N° d'ordre : 40393

UNIVERSITÉ LILLE 1 SCIENCES ET TECHNOLOGIES
ECOLE NATIONALE D'INGÉNIEURS DE TUNIS

Ecole Doctorale Sciences Pour l'Ingénieur Université Lille Nord-de-France – 072

THÈSE

présentée en vue d'obtenir le grade de

DOCTEUR

Discipline : Automatique et Informatique Industrielle

par

Nesrine ELFELLY

Ingénieur en Informatique

APPROCHE NEURONALE DE LA REPRÉSENTATION ET DE LA COMMANDE MULTIMODÈLES DE PROCESSUS COMPLEXES

dirigée par :

P. BORNE	Professeur à l'Ecole Centrale de Lille
M. BENREJEB	Professeur à l'Ecole Nationale d'Ingénieurs de Tunis
J-Y. DIEULOT	Maître de Conférences à l'Ecole Polytechnique Universitaire de Lille

soutenue le 13 décembre 2010 devant le Jury d'examen :

Président :

N. ELLOUZE	Professeur à l'Ecole Nationale d'Ingénieurs de Tunis

Rapporteurs :

A. EL MOUDNI	Professeur à l'UTBM, Centre de Belfort
N. BENHADJ BRAÏEK	Professeur à l'Ecole Polytechnique de Tunisie

Examinateurs :

F-G. FILIP	Membre de l'Académie Roumaine
S. BELGHITH	Professeur à l'Ecole Nationale d'Ingénieurs de Tunis

Thèse préparée en cotutelle au sein

du Laboratoire d'Automatique, Génie Informatique et Signal (LAGIS, FRE CNRS 3303), EC Lille

et de l'Unité de Recherche LA.R.A. Automatique, ENIT

A l'âme de mon père…

A ma mère…

A mon mari Adnen…

A mon frère et mes sœurs…

A tous ceux qui me sont chers…

Avant-propos

Les travaux présentés dans cette thèse ont été effectués au sein du Laboratoire d'Automatique, Génie Informatique et Signal (LAGIS) de l'Ecole Centrale de Lille et de l'Unité de Recherche LA.R.A. Automatique de l'Ecole Nationale d'Ingénieurs de Tunis (ENIT).

Je suis particulièrement sensible au grand honneur que Monsieur le Professeur Noureddine ELLOUZE, Directeur de l'Unité de Recherche LSTS de l'ENIT, me fait en acceptant de présider mon Jury d'Examen. Qu'il trouve ici l'expression de ma profonde reconnaissance.

C'est un agréable devoir pour moi d'exprimer ma profonde gratitude aux directeurs de cette thèse : Monsieur Pierre BORNE, Professeur à l'Ecole Centrale de Lille, Monsieur le Professeur Mohamed BENREJEB, Directeur de l'Unité de Recherche LA.R.A. Automatique de l'ENIT et Monsieur Jean-Yves DIEULOT, Maître de conférences - HDR à l'Ecole Polytechnique Universitaire de Lille. Je tiens à les remercier et leur manifester ma sincère reconnaissance pour leurs grandes qualités humaines, leur encadrement plein d'enthousiasme et de rigueur et pour la confiance dont ils ont fait preuve à mon égard.

Je tiens à exprimer ma vive gratitude à Monsieur Abdellah EL MOUDNI, Professeur à l'Université de Technologie Belfort-Montbéliard et à Monsieur Naceur BENHADJ BRAÏEK, Professeur à l'Ecole Polytechnique de Tunisie pour l'intérêt qu'ils ont bien voulu porter à mes travaux en acceptant d'être les rapporteurs de cette thèse. Qu'ils trouvent ici, le témoignage de ma profonde reconnaissance.

Je souhaite également témoigner de mes vifs remerciements à Monsieur Florin-Gheorghe FILIP, Vice-Président de l'Académie Roumaine et à Madame Safya BELGHITH, Professeur à l'ENIT d'avoir accepté de participer à mon Jury d'Examen.

Finalement, ces remerciements ne seraient pas complets si je n’y associais pas toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce travail, et tout spécialement, mes collègues pour leur amicale présence, tout le personnel de l’Ecole Centrale de Lille et en particulier du LAGIS et de l’unité de recherche LA.R.A. Automatique pour leur bonne humeur et leur disponibilité.

Table des matières

Liste des figures **11**

Introduction générale **15**

Sur les approches multimodèles **21**

I.1. Introduction 23

I.2. Systèmes et modélisation 24
I.2.1. Systèmes linéaires 26
I.2.2. Systèmes non-linéaires 28
I.2.2.1. Systèmes bilinéaires 28
I.2.2.2. Systèmes non-linéaires affines en la commande 29

I.3. Vers la représentation multimodèle 29
I.3.1. Origine des multimodèles 30
I.3.2. Principe de la représentation multimodèle 32
I.3.3. Classes des systèmes multimodèles 35
I.3.3.1. Système multimodèle à modèle global explicite 36
I.3.3.2. Système multimodèle à modèle global implicite 36

I.4. Méthodes de mise en œuvre des approches multimodèles 37
I.4.1. Méthode basée sur des techniques d'identification 38
I.4.2. Méthode basée sur le principe de linéarisation 38
I.4.3. Méthode basée sur la transformation polytopique convexe 39

I.5. Les différents concepts des approches multimodèles 41
I.5.1. Approche multimodèle par modélisation idéale 41
I.5.2. Approche multimodèle par modélisation locale 42
I.5.3. Approche multimodèle par modélisation générique 44

I.6. Estimation des validités des modèles 47
I.6.1. Définition de la validité 47
I.6.2. Approches de calcul des validités 48
I.6.2.1. Approche directe de calcul des validités 48
I.6.2.2. Approche floue de calcul des validités 49
I.6.2.3. Approche probabiliste de calcul des validités 51
I.6.2.4. Approche géométrique de calcul des validités 52
I.6.2.5. Approche des résidus pour le calcul des validités 54

I.7. Modélisation de processus complexes – Techniques neuro-floues – Position du problème 56
I.7.1. Synthèse des systèmes multimodèles à base de linéarisation 56
I.7.2. Structures multimodèles à base de réseaux de neurones 57
I.7.3. Aide à la conception de structures multimodèles par des méthodes neuro-floues 58

I.8. Conclusion 59

Méthodes de classification pour l'aide à la conception d'une représentation multimodèle de processus complexes 61

II.1. Introduction ----- 63

II.2. Démarche envisagée pour la détermination de la structure multimodèle ----- 64
II.2.1. Procédure de construction de la base de modèles ----- 65
II.2.2. Détermination du nombre de modèles ----- 66
II.2.2.1. Collecte de données et type entrée/sortie du système 67
II.2.2.2. Apprentissage compétitif pénalisant le rival pour la sélection du nombre de modèles 68
II.2.3. Application de techniques de classification pour la détermination des classes de fonctionnement du système ----- 72
II.2.3.1. Réseaux de neurones artificiels et apprentissage 74
II.2.3.2. Classification par les cartes de Kohonen 75
II.2.3.3. Classification par l'algorithme des K-moyennes 80
II.2.3.4. Classification par l'algorithme des K-moyennes floues 83
II.2.3.5. Hypothèse sur la sélection de la méthode de classification 85

II.3. Identification paramétrique des modèles de la base ----- 86
II.3.1. Estimation des ordres des modèles ----- 86
II.3.2. Estimation des paramètres des modèles ----- 88

II.4. Estimation des degrés de pertinence des modèles et validation de la modélisation proposée ----- 89
II.4.1. Méthodes adoptées pour le calcul des validités ----- 90
II.4.1.1. Résidus et validités 90
II.4.1.2. Première méthode de calcul : validités simples 92
II.4.1.3. Deuxième méthode de calcul : validités renforcées 93
II.4.2. Hypothèse sur le choix de la méthode de calcul des validités ----- 94
II.4.3. Test et validation de la structure multimodèle proposée ----- 96

II.5. Exemples de modélisation de processus complexes ----- 97
II.5.1. Exemple 1 : Système discret du second ordre avec des paramètres dépendant du temps ----- 97
II.5.2. Exemple 2 : Système du second ordre avec des paramètres variant en fonction de l'entrée et de la sortie ----- 105
II.5.3. Exemple 3 : Modélisation d'un système mécanique : bras manipulateur --- 113
II.5.4. Exemple 4 : Modélisation d'un réacteur biologique ----- 118

II.6. Interprétations des résultats de la mise en œuvre de l'approche proposée ---------- 121

II.7. Conclusion ----- 125

Commande multimodèle de processus complexes 127

III.1. Introduction ----- 129

III.2. Principe de la commande multimodèle ----- 130

III.3. Exemples de systèmes de commande ----- 132
III.3.1. Régulateur PID ----- 133
III.3.2. Commande à modèle interne ----- 134
III.3.3. Commande par retour d'état ----- 134
III.3.4. Commande de type RST ----- 134

III.4. Stratégies de la commande multimodèle ---------- 135
III.4.1. Stratégie de commande multimodèle par commutation ---------- 135
III.4.2. Stratégie de commande multimodèle par fusion ---------- 136
III.4.2.1. Fusion des commandes partielles 137
III.4.2.2. Fusion des paramètres des commandes 138

III.5. Description de l'approche envisagée pour la commande multimodèle de processus complexes---------- 139
III.5.1. Calcul des commandes partielles par la loi de régulation R-S-T---------- 140
III.5.2. Calcul de la commande globale du système ---------- 143

III.6. Mise en œuvre sur des exemples de systèmes complexes ---------- 145
III.6.1. Exemple 1 : Commande d'un système discret du second ordre avec des paramètres dépendant du temps ---------- 145
III.6.2. Exemple 2 : Commande d'un système discret non linéaire---------- 154
III.6.3. Interprétations des résultats---------- 160

III.7. Conclusion ---------- 161

Conclusion générale 163

Bibliographie..... 165

Annexe A. Approche floue de type Takagi-Sugeno..... 175

Annexe B. Méthodes de renforcement des validités..... 177

Liste des figures

Figure I.1. Système « boîte noire » 32
Figure I.2. Représentation du système par un seul modèle global 33
Figure I.3. Modélisation du système par une approche multimodèle 33
Figure I.4. Principe de la commutation multimodèle 34
Figure I.5. Principe de la fusion multimodèle 35
Figure I.6. Système multimodèle à modèle global explicite 36
Figure I.7. Système multimodèle à modèle global implicite 37
Figure I.8. Domaines de validité disjoints 43
Figure I.9. Chevauchement des domaines de validité 43
Figure I.10. Distance géométrique 52
Figure I.11. Modélisation de la distance du système par rapport aux modèles locaux 53
Figure II.1. Exemple d'une carte de Kohonen à structure unidimensionnelle 77
Figure II.2. Exemple d'une carte de Kohonen à structure bidimensionnelle 77
Figure II.3. Principe de l'évaluation de la structure multimodèle 96
Figure II.4. Exemple 1 : Signal d'excitation 98
Figure II.5. Exemple 1 : Détermination du nombre de classes 99
Figure II.6. Exemple 1 : Résultats de la classification (carte de Kohonen) 99
Figure II.7. Exemple 1 : Erreurs absolues entre la sortie réelle et les sorties multimodèles (sélection de la méthode de classification) 101
Figure II.8. Exemple 1 : Sorties réelle et multimodèle (validités simples) 103
Figure II.9. Exemple 1 : Sorties réelle et multimodèle (validités renforcées) 103
Figure II.10. Exemple 1 : Erreurs absolues entre la sortie réelle et les sorties multimodèles (sélection de la méthode de calcul des validités) 104
Figure II.11. Exemple 1 : Sorties réelle et multimodèle 105
Figure II.12. Exemple 2 : Signal d'excitation 106
Figure II.13. Exemple 2 : Détermination du nombre de classes 107
Figure II.14. Exemple 2 : Résultats de la classification (K-moyennes floues) 108
Figure II.15. Exemple 2 : Erreurs absolues entre la sortie réelle et les sorties multimodèles (sélection de la méthode de classification) 109
Figure II.16. Exemple 2 : Sorties réelle et multimodèles (validités simples/renforcées) 110

