, Fes Morocco. He is pursuing his Ph. D. degree. His research areas include fuzzy control systems, linear and non-Linear systems and Robust Control.



Ismail Boumhidi is a Professor of electronics at the Faculty of Sciences, Fez Morocco. He received from the University of Sidi Mohamed Ben Abdellah, Morocco his "Doctorat de 3ième cycle" degree in 1994 and "Doctorat d'Etat" degree in 1999. His research areas include adaptive robust control, multivariable nonlinear systems, and fuzzy logic control with applications.