Figure II.17. Exemple 2 : Erreurs absolues entre la sortie réelle et les sorties multimodèles (sélection de la méthode de calcul des validités) 111
Figure II.18. Exemple 2 : Sorties réelle et multimodèle 112
Figure II.19. Exemple 2 : Comparaison entre l'approche multimodèle proposée et l'approche classique de modélisation 113
Figure II.20. Exemple 3 : Système mécanique (bras manipulateur) 114
Figure II.21. Exemple 3 : Détermination du nombre de classes 115
Figure II.22. Exemple 3 : Résultats de la classification (K-moyennes) 116
Figure II.23. Exemple 3 : Sorties réelle et multimodèle (séquence d'entrée u_1) 117
Figure II.24. Exemple 3 : Sorties réelle et multimodèle (séquence d'entrée u_2) 118
Figure II.25. Exemple 4 : Sorties réelle et multimodèle (séquence d'entrée u_1) 120
Figure II.26. Exemple 4 : Séquence d'entrée u_2 120
Figure II.27. Exemple 4 : Sorties réelle et multimodèle (séquence d'entrée u_2) 121
Figure II.28. Organigramme résumant les étapes d'implémentation de l'approche proposée pour la représentation multimodèle 124
Figure III.1. Principe de la commande multimodèle 131
Figure III.2. Principe de la régulation en boucle fermée 133
Figure III.3. Principe de la stratégie de commande par commutation 136
Figure III.4. Principe de la stratégie de commande par fusion des commandes partielles 137
Figure III.5. Principe de la stratégie de commande multimodèle par fusion des paramètres des commandes partielles 138
Figure III.6. Structure du régulateur R-S-T associé au modèle M_i 141
Figure III.7. Exemple 1 : Evolutions des commandes partielles (consigne 1) 147
Figure III.8. Exemple 1 : Evolutions de la consigne 1 et de la réponse du système (fusion des commandes / validités simples) 148
Figure III.9. Exemple 1 : Evolutions de la consigne 1 et de la réponse du système (fusion des commandes / validités renforcées) 148
Figure III.10. Exemple 1 : Evolutions de la consigne 1 et de la réponse du système (fusion des paramètres des commandes / validités simples) 150
Figure III.11. Exemple 1 : Evolutions de la consigne 1 et de la réponse du système (fusion des paramètres des commandes / validités renforcées) 150
Figure III.12. Exemple 1 : Erreurs absolues entre la consigne 1 et les réponses du système obtenues par les deux méthodes de fusion 152

Figure III.13. Exemple 1 : Evolutions de la consigne 2 et de la réponse du système (fusion des commandes / validités renforcées) 153
Figure III.14. Exemple 1 : Evolutions de la consigne 2 et de la réponse du système (fusion des paramètres des commandes / validités renforcées)..... 153
Figure III.15. Exemple 1 : Erreurs absolues entre la consigne 2 et les réponses du système obtenues par les deux méthodes de fusion 154
Figure III.16. Exemple 2 : Sorties réelle et multimodèle (séquence d'entrée u_1)..... 156
Figure III.17. Exemple 2 : Séquence d'entrée u_2..... 156
Figure III.18. Exemple 2 : Sorties réelle et multimodèle (séquence d'entrée u_2)..... 157
Figure III.19. Exemple 2 : Erreurs absolues entre la consigne 1 et les réponses du système obtenues par les deux méthodes de fusion 159
Figure III.20. Exemple 2 : Evolutions de la consigne 1 et de la réponse du système (Méthode de fusion des paramètres des commandes)..... 159
Figure III.21. Exemple 2 : Evolutions de la consigne 2 et de la réponse du système (Méthode de fusion des paramètres des commandes)..... 160

Introduction générale

Dans la majorité des cas, en automatique, avant de procéder à toute étude, un processus doit tout d'abord être modélisé. De ce fait, la modélisation s'avère une étape indispensable et fondamentale dans la mesure où elle pourrait fortement conditionner la réussite de l'étude ou l'analyse du processus considéré. Plusieurs objectifs peuvent motiver la modélisation d'un système, parmi lesquels on peut citer par exemple l'analyse des interactions et des phénomènes comportementaux afin d'appréhender ou améliorer la connaissance du système, le diagnostic de son fonctionnement et enfin la conception d'une commande permettant d'assurer un comportement désiré.

Pour qu'elle soit efficace et réussie, une modélisation doit permettre, à travers l'établissement de relations liant les variables caractéristiques entre elles, de résumer les aspects essentiels du processus sous une forme accessible et aisément exploitable et de représenter d'une manière rigoureuse le comportement de celui-ci dans les différents domaines de fonctionnement. La finesse de la modélisation dépend de son usage, et, par exemple dans un but de commande, du cahier des charges.

Lorsque l'on considère des systèmes très complexes, faisant intervenir de multiples phénomènes physiques, cette phase de modélisation devient une tâche fastidieuse dans le sens où il est difficile voire même impossible de concevoir, à partir des lois physiques existantes, un modèle simple qui tienne compte de toute cette complexité qui peut être due à une forte non linéarité, une non stationnarité, un large domaine de fonctionnement, des changements dans les paramètres du système ou des perturbations externes. Depuis de nombreuses années, certains chercheurs se sont intéressés à la modélisation et la commande de systèmes non linéaires en mettant en œuvre des structures particulières de modèles non linéaires, à savoir les modèles bilinéaires, les modèles non linéaires affines en la commande et les modèles polynomiaux [Brockett, 76] [Derese et al, 80] [Fliess, 76] [Rotella et al, 88]. D'autres se sont penchés sur les systèmes de grande dimension en définissant des modèles de structure hiérarchisée [Benrejeb, 80] [Borne, 76].

Par ailleurs, le degré de difficulté de conception et de mise en œuvre d'un système de commande dépend de la complexité du ou des modèles générés. Par conséquence, il s'est avéré intéressant de recourir à des approches pratiques de modélisation de systèmes

complexes faisant intervenir des modèles plus simples. Dans ce contexte, plusieurs travaux de recherche ont été proposés. Malgré que les méthodes mises en œuvre soient parfois très différentes, ces travaux partent du même principe qui consiste à remplacer le modèle unique et éventuellement très complexe par un ensemble de modèles plus simples, cette approche étant désignée sous le nom de représentation multimodèle [Baruch et al, 08] [Cho et al, 07] [Delmotte, 97] [Johansen et al, 99] [Kardous Khaldi, 04] [Ksouri-Lahmari et al, 04] [Mezghani, 00] [Takagi et al, 85].

L'approche multimodèle a été donc proposée comme une alternative pour appréhender le comportement non linéaire et complexe d'un système par un ensemble de modèles linéaires ou non, formant ce qu'on appelle « base » ou « bibliothèque » de modèles et caractérisant le fonctionnement du système dans différentes zones de fonctionnement. Malgré le succès de son application dans différents domaines, de nombreux problèmes liés à sa synthèse restent ouverts. On peut citer en particulier la détermination du nombre et des paramètres des différents modèles ainsi que le choix de la méthode adéquate de calcul des degrés de contribution des modèles pour la déduction de la sortie multimodèle et pour la conception d'une commande globale appropriée permettant d'atteindre les performances souhaitées. La différence principale entre les études proposées dans ce domaine se retrouve au niveau du type des modèles et de la méthode adoptée pour l'identification des modèles. Dans la plupart des cas, il s'agit de modèles linéaires, comme par exemple les modèles flous de type Takagi-Sugeno [Takagi et al, 85] [Angelov et al, 04] qui sont généralement obtenus par application de méthodes de linéarisation autour de différents points de fonctionnement et en partant d'un modèle global de référence.

Cependant, la représentation multimodèle d'un système devient plus complexe quand on ne dispose d'aucune information a priori, que ce soit sur le modèle global du processus ou sur ses domaines de fonctionnement. Dans ce cas, on parle d'un système de type « boîte noire » et les modèles de la base doivent être déterminés à partir des mesures de type entrée/sortie qui représentent les seules informations exploitables sur le processus à étudier. La littérature révèle deux grandes classes de structures multimodèles conçues à cet effet et qui sont basées sur des techniques neuro-floues. En effet, certaines études [Manioudakis et al, 01] [Selmic et al, 01] [Vieira et al, 04] proposent de concevoir un seul réseau de neurones en utilisant une représentation directe des dynamiques non linéaires à travers les neurones. D'autres suggèrent de définir plusieurs réseaux de neurones, chacun correspondant à un

domaine de fonctionnement bien particulier [Fu et al, 07] [Yu, 06]. L'inconvénient majeur de ces approches réside dans le fait que le modèle neuronal d'identification n'est pas paramétrique, ce qui ne permet pas d'utiliser l'information obtenue directement lors de la conception de la commande. Dans le cadre de nos travaux, nous nous intéressons à la deuxième classe des multimodèles dont le principe consiste, tout d'abord, à répartir les mesures en classes, calculer le nombre de modèles de la base et ensuite estimer leurs paramètres et domaines de fonctionnement. La dernière étape consiste à établir les fonctions permettant de calculer en ligne les degrés de validité des différents modèles. Les approches élaborées, dans ce contexte, évoquent l'utilisation de différentes techniques de classification neuronales (essentiellement les cartes de Kohonen) [Ben Aoun Talmoudi, 05] [Cho et al, 06] [Kukolj et al, 04] ou neuro-floues [Cheng et al, 06] [Gegúndez et al, 08] [Ronen et al, 02].

Par ailleurs, grâce à leur simplicité et leur capacité de classification, les algorithmes des K-moyennes se sont révélés efficaces pour le regroupement de données. Pour résumer, bien que différentes architectures multimodèles, utilisant les réseaux de neurones, ont été proposées, il n'y a pas eu beaucoup de travaux portant sur l'utilisation des techniques de classification basées sur les réseaux de neurones et les algorithmes des K-moyennes appliquées à la représentation multimodèle classique. Les problèmes les plus pertinents se rapportent à la procédure de classification de données système et l'exploitation de ses résultats pour l'identification des modèles.

Ainsi, nous proposons, dans ce mémoire, une approche pratique pour la représentation multimodèle de processus complexes basée sur diverses techniques de classification (cartes de Kohonen et algorithmes des K-moyennes) en utilisant seulement les données système de type entrée/sortie.

D'une façon générale, l'adoption d'une représentation multimodèle entraîne naturellement la conception d'une commande multimodèle. Cette dernière a suscité un intérêt considérable ces dernières années et a été appliquée dans différents domaines à savoir dans les secteurs de la chimie et la biologie ([Böling et al, 07] [Liu et al, 08] [Özkan et al, 03]) ou ceux de l'ingénierie et de l'industrie ([Baruch et al, 04] [Cho et al, 06] [Cutululis et al, 06]). La procédure consiste à concevoir un contrôleur pour chaque modèle de la base et la commande globale est alors déduite en utilisant les contrôleurs partiels. Dans ce contexte, les études élaborées diffèrent par les lois de commande adoptées pour les modèles et la stratégie de conception du contrôleur global. La littérature révèle deux grandes classes de méthodes de

calcul de la commande globale, soit une commutation entre les contrôleurs partiels [Fu et al, 07], soit une fusion en utilisant les validités des modèles [Sadati et al, 04] [Lorimier et al, 03].

Dans ce mémoire, une commande adaptative multimodèle est proposée. Cette dernière est basée sur une loi de commande de type RST pour les contrôleurs partiels et une stratégie de fusion pour le calcul de la commande globale.

Les travaux présentés dans ce mémoire s'articulent autour de trois chapitres.

Le principe chapitre permet d'introduire la notion de représentation multimodèle. Tout d'abord, une classification des systèmes et des modèles est présentée. La deuxième et la troisième partie sont consacrées à la description du principe des approches multimodèles en passant par les différentes classes des structures multimodèles et les méthodes de leur obtention. Ensuite, nous proposons une classification des concepts des approches multimodèles et nous présentons les différentes approches de calcul des degrés de pertinence des modèles. Dans la dernière partie de ce chapitre, nous posons la problématique en essayant de situer l'approche proposée par rapport aux travaux élaborés dans le domaine et en mettant l'accent sur les apports et les particularités de notre étude.

Le deuxième chapitre est consacré à la description et à la mise en œuvre de l'approche envisagée pour la représentation multimodèle de processus complexes. En partant des mesures collectées sur le processus et décrivant sa dynamique, cette approche consiste tout d'abord à déterminer le nombre de modèles en proposant une nouvelle technique basée sur un apprentissage compétitif pénalisant le rival. Ensuite, l'utilisation d'un algorithme de classification adéquat permet la sélection des différents domaines de fonctionnement à partir desquels les modèles sont déduits en faisant appel à des méthodes classiques d'identification structurelle et paramétrique. Une stratégie de fusion permet de déduire la sortie multimodèle en mettant en œuvre l'une des méthodes de calcul des validités présentées et qui reposent sur l'approche des résidus. A travers deux hypothèses, nous proposons de distinguer deux classes de systèmes complexes et d'adapter en conséquence le traitement au niveau du choix de l'algorithme de classification le plus approprié d'une part et de la méthode de calcul des degrés de pertinence des modèles d'autre part. L'étude de quelques systèmes dynamiques complexes permet de confirmer nos hypothèses et de conclure sur l'intérêt et l'efficacité de l'approche suggérée.

Le troisième et dernier chapitre aborde le problème de la commande des systèmes complexes en se servant de la base de modèles, que l'on appelle « commande multimodèle ».

Différentes stratégies de commande sont alors exposées. Ayant opté pour la stratégie de fusion, nous détaillons la procédure envisagée pour concevoir une commande multimodèle globale. Le chapitre se termine par l'illustration de cette procédure à travers deux exemples de simulation. Les résultats obtenus témoignent de l'efficacité de la commande multimodèle proposée en adoptant la méthode de fusion des paramètres des contrôleurs partiels.

Chapitre I

Sur les approches multimodèles

I.1. Introduction

I.2. Systèmes et modélisation

I.3. Vers la représentation multimodèle

I.4. Méthodes de mise en œuvre des approches multimodèles

I.5. Les différents concepts des approches multimodèles

I.6. Estimation des validités des modèles

I.7. Modélisation de processus complexes – Techniques neuro-floues – Position du problème

I.8. Conclusion

I.1. Introduction

La phase de modélisation constitue la première étape déterminante dans l'analyse et la synthèse de processus industriels. En effet, pour qu'elles soient réussies, l'étude et la commande d'un système reposent généralement sur une bonne modélisation. Cette dernière vise à établir les relations qui lient les variables caractéristiques du processus entre elles et à représenter d'une manière rigoureuse le comportement de ce processus dans l'ensemble de son espace de fonctionnement. Ces relations sont exprimées à travers un ou plusieurs modèles, d'une façon explicite (modèle mathématique classique) ou implicite (les réseaux de neurones par exemple). Une classification des systèmes et des modèles, les plus utilisés, est présentée dans ce chapitre en allant des plus simples tels que les systèmes linéaires aux plus compliqués nécessitant des représentations non-linéaires.

Cependant, la phase de modélisation devient beaucoup plus délicate lorsqu'il s'agit de systèmes complexes, incertains et fortement non-linéaires. Dans ce cas, deux situations sont envisageables, soit le modèle obtenu est très complexe ce qui le rend parfois inexploitable pour la conception de la commande ; soit il est impossible de concevoir un modèle qui tienne compte de toute la complexité du processus étudié à partir des lois physiques existantes. En conséquence, il est devenu nécessaire de recourir à des approches pratiques qui permettent de faire face à cette complexité au niveau de la modélisation et de la commande. C'est dans ce cadre que se situe l'approche multimodèle qui a été proposée comme une alternative pour appréhender le comportement non linéaire et complexe d'un système par un ensemble de modèles décrivant la dynamique du système dans différentes zones de fonctionnement.

Dans ce chapitre, on commence par présenter le principe de cette approche efficace et puissante pour la représentation de processus complexes en passant par les différentes classes des multimodèles et les méthodes de leur mise en œuvre. Ensuite, on propose une classification des concepts adoptés pour la conception de systèmes multimodèles et des approches de calcul des degrés de pertinence des modèles représentant le système. A la fin de ce chapitre, nous allons situer l'approche proposée dans ce mémoire pour la modélisation et la commande de processus complexes par rapport aux travaux déjà élaborés dans le domaine en mettant l'accent sur les apports et les particularités de notre étude.

I.2. Systèmes et modélisation

En Automatique, on définit de façon informelle un système comme étant un ensemble de composants interconnectés entre eux en vue de remplir une tâche bien précise. L'interaction entre les différents composants du système est réalisée à travers un certain nombre de canaux d'entrée et de sortie plus ou moins bien définis, canaux caractérisés par des flux d'énergie, de matière ou d'informations. Pour l'étude et la commande d'un système, il est nécessaire de le représenter au moyen d'un ou de plusieurs modèles, qui eux-mêmes constituent l'ensemble des relations explicatives du fonctionnement du système en question. Cette modélisation n'est qu'une approximation du processus, elle n'est pas unique et ne peut être exacte [Borne et al, 92b].

Un modèle idéal doit posséder trois propriétés fondamentales : la simplicité, la précision et la généralité. La simplicité du modèle implique que ce dernier doit être aisé à comprendre et à interpréter, facile à utiliser et conduire à une implémentation simple quand il est utilisé à des fins de prédiction. La précision implique que le modèle décrit correctement le système et procure des prédictions exactes auxquelles on peut se fier. Enfin, la généralité du modèle lui confère la capacité de prendre en compte un nombre varié de situations différentes [Domlan 06].

Plusieurs types de modèles peuvent être considérés suivant l'objectif visé et la nature des commandes envisagées. D'une façon générale, ces modèles peuvent être groupés en quatre classes [Borne et al, 92b] [Feliot, 97] [Kardous Khaldi, 04].

- Les modèles graphiques à partir desquels on déduit les modèles de type 'équation'. Dans ce cas, on peut évoquer par exemple des graphes de fluence, des schémas fonctionnels, des réseaux de Pétri et des bond-graphs.

- Les modèles linguistiques qui sont des modèles définis par des variables linguistiques déterminées par expertise ou par apprentissage. Ils sont compliqués à établir et conviennent plutôt à la réalisation de commandes de type flou [Johansen, 94] [Murray-Smith et al, 97].

- Les modèles non paramétriques, appelés aussi modèles analytiques, sont des modèles qui reproduisent la liaison entre les entrées et les sorties du système par des relations non explicites, tels que les réseaux de neurones.

- Les modèles paramétriques se présentent sous forme de relations explicites entre les variables qui décrivent le processus. Généralement, il s'agit d'équations différentielles, d'équations récurrentes ou de fonctions de transfert lorsqu'en particulier le système est linéaire.

Selon le type des variables mises en jeu dans l'évolution du processus, deux principales classes de systèmes sont à distinguer : les systèmes discrets et les systèmes continus [Borne et al, 92b]. Les premiers sont des processus qui mettent en jeu des informations qui ne sont prises en compte qu'à des moments précis. En général, ces instants sont espacés d'une durée constante appelée période d'échantillonnage. Un système est continu, par opposition à un système discret, lorsque les variations des grandeurs physiques de sortie, qui le caractérisent, peuvent être définies à chaque instant (leurs évolutions suivent des fonctions continues). On parle aussi dans ce cas de système analogique.

Pour tout processus, trois ensembles de variables peuvent être identifiés : les variables de sortie qui sont accessibles à la mesure, les variables d'entrée agissant sur son comportement telles que les commandes ou les perturbations et les variables d'état au moyen desquelles on exprime l'influence du comportement passé du système sur son évolution future. Les variables d'état sont des grandeurs physiques continues du système (elles doivent être dérivables) et doivent être indépendantes les unes des autres. Elles sont généralement rassemblées dans un même vecteur. La connaissance de toutes les variables à un instant t doit permettre de connaître toutes les valeurs des variables de sortie du système à un instant $t+dt$.

Le comportement d'un processus peut être décrit en fonction de ces différentes variables par un ensemble d'équations, formant ce qu'on appelle représentation d'état. Cette représentation, qui peut être linéaire ou non, doit rendre compte de l'état du système à n'importe quel instant futur si l'on connait les valeurs initiales et l'évolution des entrées. Elle peut être continue ou discrète et n'est pas unique. En effet, un même système peut être décrit avec des variables d'état différentes mais leur nombre est toujours le même. Ce nombre, désigné par la lettre *n*, représente l'ordre du système.

D'une façon générale, la représentation d'état possède la forme suivante :

$$\begin{cases} d\left[x(t)\right] = f\left(x(t), u(t), t\right); x(0) = x_0 \\ y(t) = g\left(x(t), u(t), t\right) \end{cases}; \tag{I.1}$$

où $t \in \mathbb{R}^+$ est la variable temps et $x \in \mathbb{R}^n$, $y \in \mathbb{R}^q$ et $u \in \mathbb{R}^m$ désignent respectivement le vecteur état, le vecteur de sortie et le vecteur d'entrée.

$d\left[x\right]$ désigne l'opérateur de décalage temporel $x(t+1)$ quand il s'agit de systèmes discrets et l'opérateur de dérivation $\dot{x}$ pour les systèmes continus.

La forme des fonctions $f(\cdot)$ et $g(\cdot)$ dépend du type du système étudié. Dans les paragraphes suivants, nous allons présenter les systèmes les plus utilisés et qui peuvent être groupés en deux grandes classes : systèmes linéaires et systèmes non-linéaires. Une autre catégorie particulière de systèmes a suscité l'intérêt de plusieurs chercheurs, il s'agit des systèmes de grande dimension [Benrejeb, 80] [He et al, 03].

I.2.1. Systèmes linéaires

Si les fonctions $f(\cdot)$ et $g(\cdot)$ sont linéaires en l'état et en la commande alors le système (I.1) est dit linéaire. Un système linéaire est un système qui a les propriétés suivantes :

- Principe de superposition : si l'entrée du système se décompose en une somme de plusieurs entrées alors la sortie du système sera la somme des sorties correspondant à chaque entrée séparée.
- Causalité : toutes les valeurs sont nulles avant le début de l'expérience (correspondant à l'application de la première consigne).

La représentation d'état d'un système linéaire peut s'écrire donc de la façon suivante :

$$\begin{cases} d\left[x(t)\right] = A(t)x(t) + B(t)u(t); \\ y(t) = C(t)x(t) + D(t)u(t); \end{cases} \tag{I.2}$$

avec :

$A(t) \in \mathbb{R}^{n \times n}$ est la matrice caractéristique du système en régime libre ;

$B(t) \in \mathbb{R}^{n \times m}$ est la matrice de commande ;

$C(t) \in \mathbb{R}^{q \times n}$ est la matrice d'observation ;

$D(t) \in \mathbb{R}^{q \times m}$ est la matrice de transmission directe.

Dans le cas où ces matrices ne dépendent pas explicitement du temps, le système linéaire est dit stationnaire.

Outre cette représentation temporelle différentielle, le comportement d'un système dynamique linéaire invariant peut être décrit par des représentations fréquentielles telles que les fonctions de transfert pour le cas monovariable, et les matrices de transfert qui en représentent une généralisation (cas multivariable). Il s'agit de représentations externes ne dépendant que de l'influence des entrées sur les sorties [Borne et al, 92a].

Dans le cas des systèmes continus, la fonction de transfert se présente sous forme de transmittance en $s, T(s) \in \mathbb{R}^{q \times m}$, obtenue par transformation de Laplace des équations différentielles liant les entrées aux sorties du processus :

$$T(s) = \frac{Y(s)}{U(s)}; \tag{I.3}$$

où $Y(s)$ et $U(s)$ sont les transformées de Laplace respectivement de $y(t)$ et $u(t)$.

Lorsqu'il s'agit de systèmes discrets, la matrice de transfert est définie par une transmittance en z, $T(z) \in \mathbb{R}^{q \times m}$, obtenue par transformation en z des équations de récurrence liant les variables d'entrée aux variables de sortie à des instants discrets :

$$T(z) = \frac{Y(z)}{U(z)}; \tag{I.4}$$

où $Y(z)$ et $U(z)$ désignant respectivement les transformées en z de $y(t)$ et $u(t)$.

Toutefois, il est possible de passer de la représentation d'état à la représentation différentielle, et vice versa, grâce à la relation reliant la matrice de transfert T ($T(s)$ ou $T(z)$) à sa réalisation (A, B, C, D) :

$$T(op) = D + C(opI - A)^{-1}B; \tag{I.5}$$

où op désigne l'opérateur de Laplace s ou l'opérateur z.

I.2.2. Systèmes non-linéaires

Un système non-linéaire est un système qui ne satisfait pas le principe de superposition. Il s'agit d'un processus qui présente un ou plusieurs aspects non-linéaires, par exemple une saturation. Dans cette classe, on peut distinguer des formes particulières de non-linéarité relativement simples dont on présente deux exemples : les systèmes bilinéaires et les systèmes non-linéaires affines en la commande [Benhadj Braiek, 95] [Kardous Khaldi, 04].

I.2.2.1. Systèmes bilinéaires

Les systèmes bilinéaires sont une classe particulière de systèmes non-linéaires qui permettent de représenter une variété de procédés physiques. De nombreux exemples pratiques incluent des applications dans les domaines nucléaire, thermique, chimique, biologique, immunologique ou encore socio-économique [Mohler, 91]. Cette classe de systèmes peut également être utilisée pour approximer certains systèmes [Mohler et al, 80].

Les systèmes bilinéaires sont caractérisés par une fonction $g(\cdot)$ linéaire en l'état et en l'entrée et une fonction $f(\cdot)$ linéaire séparément en l'état et en l'entrée, donc pouvant présenter des produits de variables d'état par des variables d'entrée.

Dans le cas stationnaire, l'équation d'état d'un tel système s'écrit sous la forme suivante :

$$\begin{cases} d\left[x(t)\right] = A_0(t)x(t) + \sum_{i=1}^{m} u_i(t)\left(A_i(t)x(t) + B_i\right); \\ y(t) = C(t)x(t) + D(t)u(t); \end{cases} \tag{I.6}$$

où $A_i (i = 0,\ldots,m)$, $B_i (i = 1,\ldots,m)$, C et D sont des matrices constantes de dimensions convenables.

Cette structure, rappelant la forme linéaire (I.2) permet d'importantes simplifications dans l'étape de commande. Elle est fréquemment utilisée puisque plusieurs systèmes physiques semblent être présentés sous formes de modèles bilinéaires ou sont approximés par cette structure. Plusieurs études ont porté sur l'étude, l'identification et la commande des systèmes bilinéaires [Chen et al, 00] [Derese et al, 80] [Sasaki et al, 03].

I.2.2.2. Systèmes non-linéaires affines en la commande

La structure des systèmes non-linéaires affines en la commande vient généraliser les systèmes bilinéaires (I.6) et élargir la classe des systèmes non-linéaires apparentés aux structures linéaires (I.2). La présentation de ce type de systèmes découle de l'expression (I.1) en considérant la fonction $f(\cdot)$ comme affine des variables d'entrée.

$$\begin{cases} d[x(t)] = f_1(x(t),t) + f_2(x(t),t)u(t); \\ y(t) = g(x(t),t); \end{cases} \tag{I.7}$$

où $f(\cdot)$ est une fonction vectorielle non-linéaire de l'état et du temps :

$$f_2(\cdot) = \begin{bmatrix} f_{2_1}(\cdot) & f_{2_2}(\cdot) & \cdots & f_{2_m}(\cdot) \end{bmatrix};$$

ce qui permet d'écrire le terme $f_2(x(t),t)u(t)$ sous la forme suivante :

$$f_2(x(t),t)u(t) = \sum_{i=1}^{m} f_{2_i}(x(t),t)u_i(t). \tag{I.8}$$

Dans le contexte de l'étude et la commande de ce type de systèmes, plusieurs travaux ont été élaborés utilisant des méthodes différentes telles que les séries de Volterra [Lesiak et al, 78], la géométrie différentielle et l'algèbre de Lie [Isidori et al, 81], ou encore les séries formelles [Fliess, 76] et plus récemment les méthodes basées sur l'algèbre tensorielle de Kronecker [Benhadj Braiek, 90].

I.3. Vers la représentation multimodèle

Pour l'étude et la commande d'un système physique donné, il faut tout d'abord commencer par élaborer un modèle qui puisse reproduire au mieux son comportement possible dans l'ensemble de son domaine de fonctionnement. La complexité du modèle croit proportionnellement avec la complexité du système. Pour certains systèmes, on peut se contenter des modèles simples linéaires stationnaires pour les présenter. Mais malheureusement, dans la vie réelle, il s'agit très souvent de systèmes assez complexes. Une complexité qui peut se présenter sous différents aspects tels que une forte non-linéarité, une non-stationnarité, un vaste domaine de fonctionnement, des changements dans les paramètres du système ou bien aussi des perturbations externes. Pour ce genre de systèmes, les modèles simples (linéaires stationnaires) s'avèrent insuffisants et le recours aux modèles plus

complexes (non-linéaires et non-stationnaires ou bien aussi de grande dimension) permettent évidemment une meilleure approximation du comportement du processus en question, mais sont en revanche difficiles à mettre en œuvre surtout lorsqu'il s'agit de la conception d'une commande [Benhadj Braiek, 95] [Benrejeb, 80] [Rotella et al, 88]. Il est aussi parfois difficile voire même impossible de concevoir un seul modèle représentatif global qui puisse tenir compte de toute la complexité du système.

Une possibilité pratique est donc de recourir à une combinaison d'approches locales en introduisant la notion de représentation multimodèle. Dans ce contexte, la solution consiste alors soit à étudier localement le système pour définir un modèle pour chaque situation, soit, si un modèle a déjà été défini, à le linéariser autour d'un ou de plusieurs points de fonctionnement. Ainsi, les modèles obtenus sont plus simples que le modèle global initial, ce qui facilite la tâche du calcul de la commande.

Plusieurs chercheurs se sont intéressés à l'approche multimodèle et plusieurs méthodes ont été proposées tenant compte des différents aspects physiques des processus complexes [Johansen et al, 99]. Ainsi, on a vu la naissance des approches LPV (Linéaires à Paramètres Variants), polytopique, floue, etc. [Courties, 99] [Delmotte, 97] [Dubois, 95] [Ksouri-Lahmari et al, 04] [Mezghani, 00] [Murray-Smith et al, 97] [Schorten et al, 99] [Stipanović, 00] [Takagi et al, 85] [Talmoudi et al, 08].

I.3.1. Origine des multimodèles

Depuis plusieurs décennies, la représentation d'état a toujours servi comme outil efficace de modélisation. Dans un premier temps, la tendance était d'utiliser des modèles Linéaires Invariants dans le Temps (LTI). Cette approche a fait ses preuves dans plusieurs domaines dans la mesure où nombre de systèmes non-linéaires, en première approximation et au voisinage d'un point de fonctionnement, sont bien représentés par un modèle LTI. Dans ce contexte, plusieurs outils d'analyse et de synthèse, qu'on peut qualifier d'assez riches, ont été développés. Néanmoins, une telle approche présente un inconvénient lié à son aspect uniquement local. En effet, le modèle linéaire n'est qu'une description locale de la dynamique du système.

Une approche globale basée sur de multiples modèles LTI (linéaires ou affines) autour de différents points de fonctionnement a été élaborée ces dernières années (voir par exemple

[Chadli, 02] [Murray-Smith et al, 97]). L'interpolation de ces modèles locaux à l'aide de fonctions d'activation normalisées permet de modéliser le système global non-linéaire. Cette approche, dite multimodèle, s'inspire des modèles flous de type Takagi-Sugeno (T-S) (Annexe A), qui sont en fait un cas particulier des représentations multimodèles. En effet, un multimodèle réalise une partition floue de l'espace caractéristique dit aussi espace de décision (c'est l'espace caractérisé par l'ensemble des variables caractéristiques (de décision) qui peuvent être des variables d'état mesurables et/ou la commande). Les zones de fonctionnement sont définies en termes de propositions sur les variables de prémisse [Takagi et al, 85]. Cette représentation est connue pour ses propriétés d'approximateur universel [Buckley, 93].

Dans la littérature, de nombreux types de modèles flous ont été proposés. Cependant on peut distinguer deux classes principales de modèles flous : le modèle flou de Mamdani [Mamdani et al, 75] et le modèle flou de T-S. La principale différence entre ces deux modèles réside dans la partie conséquence. Le modèle flou de Mamdani utilise des sous ensembles flous dans la partie conséquence alors que le modèle flou de T-S utilise des fonctions (linéaires ou non-linéaires) des variables mesurables. Dans le modèle T-S, la partie conséquence est un modèle linéaire (représentation d'état, modèle autorégressif). Afin d'exploiter la théorie très riche des modèles LTI et notamment des propriétés de stabilité émises sous forme d'inégalités matricielles linéaires [Feng, 06], le modèle de T-S dont la partie conséquence est un modèle linéaire en représentation d'état est de loin le plus utilisé en analyse et commande.

C'est en partant des modèles T-S que plusieurs chercheurs se sont intéressés à l'approche multimodèle en essayant de la faire évoluer et l'adapter suivant les types des processus à étudier. Ainsi, ou peut citer par exemple les méthodes d'identification multimodèle qui ont été développées pour les systèmes complexes incertains à paramètres bornés [Kardous Khaldi, 04] [Ksouri-Lahmari et al, 04]. Une extension pour les systèmes discrets incertains a été proposée par Mezghani [Mezghani, 00]. Le cas des systèmes multivariables a été abordé par Raissi [Raissi et al, 01]. Dans le même contexte, d'autres études (voir par exemple [Leith et al, 99] et [Schorten et al, 99]) supposent la représentation de systèmes complexes par un ensemble de modèles locaux définis à partir d'un modèle global de référence et des méthodes de linéarisation.

I.3.2. Principe de la représentation multimodèle

L'approche multimodèle suppose la définition d'un ensemble de modèles constituant la base de modèles. Chacun de ces modèles représente le processus dans un domaine particulier de son espace de fonctionnement. Ainsi, au lieu de considérer un modèle global qui couvre différentes situations et probablement plus complexe, une alternative consiste à utiliser des modèles locaux pour chaque situation et une loi de passage d'un modèle à un autre. De cette façon, on obtient un modèle beaucoup plus simple à manipuler tout en gardant la capacité de fournir des prédictions suffisamment précises dans le domaine d'activation de chaque modèle. Aussi, chacun de ces modèles ne constitue pas une représentation fidèle du processus, et en général il est même faux, sauf exceptionnellement lorsque le système est régi par le comportement local correspondant [Delmotte, 97].

L'approche multimodèle a donc pour objectif de permettre la réduction de la complexité d'un système en l'étudiant sous certaines conditions particulières, entraînant une simplicité dans la conception de la commande. Pour sa mise en œuvre, deux cas de figure peuvent se présenter :

- soit il s'agit d'un système « boîte noire » où l'on ne dispose que des mesures entrée/sortie suite à la difficulté ou l'impossibilité d'élaborer un modèle mathématique qui puisse reproduire le système dans son espace de fonctionnement. Dans ce cas, l'approche multimodèle s'avère efficace et puissante pour pallier aux difficultés relatives à la modélisation de ce genre de systèmes complexes ;

- soit on dispose d'un modèle mathématique non-linéaire (modèle de connaissance) que l'on exploite pour déduire la base de modèles.

Les figures I.1 à I.3 illustrent la modélisation d'un système par une approche multimodèle pour les deux cas précédemment présentés.

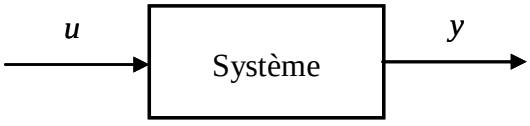


Figure I.1. Système « boîte noire »

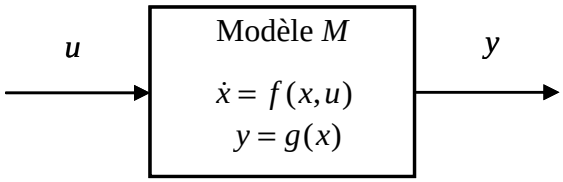


Figure I.2. Représentation du système par un seul modèle global

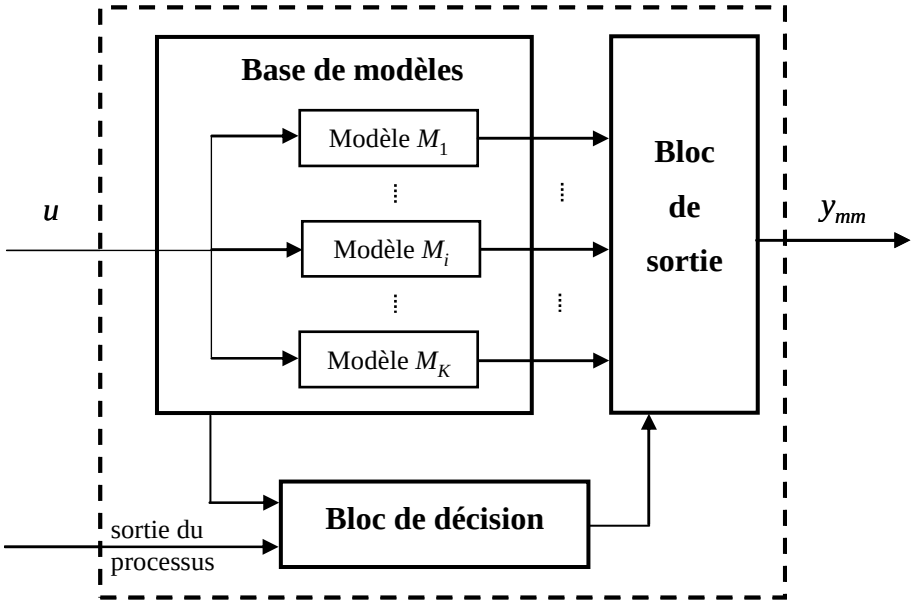


Figure I.3. Modélisation du système par une approche multimodèle

Sur la figure I.3, on distingue trois principaux blocs qu'on présentera dans les prochains paragraphes.

En adoptant l'approche multimodèle, le schéma est radicalement différent, et sa structure apparaît nettement plus complexe. En effet, comme son nom l'indique, un multimodèle fait appel à plusieurs modèles, comme le montre la figure I.3 (*K* modèles). Ces modèles constituent ce qu'on appelle « base » ou « bibliothèque » de modèles. Ces modèles peuvent être locaux ou génériques (paragraphe I.5), de même structure ou de structures et d'ordres différents. Chaque élément de la base est une représentation simplifiée du système global et ne peut reproduire le comportement du système que dans un ou quelques domaines de fonctionnement bien particuliers, d'où le rôle du « bloc de décision ».

Partant du principe qu'au aucun modèle ne constitue une représentation fidèle du système dans l'ensemble de son espace de fonctionnement, il est donc nécessaire de quantifier à chaque instant cette fidélité, il s'agit d'un « mécanisme de décision ». Ce mécanisme permet alors d'estimer la pertinence appelée aussi validité de chaque modèle en fonction d'un certain nombre de paramètres. Les entrées du bloc de décision peuvent être de natures très diverses, comme des mesures d'entrée et/ou de sotie du système, certaines variables internes des modèles ou encore des paramètres environnementaux ou contextuels.

A partir des sorties de la base de modèles et de celles du bloc de décision (validités v_i ; $i = 1, \ldots, K$), le bloc de sortie a pour fonction de combiner toutes ces informations afin de calculer la sortie multimodèle à chaque instant. Cette combinaison peut induire une simple sélection à chaque instant du modèle le plus pertinent (cas de la commutation : figure I.4) ou bien une fusion des paramètres ou des sorties partielles, issus de la base des modèles, pondérés par les validités (cas de la fusion : figure I.5).

- **Multimodèle : cas de la commutation**

A un instant donné, un seul modèle (celui qui possède le degré de pertinence le plus élevé) représente le comportement du processus dans le domaine de fonctionnement en question. Le multimodèle est donc équivalent à un seul modèle de la base.

La sortie multimodèle est donc égale à la sortie d'un seul modèle M_j :

$$y_{mm} = y_j \quad j = 1, \ldots, K. \tag{I.9}$$

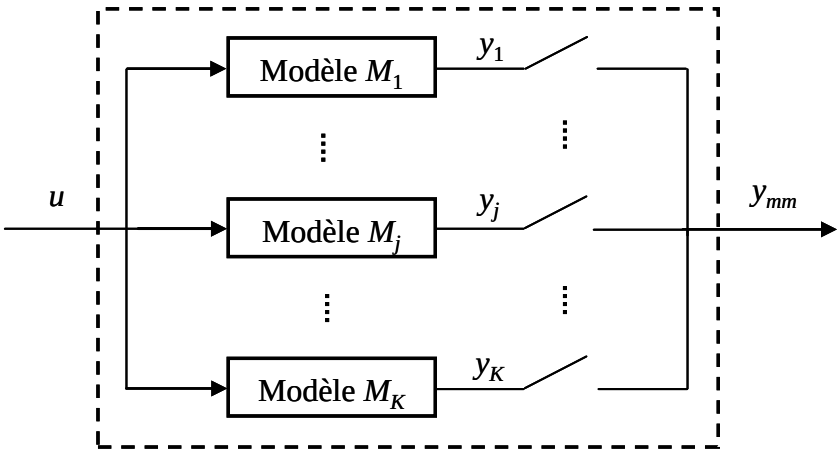


Figure I.4. Principe de la commutation multimodèle

L'avantage principal de cette méthode est la simplicité au niveau du mécanisme de décision. Par contre, elle peut nécessiter la présence de nombreux modèles dans la base surtout lorsqu'il s'agit d'un processus assez complexe et la règle de commutation devient ainsi difficile à déterminer.

- **Multimodèle : cas de la fusion**

Dans ce cas de figure, la sortie multimodèle est obtenue à travers une fusion des paramètres (f_i, g_i) ou des sorties y_i des différents modèles M_i pondérés par leurs validités respectives v_i ($i = 1, \ldots, K$).

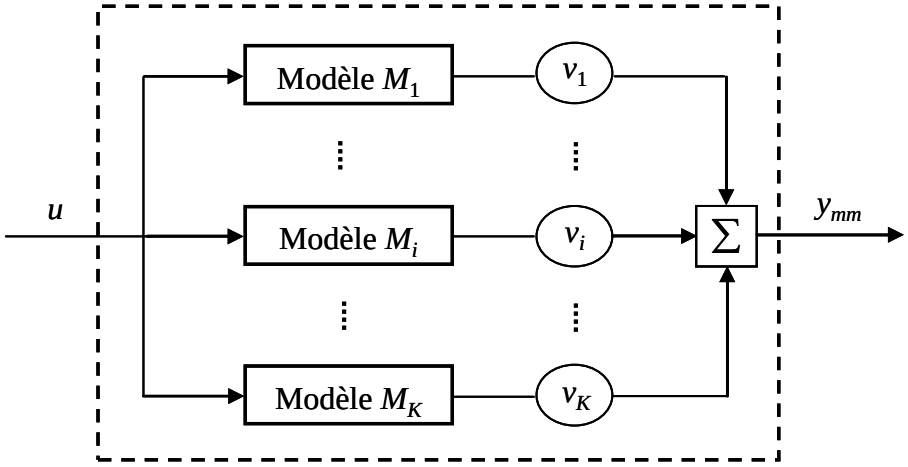


Figure I.5. Principe de la fusion multimodèle

Lorsqu'il s'agit d'une fusion, deux cas de figure se présentent : une fusion entre les paramètres des modèles (f_i, g_i) traduisant un modèle global explicite ou bien une fusion entre les sorties partielles y_i ($i = 1, \ldots, K$) des modèles induisant un modèle global implicite.

I.3.3. Classes des systèmes multimodèles

Dans le système multimodèle, la structure du bloc de sortie ainsi que la nature de ses entrées dépendent de la stratégie de fusion adoptée. Dans la littérature, deux approches de fusion existent donnant lieu à deux classes des systèmes multimodèles : système à modèle global explicite et système à modèle global implicite.

I.3.3.1. Système multimodèle à modèle global explicite

Pour ce type de système multimodèle, la fusion est effectuée au niveau des paramètres des modèles. Le modèle global, traduit par (f, g), peut alors être exprimé à travers les expressions mathématiques des différents couples de fonctions (f_i, g_i) et les validités v_i correspondantes ($i = 1, \ldots, K$) (figure I.6).

Cette classe de multimodèle apparaît intéressante pour étudier des systèmes complexes puisqu'elle permet de construire des modèles explicites et offre la possibilité d'étudier certaines propriétés. Cependant, elle présente une contrainte très forte pour les structures des modèles de la base qui doivent être identiques, ce qui limite le choix des modèles, conduisant ainsi à moins de liberté et de souplesse par rapport à la classe des multimodèles à modèle global implicite.

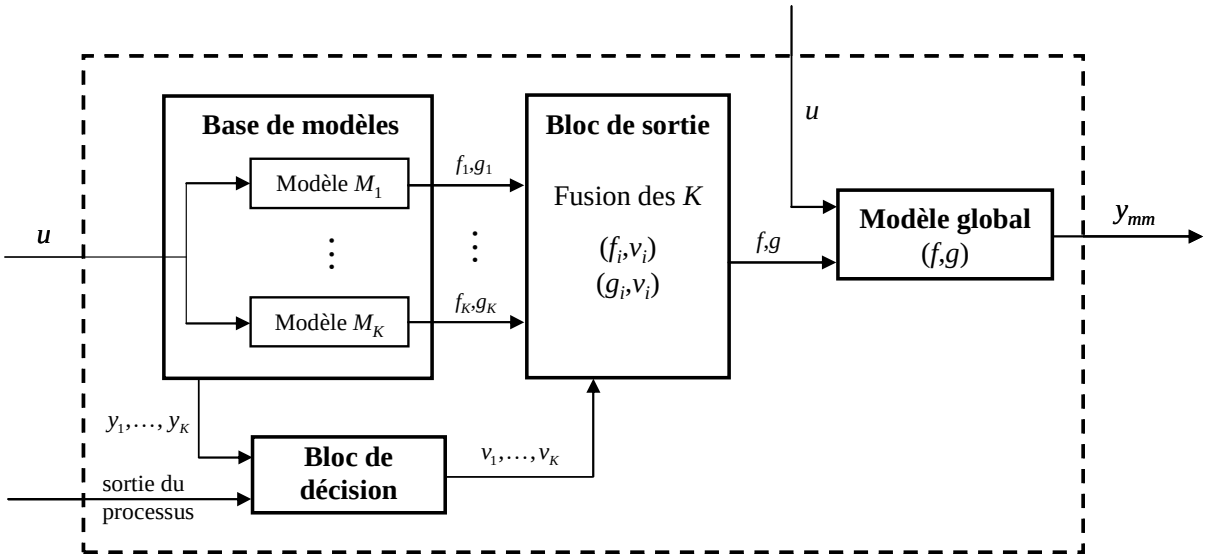


Figure I.6. Système multimodèle à modèle global explicite

I.3.3.2. Système multimodèle à modèle global implicite

Contrairement au modèle global explicite, le système multimodèle ne permet pas de déduire une relation explicite entre l'entrée u et la sortie y_{mm} du système. En effet, la sortie multimodèle y_{mm} est calculée, à chaque instant, en effectuant une fusion entre les sorties partielles y_i des modèles pondérées par les validités v_i ($i = 1, \ldots, K$) (figure I.7). Cette classe des multimodèles offre la possibilité de choisir des modèles de différentes structures.

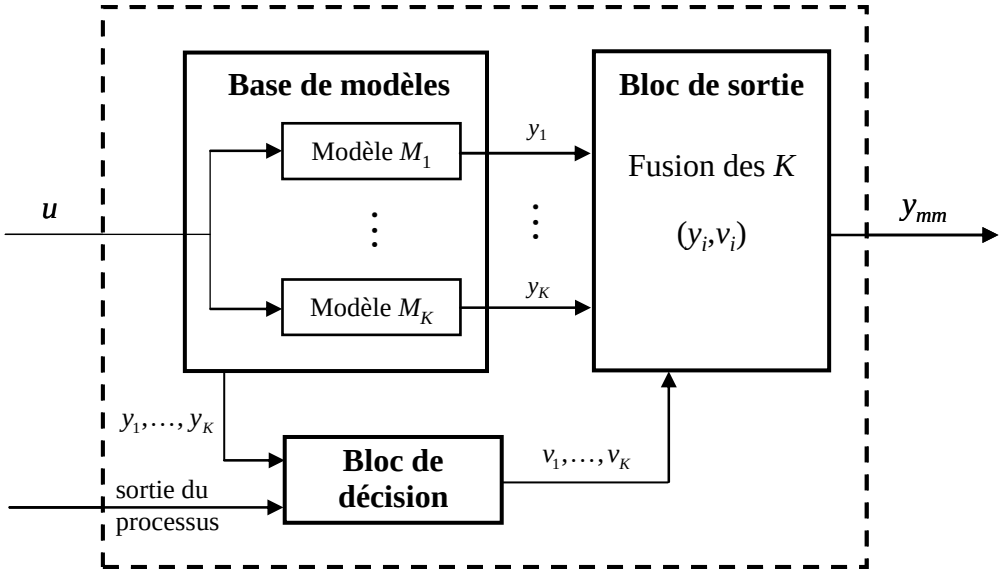


Figure I.7. Système multimodèle à modèle global implicite

I.4. Méthodes de mise en œuvre des approches multimodèles

D'une façon générale, l'approche multimodèle permet de représenter les systèmes complexes non-linéaires sous forme d'une interpolation entre des modèles en général linéaires (ou affines) locaux. Chacun de ces modèles est un système dynamique LTI (Linéaire à Temps Invariant) valide autour d'un certain point de fonctionnement. Dans la littérature, on parle principalement de trois méthodes de détermination de la base des modèles [Chadli, 02]. La première est basée seulement sur les mesures entrée/sortie du système à partir desquelles les différents paramètres des modèles sont estimés. Pour la deuxième et la troisième méthode, on suppose disposer d'un modèle mathématique non-linéaire (modèle de référence) et les modèles locaux sont obtenus soit par linéarisation autour de différents points de fonctionnement (présence de la constante de linéarisation), soit par transformation polytopique convexe [Blanco, 01] [Tanaka et al, 98]. Il est à noter que chacune des trois méthodes, qu'on va présenter, peut s'appliquer aussi bien au cas continu qu'au cas discret.

I.4.1. Méthode basée sur des techniques d'identification

Les systèmes complexes concernés par cette méthode sont des systèmes de type « boîte noire » c'est-à-dire qu'on dispose seulement des mesures entrée/sortie comme informations sur le système. A partir de ces données, les modèles sont identifiés autour de différents points de fonctionnement. Indépendamment du type du multimodèle (modèles linéaires, réseaux de neurones, etc.) choisi, cette identification requiert la recherche d'une structure optimale, l'estimation des paramètres et la validation de la structure multimodèle ainsi obtenue [Elfelly et al, 09] [Gasso, 00]. Dans la plupart des cas, on adopte la représentation d'état pour les différents modèles de la base, ce qui facilite la synthèse des lois de commande (ou d'observateurs) et permet ainsi d'étendre au cas non-linéaire des techniques de commande par retour d'état.

Dans ce contexte, plusieurs travaux, portant sur la représentation et la commande multimodèles des systèmes de type « boîte noire », ont été entrepris (voir par exemple [Baruch et al, 08] [Cho et al, 07] [Principe et al, 98] [Talmoudi et al, 08]).

I.4.2. Méthode basée sur le principe de linéarisation

Dans ce cas, on suppose que l'on dispose d'un modèle mathématique non-linéaire de référence, appelé aussi modèle de connaissance, du processus physique. Le principe consiste à procéder à une linéarisation du modèle global autour de certain points de fonctionnement [Delmotte, 97] [Leith et al, 99]. On considère le système non-linéaire suivant :

$$\dot{x}(t) = f\left(x(t), u(t), t\right); \tag{I.10}$$

où x est le vecteur d'état ($x \in \mathbb{R}^n$) et u le vecteur d'entrée ($u \in \mathbb{R}^m$).

La fonction $f(\cdot)$ est supposée continûment dérivable. La linéarisation de celle-ci autour d'un point de fonctionnement arbitraire $\left(x_i, u_i\right) \in \mathbb{R}^n \times \mathbb{R}^m$ est :

$$\dot{x}(t) = A_i\left(x(t) - x_i\right) + B_i\left(u(t) - u_i\right) + f\left(x_i, u_i\right); \tag{I.11}$$

que l'on peut réécrire sous la forme :

$$\dot{x}(t) = A_i x(t) + B_i u(t) + r_i; \tag{I.12}$$

sachant que :

$$\begin{cases} A_i = \left.\dfrac{\partial f(x,u)}{\partial x}\right|_{\substack{x=x_i \\ u=u_i}} \\ B_i = \left.\dfrac{\partial f(x,u)}{\partial x}\right|_{\substack{x=x_i \\ u=u_i}} \\ r_i = f(x_i,u_i) - A_i x_i - B_i u_i \end{cases} ; \qquad \text{(I.13)}$$

A_i et B_i désignent les jacobiennes de $f(x,u)$ respectivement par rapport à x et u ; r_i représente la constante de linéarisation.

En supposant que les modèles locaux sont issus d'une linéarisation autour de K points de fonctionnement (x_i, u_i), la formulation multimodèle s'écrit alors comme suit :

$$\dot{x}(t) = \sum_{i=1}^{K} v_i(t)\left(A_i x(t) + B_i u(t) + r_i\right); \qquad \text{(I.14)}$$

où les v_i représentent les degrés de pertinence ou validités des modèles calculées à chaque instant.

Pour cette méthode de détermination de la base de modèles, le nombre de ces derniers dépend de la précision de modélisation souhaitée ainsi que de la complexité du système non-linéaire à étudier, notamment l'étendue de son domaine de fonctionnement.

I.4.3. Méthode basée sur la transformation polytopique convexe

Cette méthode s'applique pour les processus décrits par un modèle non-linéaire affine en la commande. Pour illustrer le principe de la transformation polytopique convexe, nous allons considérer la cas général d'un système non-linéaire affine en la commande :

$$\begin{cases} d[x] = f\left(x(t)\right) + B\left(x(t)\right)u(t); \\ y(t) = g\left(x(t)\right) + D\left(x(t)\right)u(t); \end{cases} \qquad \text{(I.15)}$$

où $x \in \mathbb{R}^n$, $y \in \mathbb{R}^q$, $u \in \mathbb{R}^m$, $f(x) \in \mathbb{R}^n$, $g(x) \in \mathbb{R}^q$, $B(x) \in \mathbb{R}^{n\times m}$, $D(x) \in \mathbb{R}^{q\times m}$ et $d[x]$ désigne $x(t+1)$ pour les systèmes échantillonnés et $\dot{x}(t)$ pour les systèmes continus.

La méthode est basée sur une transformation polytopique convexe de fonctions scalaires origine de la non-linéarité. L'avantage d'une telle méthode est de ne pas engendrer d'erreur d'approximation et de réduire le nombre de modèles par rapport à la méthode de linéarisation.

Parmi les techniques basées sur cette approche, nous allons présenter celle qui s'appuie sur la bornitude des termes non-linéaires [Blanco, 01].

On commence tout d'abord par rappeler le principe de la bornitude des fonctions continues. Soit $v(\cdot)$ une fonction bornée de $[-a,b]$ vers $\mathbb{R}$ ($(a,b)\in(\mathbb{R}^+\times\mathbb{R}^+)$).

Pour tout $x(t)\in[a,b]$, il existe alors deux fonctions v^1 et v^2 et deux scalaires α et β tels que :

$$\begin{cases} v^i(\cdot):[-a,b]\to[0,1],\ i=1,2 \\ \qquad x(t)\mapsto v^i\left(x(t)\right) \\ v\left(x(t)\right)=\alpha v^1\left(x(t)\right)+\beta v^2\left(x(t)\right) \\ v^1\left(x(t)\right)+v^2\left(x(t)\right)=1 \end{cases} \tag{I.16}$$

Une décomposition évidente consiste à considérer :

$$\begin{cases} \alpha=\min\limits_{x\in[a,b]}\left(v\left(x(t)\right)\right);\ \beta=\max\limits_{x\in[a,b]}\left(v\left(x(t)\right)\right); \\ v^1\left(x(t)\right)=\dfrac{\beta-v\left(x(t)\right)}{\beta-\alpha};\ v^2\left(x(t)\right)=\dfrac{v\left(x(t)\right)-\alpha}{\beta-\alpha}. \end{cases} \tag{I.17}$$

Les fonctions $f(\cdot)$ et $g(\cdot)$ du système (I.15) étant continues et bornées sur un domaine $U\subset\mathbb{R}^n$ avec des conditions initiales nulles ($f(0)=0$ et $g(0)=0$), peuvent se mettre sous la forme suivante :

$$\begin{cases} f\left(x(t)\right)=A\left(x(t)\right)x(t); \\ g\left(x(t)\right)=C\left(x(t)\right)x(t). \end{cases} \tag{I.18}$$

Le système (I.15) peut alors s'écrire comme suit :

$$\begin{pmatrix} d[x] \\ y(t) \end{pmatrix}=\begin{pmatrix} A\left(x(t)\right) & B\left(x(t)\right) \\ C\left(x(t)\right) & D\left(x(t)\right) \end{pmatrix}\begin{pmatrix} x(t) \\ u(t) \end{pmatrix}. \tag{I.19}$$

Posons :

$$E\left(x(t)\right)=\begin{pmatrix} A\left(x(t)\right) & B\left(x(t)\right) \\ C\left(x(t)\right) & D\left(x(t)\right) \end{pmatrix}. \tag{I.20}$$

La fonction matricielle $E\left(x(t)\right)$ est continue et bornée. Partant du principe de bornitude (I.16) et faisant appel par exemple aux transformations du type (I.17), il est possible de borner

chaque terme non constant de la matrice $E(x(t)) = \{e_{ij}\}$ par des paires de scalaires $(\alpha_{ij}, \beta_{ij})$ et en exprimer des paires de coefficients $\left(v_{ij}^1(x(t)), v_{ij}^2(x(t))\right)$.

Ainsi, la matrice $E(x(t))$ peut s'écrire sous la forme suivante :

$$E(x(t)) = \sum_{k=1}^{K} v_k(x(t)) E_k = \sum_{k=1}^{K} v_k(x(t)) \begin{pmatrix} A_k & B_k \\ C_k & D_k \end{pmatrix}; \qquad \text{(I.21)}$$

où v_k désigne la fonction d'activation (ou de pondération) du modèle M_k.

Le nombre K de modèles locaux ainsi obtenus dépend du nombre de transformations effectuées, donc du nombre de termes non linéaires [Kardous Khaldi, 04].

Cette méthode d'obtention du multimodèle, comparée à la méthode par linéarisation, présente l'avantage de réduire le nombre de modèles, ce qui pourrait être intéressant surtout lorsqu'il s'agit d'un système ayant un large du domaine de fonctionnement (plusieurs points de fonctionnement). Cet avantage pourrait être ressenti lors du calcul de la commande. Par exemple, dans le contexte de la synthèse de régulateurs par analyse convexe [Boyd et al, 94], le nombre de contraintes LMI (Inégalités Matricielles Linéaires) dépend du nombre de modèles locaux. La réduction de ce nombre, dépendant de la méthode de transformation, est synonyme de moins de conservatisme (une condition est dite conservative lorsqu'elle est trop contraignante ou pessimiste par rapport au problème considéré).

I.5. Les différents concepts des approches multimodèles

Suivant la quantité et le type d'information disponible sur le système à étudier, on distingue trois concepts des approches multimodèles : modélisation idéale, locale ou générique.

I.5.1. Approche multimodèle par modélisation idéale

Dans le cas où on dispose de suffisamment d'informations a priori sur le système, une modélisation idéale est envisagée permettant ainsi d'obtenir des modèles précis. A chaque instant, un seul modèle est valide et son degré de pertinence est égal à 1 alors que les validités de tous les autres modèles sont nulles. Par conséquent, une commutation entre les modèles de la base s'avère suffisante.

Pour chaque domaine de fonctionnement, un seul modèle est capable de reproduire idéalement le comportement du processus étudié. La séquence de commutation entre les modèles peut être déterminée a priori. Pour ce type de modélisation, il s'agit d'une approche directe pour le calcul des validités (paragraphe I.6.2.1).

Malgré la simplicité de ce type de modélisation, il est tout de même important de noter qu'elle nécessite une bonne expertise permettant d'acquérir suffisamment de connaissances sur le système à étudier. D'autre part, elle peut induire un nombre important de modèles surtout s'il s'agit d'un processus avec un espace de fonctionnement étendu.

I.5.2. Approche multimodèle par modélisation locale

A la différence de la modélisation idéale, ce type de modélisation ne nécessite pas assez de connaissances a priori sur le système à étudier. Comme son nom l'indique, ce type de modélisation donne lieu à des modèles locaux où chacun représente le processus au voisinage d'un point de fonctionnement particulier. Si on dispose d'un modèle de référence du système complexe, l'ensemble des modèles locaux peut être déterminé en procédant par linéarisation du modèle global au voisinage de certains points de fonctionnement. Si aucun modèle n'est disponible, des méthodes d'identification peuvent être mises en œuvre en considérant à chaque fois un sous-ensemble de données relatives à un domaine de fonctionnement particulier.

Comme la modélisation idéale, la modélisation locale est connue pour sa simplicité aussi bien au niveau de la construction des modèles qu'au niveau du calcul de la commande globale. En effet, si on considère un système complexe ayant un large espace de fonctionnement, il est intéressant de construire différents modèles locaux, chacun spécifique à un domaine de fonctionnement particulier.

La modélisation locale se base sur une décomposition de l'espace global de fonctionnement du système en un certain nombre de domaines de fonctionnement qui sont eux-mêmes les domaines de validités.

Ainsi, à chaque modèle M_i de la base, est attribué un domaine de fonctionnement ou de validité D_i. Les domaines de validité D_i peuvent être disjoints, comme l'illustre la figure I.8.

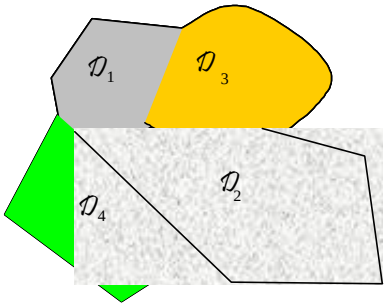


Figure I.8. Domaines de validité disjoints

A un instant donné, un seul modèle est valable dont la validité est égale à 1, toutes les autres sont nulles. Ce type de situation est fréquent dans le cas de systèmes à configurations multiples, ou à plusieurs modes de fonctionnement.

Un autre cas de figure peut se présenter où les domaines de validités se chevauchent comme le montre la figure I.9.

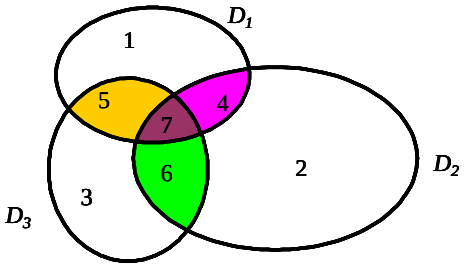


Figure I.9. Chevauchement des domaines de validité

Dans ce cas, plusieurs situations sont envisageables : soit un seul modèle est valable (régions 1, 2 et 3), soit deux modèles ou plus sont partiellement valables (régions 4, 5, 6 et 7), [Ksouri-Lahmari, 99].

Par opposition à la modélisation idéale, la commutation n'est en général pas la bonne approche à adopter pour le calcul de la sortie et/ou de la commande multimodèle même dans le cas où les domaines de fonctionnement ou de validité sont disjoints. En effet, il a été démontré [Ksouri-Lahmari, 99] que le recours à la stratégie de commutation pour le calcul de la commande multimodèle pourrait provoquer des à-coups indésirables au niveau de la

commande, et qu'en général elle est moins efficace que la stratégie de fusion, notamment lorsque les domaines de validité des modèles locaux se chevauchent [Delmotte, 97]. D'une façon générale, il est donc plus intéressant de faire appel à la stratégie de fusion lorsqu'il s'agit de modèles locaux [Narendra et al, 95] [Narendra et al, 97].

Pour la modélisation locale, trois approches de calcul des validités sont envisageables : les approches floue, probabiliste et géométrique (paragraphes I.6.2.2, I.6.2.3 et I.6.2.4).

La simplicité de la modélisation locale n'empêche pas celle-ci de présenter deux limitations. La première concerne le nombre de modèles qui risque d'être assez grand surtout lorsqu'il s'agit d'un système à plusieurs régimes de fonctionnement, ce qui peut rendre pénible et délicate la tâche de conception de la commande multimodèle. La deuxième limitation pourrait être ressentie dans le cas on ne dispose pas des connaissances suffisantes pour pouvoir déduire tous les modèles locaux. Pour remédier à ces limitations, un autre concept de multimodèle a été proposé, il s'agit de la modélisation générique.

I.5.3. Approche multimodèle par modélisation générique

En la comparant aux concepts de modélisation précédemment décrits, la modélisation générique est la moins exigeante au niveau de la quantité d'information disponible sur le processus à étudier. Les modèles engendrés par cette méthode ne correspondent pas nécessairement à des domaines de fonctionnement/validité connus à priori ou prédéterminés. Ainsi, la séquence de commutation entre les modèles n'est pas connue a priori. Par conséquent, l'approche de fusion pour le calcul de la sortie et de la commande multimodèles doit être adoptée sachant que les validités des modèles sont déterminées en ligne par l'approche des résidus (paragraphe I.6.2.5).

Grâce à sa souplesse et sa flexibilité, la modélisation générique a été adoptée dans plusieurs travaux de recherche [Elfelly et al, 10a] [Kardous Khaldi, 04] [Ksouri-Lahmari, 99] [Mezghani, 00]. Par conséquent, plusieurs méthodes de construction de la base de modèles ont été proposées. La plupart de ces méthodes concernent des processus complexes incertains à paramètres bornés et dont on connait l'intervalle de variation. Ces méthodes utilisent généralement les valeurs extrêmes des paramètres et se basent sur l'approche algébrique de Kharitonov [Chapellat et al, 89] pour la détermination des modèles de la base dont le nombre est limité à quatre (qui correspondent aux modèles extrêmes) ou cinq (avec l'ajout d'un

modèle moyen) [Ksouri-Lahmari, 99] [Mezghani, 00]. Dans ce qui suit, on présente l'approche de Kharitonov.

- **Approche algébrique de Kharitonov**

Cette approche est fondée sur le théorème de Kharitonov [Kharitonov, 78]. A l'origine, ce théorème a été établi pour l'étude du signe des racines de polynômes à coefficients incertains et bornés. Ainsi, il n'est pas nécessaire d'appliquer le critère de Routh-Hurwitz à l'ensemble (infini) de polynômes, il suffit de considérer quatre polynômes particuliers.

Soit le polynôme caractéristique : $P(p) = \det(pI - A)$

$$P(p) = a_0 + a_1 p + a_2 p^2 + \ldots + a_{n-1} p^{n-1} + a_n p^n; \tag{I.22}$$

avec $\underline{a_i} \le a_i \le \overline{a_i}\ (i = 1, \ldots, n); \underline{a_i} = \min_i(a_i)$ et $\overline{a_i} = \max_i(a_i)$.

Kharitonov [Kharitonov, 78] a démontré que $P(p)$ a ses racines à partie réelle négative $\forall a_i \in \left[\underline{a_i}, \overline{a_i}\right]$ et $\forall i$; si et seulement si :

$$\begin{cases} P_1(p) = \underline{a_0} + \underline{a_1} p + \overline{a_2} p^2 + \overline{a_3} p^3 + \underline{a_4} p^4 + \ldots \\ P_2(p) = \underline{a_0} + \overline{a_1} p + \overline{a_2} p^2 + \underline{a_3} p^3 + \underline{a_4} p^4 + \ldots \\ P_3(p) = \overline{a_0} + \overline{a_1} p + \underline{a_2} p^2 + \underline{a_3} p^3 + \overline{a_4} p^4 + \ldots \\ P_4(p) = \overline{a_0} + \underline{a_1} p + \underline{a_2} p^2 + \overline{a_3} p^3 + \overline{a_4} p^4 + \ldots \end{cases} \tag{I.23}$$

P_1, P_2, P_3 et P_4 sont tous des polynômes de type Hurwitz (toutes les racines à partie réelle négative).

On considère le système linéaire stationnaire suivant, représenté sous forme canonique compagne :

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ -a_0 & -a_1 & \cdots & -a_{n-2} & -a_{n-1} \end{pmatrix} x; \tag{I.24}$$

avec $\underline{a_i} \le a_i \le \overline{a_i} (i = 1, \ldots, n)$.

Le polynôme caractéristique est donné par l'expression (I.22). L'étude de stabilité de ce système revient à étudier la stabilité des quatre sous-systèmes suivants :

$$\dot{x}_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots \\ \vdots & 0 & 1 & 0 & \\ & \vdots & 0 & 1 & \ddots \\ & & \vdots & 0 & \ddots \\ & & & \vdots & \ddots \\ 0 & 0 & 0 & 0 & \cdots \\ -\underline{a_0} & -\underline{a_1} & -\overline{a_2} & -\overline{a_3} & \cdots \end{pmatrix} x_1 ; \tag{I.25}$$

$$\dot{x}_2 = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots \\ \vdots & 0 & 1 & 0 & \\ & \vdots & 0 & 1 & \ddots \\ & & \vdots & 0 & \ddots \\ & & & \vdots & \ddots \\ 0 & 0 & 0 & 0 & \cdots \\ -\underline{a_0} & -\overline{a_1} & -\overline{a_2} & -\underline{a_3} & \cdots \end{pmatrix} x_2 ; \tag{I.26}$$

$$\dot{x}_3 = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots \\ \vdots & 0 & 1 & 0 & \\ & \vdots & 0 & 1 & \ddots \\ & & \vdots & 0 & \ddots \\ & & & \vdots & \ddots \\ 0 & 0 & 0 & 0 & \cdots \\ -\overline{a_0} & -\overline{a_1} & -\underline{a_2} & -\underline{a_3} & \cdots \end{pmatrix} x_3 ; \tag{I.27}$$

$$\dot{x}_4 = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots \\ \vdots & 0 & 1 & 0 & \\ & \vdots & 0 & 1 & \ddots \\ & & \vdots & 0 & \ddots \\ & & & \vdots & \ddots \\ 0 & 0 & 0 & 0 & \cdots \\ -\overline{a_0} & -\underline{a_1} & -\underline{a_2} & -\overline{a_3} & \cdots \end{pmatrix} x_4 ; \tag{I.28}$$

En s'inspirant de cette approche, pour des systèmes complexes à paramètres incertains et/ou variables et dont on connait les valeurs extrêmes, il a été montré que l'approche algébrique de Kharitonov constitue un bon outil pour construire la base de modèles. Il s'agit de quatre modèles extrêmes qui peuvent être continus [Ksouri-Lahmari, 99] ou discrets [Mezghani, 00], exprimés par des fonctions de transfert ou par des représentations d'état.

Certaines études [Kardous Khaldi, 04] [Ksouri-Lahmari, 99] ont démontré l'intérêt de l'ajout d'un cinquième modèle générique dans la base, appelé modèle moyen et qui représente en quelque sorte le barycentre des modèles extrêmes.

I.6. Estimation des validités des modèles

Le mécanisme de décision, responsable de l'estimation des validités ou des degrés de pertinence v_i des modèles, joue un rôle fondamental dans la conception des systèmes multimodèles. En effet, c'est à grâce aux validités qu'on estime la contribution de chaque modèle de la base dans la reproduction du comportement du processus global dans un domaine de fonctionnement donné. Ainsi, le mécanisme de décision permet de savoir à chaque instant à quel degré un modèle est correct et dans quelle mesure il faudra l'utiliser pour le modèle global, en fonction de l'utilisation souhaitée (représentation ou commande multimodèle). Dans les paragraphes suivants, nous allons présenter les différentes approches de calcul des validités.

I.6.1. Définition de la validité

Dans la représentation multimodèle, il est indispensable de quantifier, à chaque instant, le degré de fiabilité, appelé validité, de chaque modèle. La validité v_i d'un modèle M_i est un réel compris entre 0 et 1. Lorsqu'elle est égale à 0, le modèle correspondant est considéré comme inactif et par conséquent n'a aucune influence sur le système multimodèle global. Si au contraire cette validité prend la valeur 1, le modèle représente parfaitement le processus à l'instant considéré. Ainsi, ce modèle, jugé comme parfait ou idéal, représente lui seul le système global. Ces deux valeurs extrêmes ne peuvent être atteintes que dans le cas d'une commutation. En effet, à chaque instant un seul modèle M_j est valide ($v_j = 1$) et les autres modèles M_i sont considérés comme faux ($v_i = 0; i = 1,\ldots,K$ et $i \neq j$). Sinon, dans le cas d'une fusion entre les modèles ou les commandes, ces valeurs sont rarement atteintes. En effet, aucun modèle ne peut pratiquement être idéal pour représenter parfaitement le processus.

Partant du principe qu'un modèle idéal a une validité égale à 1, une propriété fondamentale découle naturellement, il s'agit de la somme convexe des validités :

$$\begin{cases} v_i \in [0,1] \quad i = 1,\ldots,K \\ \sum_{i=1}^{K} v_i = 1 \end{cases} \tag{I.29}$$

I.6.2. Approches de calcul des validités

La méthode de calcul de validité dépend généralement de la façon dont les modèles ont été déterminés ainsi que des connaissances disponibles sur le système [Delmotte, 97]. Quand il s'agit de connaissances idéales sur le système, une approche directe de calcul des validités est envisageable. Par contre, lorsque les connaissances a priori sont partielles et s'agissant d'une modélisation locale (paragraphe I.5.2), les validités des modèles peuvent être estimées en utilisant l'un des trois formalismes : flou, probabiliste ou géométrique.

Finalement, dans le cas le plus général où on dispose de peu de connaissances a priori ou de connaissances très vagues, une modélisation générique s'impose et les validités, ne pouvant pas être prédéterminées, sont calculées en ligne en exploitant l'approche des résidus.

De toutes ces approches de calcul de validité, ressortent deux grandes classes de validité :

- les validités a priori qui peuvent être déterminées hors ligne en exploitant les connaissances disponibles a priori sur le système à étudier (approches directe, floue ou probabiliste) ;

- les validités a posteriori qui doivent être calculées en ligne en considérant les mesures effectuées à chaque instant (approche géométrique et celle des résidus).

I.6.2.1. Approche directe de calcul des validités

Dans le cas d'une modélisation idéale (paragraphe I.5.1), on dispose de connaissances idéales sur le processus ainsi que sur les domaines de fonctionnement ou de validité. Ces informations permettent de calculer les validités des modèles a priori d'une façon directe et évidente. Ces validités ont des valeurs binaires et la séquence de commutation est donc établie a priori.

Cette approche apparaît simple à mettre en œuvre mais pour qu'elle soit efficace, elle nécessite une expertise idéale permettant d'acquérir suffisamment de connaissances a priori sur le système à étudier.

I.6.2.2. Approche floue de calcul des validités

L'approche floue s'applique essentiellement pour la modélisation locale (paragraphe I.5.2). On rappelle qu'à partir des connaissances disponibles sur le processus, ce concept de modélisation suppose la détermination a priori des domaines de fonctionnement ou domaines de validité relatifs aux différents modèles locaux de la base [Delmotte, 97] [Marry-smith et al, 97]. La disposition ainsi que la façon avec laquelle ces domaines sont définis a une influence sur l'élaboration de la commande globale. En effet, un changement brusque entre les régimes de fonctionnement pourrait provoquer des discontinuités au niveau de la commande. C'est pour ces raisons que la logique floue [Zadeh, 73] s'avère un outil puissant pour la définition des domaines de validité qui peuvent être décrits comme des ensembles flous [Takagi-Sugeno, 85], permettant ainsi une transition douce à travers des méthodes d'interpolation [Johansen et al, 93].

Les ensembles flous sont caractérisés par des fonctions d'appartenance et constituent des outils naturels pour décrire les appartenances aux différents domaines de fonctionnement. Les degrés de pertinence ou validités sont déduits à travers un mécanisme d'inférence suivi d'une étape de défuzzification [Zadeh, 73].

Par exemple, dans le cas d'une modélisation de type Takagi-Sugeno, les conclusions des règles floues correspondent à des fonctions définissant des modèles analytiques :

$$Règle\ R_i:\quad SI\ z_1\ est\ Z_{i_1}\ ET\ \ldots\ ET\ z_p\ est\ Z_{i_p}\ ALORS\ y = f_i(x) \tag{I.30}$$

$z_1,\ldots,z_p$ étant les variables des prémisses et $Z_{i_j}\ (j=1,\ldots,p)$ les sous-ensembles flous. La fonction $f_i(x)$ représente au modèle M_i correspondant à la règle R_i.

Le mécanisme d'inférence utilise les prémisses de chaque règle pour calculer la pertinence du modèle correspondant. A chaque règle R_i est attribué un poids μ_i qui dépend des degrés d'appartenance des variables des prémisses z_j aux sous-ensembles flous Z_{i_j} et du choix du connecteur « *ET* » des prémisses.

Prenons par exemple le cas d'un connecteur de type produit, les poids des règles s'écrivent alors de la façon suivante :

$$\mu_i = \prod_{j=1}^{p} Z_{i_j}(z_j)\,; \tag{I.31}$$

avec $Z_{i_j}(z_j)$ les fonctions d'appartenance à valeurs dans [0,1].

Pour une défuzzyfication de type barycentre, le système global peut s'écrire comme suit :

$$f(x) = \frac{\sum_{i=1}^{r} \mu_i\, f_i(x)}{\sum_{i=1}^{r} \mu_i}\,; \tag{I.32}$$

soit :

$$y = f(x) = \sum_{i=1}^{r} \rho_i\, f_i(x)\,; \tag{I.33}$$

avec r le nombre de règles qui correspond aussi au nombre de modèles ; $\rho_i \in [0,1]$ est la fonction de pondération définie par :

$$\rho_i = \frac{\mu_i}{\sum_{i=1}^{r} \mu_i} \quad i = 1,\ldots,r\,. \tag{I.34}$$

Les $\rho_i (i = 1,\ldots,r)$ correspondent aux validités des modèles puisqu'ils vérifient la propriété de la somme convexe :

$$\begin{cases} \rho_i \geq 0 \quad i = 1,\ldots,r \\ \sum_{i=1}^{r} \rho_i = 1 \end{cases}. \tag{I.35}$$

Il est évident que l'approche floue est difficile à mettre en œuvre vu la complexité de la définition des sous-ensembles flous et des fonctions d'appartenance. Mais il est tout de même intéressant de signaler que cette approche présente quelques avantages surtout au niveau de la commande globale. En effet, cette dernière est obtenue par interpolation évitant ainsi les changements brusques lors des transitions entre les domaines de fonctionnement avec participation de tous les modèles à chaque instant et des validités déterminées hors ligne.

I.6.2.3. Approche probabiliste de calcul des validités

Le formalisme probabiliste est très utilisé en classification et en théorie de décision. Le principe repose sur la définition d'hypothèses à partir des observations et des connaissances statistiques afin de déduire à chaque instant le modèle ou le contrôleur le plus approprié.

Après plusieurs expériences et observations, on détermine les probabilités a priori des sous-modèles $P(M_i)\,(i=1,\ldots,K)$. Ensuite, on estime les densités de probabilité $p\langle Z \mid M_i \rangle$ liant les modèles à l'environnement du système, défini par le vecteur $Z=\begin{bmatrix} z_1 & z_2 & \ldots & z_m \end{bmatrix}$ (vecteur des paramètres de l'environnement) [Kardous Khaldi, 04] :

$$p\langle Z \mid M_i \rangle = \prod_{j=1}^{m} p\langle z_j \mid M_i \rangle . \tag{I.36}$$

En utilisant la règle de Bayes, la probabilité a posteriori du modèle M_i s'écrit de la façon suivante :

$$P\langle M_i \mid Z \rangle = \frac{p\langle Z \mid M_i \rangle P(M_i)}{p(Z)}; \tag{I.37}$$

avec :

$$p(Z) = \sum_{i=1}^{K} p\langle Z \mid M_i \rangle P(M_i). \tag{I.38}$$

En assimilant les probabilités a posteriori aux degrés de validité :

$$v_i = P\langle M_i \mid Z \rangle, \tag{I.39}$$

deux possibilités se présentent pour la déduction de la sortie ou de la commande multimodèle. En effet, soit on procède par commutation et dans ce cas on prend le meilleur modèle qui a la probabilité a posteriori la plus élevée ; soit on utilise une simple fusion en considérant les paramètres des modèles (ou commandes partielles) pondérés par les validités (probabilités a posteriori).

L'approche probabiliste est rarement utilisée pour l'estimation des validités dans les systèmes multimodèles.

I.6.2.4. Approche géométrique de calcul des validités

Dans le cas d'un multimodèle obtenu par linéarisation du modèle de référence autour de certains points de fonctionnement, on peut associer à chaque modèle de la base un domaine de validité représenté par une zone dans l'espace des variables d'état [Murray-Smith et al, 97]. La mise en œuvre de l'approche géométrique pour le calcul des degrés de pertinence des différents modèles paraît convenable.

On considère les notations suivantes :

- X : l'espace d'état du système ;
- x : l'état courant du système ;
- D_i : le domaine de validité du modèle M_i dans X.

La distance d_i entre le système et un modèle M_i peut être calculée de plusieurs manières.

Lorsque l'état du système est accessible, une méthode simple consiste à considérer la distance minimale entre l'état x et les états x_i du modèle M_i dans le domaine D_i :

$$d_i = \min_{x_i \in D_i} \left\| x - x_i \right\| \quad i = 1, \ldots, K \; ; \tag{I.40}$$

K étant le nombre de modèles de la base.

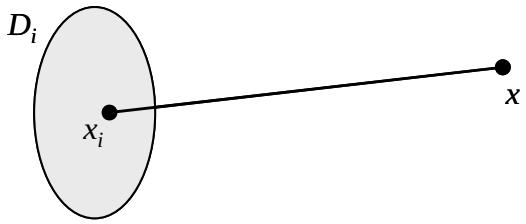


Figure I.10. Distance géométrique

Suivant la situation, les modèles peuvent avoir des domaines de validité de différentes formes et dispositions (figure I.11). En effet, il peut s'agir de domaines ponctuels (D_j) ou étendus (D_k, D_l, D_m) tout en étant disjoints (D_k et D_l) ou se chevauchant (D_l et D_m).

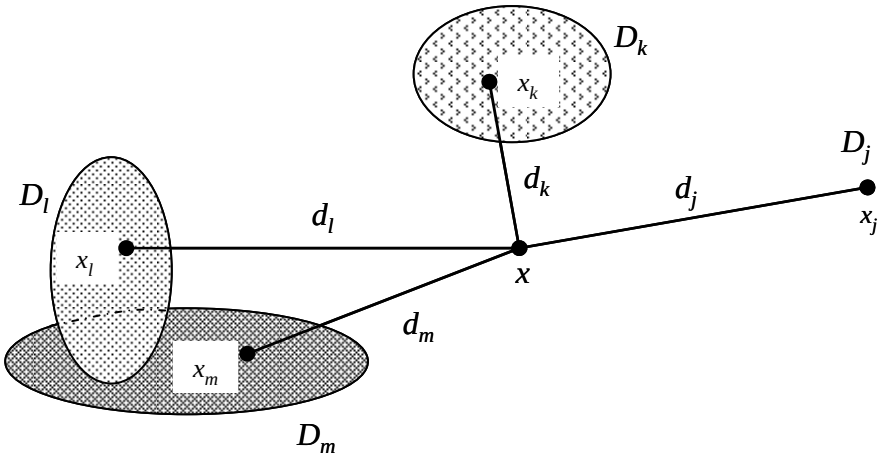


Figure I.11. Modélisation de la distance du système par rapport aux modèles locaux

Ainsi, pour chaque modèle de la base, on calcule la distance qui le sépare de l'état actuel du système. Afin de normaliser ces distances, on peut adopter l'une des deux méthodes :

- soit en considérant le maximum des distances $d_{\max}$ comme paramètre de normalisation ; la distance d_i' s'écrit alors :

$$d_i' = \frac{d_i}{d_{\max}} \quad i = 1, \ldots, K ; \tag{I.41}$$

- soit en considérant les distances relatives par rapport à la somme des distances :

$$d_i' = \frac{d_i}{\sum_{j=1}^{K} d_i} \quad i = 1, \ldots, K. \tag{I.42}$$

La deuxième méthode est plus utilisée puisqu'elle permet d'éviter l'introduction d'une variable supplémentaire $d_{\max}$.

Il est évident que plus distance du système par rapport à un modèle M_i est petite (le système est proche du domaine de validité D_i), plus la validité v_i du modèle correspondant est élevée et vice versa. Etant donné que les distances d_i' sont normalisées et que les validités sont inversement proportionnelles aux distances, une expression simple des validités peut être donnée par :

$$v_i = 1 - d_i'. \tag{I.43}$$

En littérature, certains auteurs se sont intéressés à l'approche géométrique et ont proposé d'autres expressions plus complexes de la validité en introduisant par exemple la notion de domaine de non validité [Dubois, 95] ou aussi de validité propre [Delmotte, 97].

I.6.2.5. Approche des résidus pour le calcul des validités

Comparée aux approches précédemment décrites, l'approche des résidus est la plus puissante puisqu'elle ne demande aucune connaissance a priori (dans ce cas, les autres approches deviennent inapplicables). Ainsi, les validités sont calculées en ligne à partir des mesures de la sortie du système et des sorties des modèles de la base.

L'approche des résidus est fondée sur le calcul d'une certaine erreur appelée résidu. Les résidus sont obtenus en comparant les valeurs estimées par chaque modèle de certaines variables du système aux valeurs réelles [Delmotte, 97] [Dubois, 95].

En dehors des systèmes multimodèles, la notion de résidu est très exploitée dans le domaine de la surveillance pour la détection des défauts et le diagnostic des pannes [Frank, 90]. Grâce à sa richesse et sa flexibilité, l'approche des résidus a fait un outil d'estimation des indices de validité dans plusieurs travaux de recherche portant sur les systèmes multimodèles (par exemple [Delmotte, 97] [Elfelly et al, 08] [Ksouri-Lahmari, 99] [Mezghani, 00] [Narendra et al, 97]).

Un résidu représente l'erreur entre deux valeurs ou deux ensembles de valeurs d'une même variable [Kardous Khaldi, 04]. Dans le cadre des approches multimodèles, cette variable correspond à la sortie ou à l'état du système. Ainsi, à chaque instant et pour chaque modèle de la base, un résidu est calculé permettant de quantifier l'erreur commise par chaque modèle. Prenons par exemple la sortie du système comme variable pour estimer les erreurs des différents modèles. Le résidu r_i du modèle M_i peut alors être défini comme étant l'erreur entre la valeur de la sortie réelle y du processus étudié et sa valeur y_i donnée par le modèle correspondant :

$$r_i(k) = \left\| y(k) - y_i(k) \right\| \quad i = 1, \ldots, K \; ; \tag{I.44}$$

K étant le nombre de modèle de la base et $\left\| . \right\|$ une norme choisie.

On constate que l'expression du résidu rappelle celle de la distance (I.40) dans l'approche géométrique. Par analogie avec cette dernière approche, les résidus bruts r_i peuvent être normalisés selon deux méthodes :

$$r_i^{norm}(k) = \frac{r_i(k)}{\sum_{i=1}^{K} r_i(k)} \quad i = 1, \ldots, K \; ; \tag{I.45}$$

ou bien :

$$r_i^{norm}(k) = \frac{r_i(k)}{\|y(k)\|} \quad i = 1, \ldots, K \; . \tag{I.46}$$

En optant pour la deuxième méthode (I.46), on court le risque que la variable y soit nulle. Dans ce cas, une solution consiste à prolonger les résidus par continuité [Delmotte, 97]. Il faut aussi signaler que les valeurs de ces résidus peuvent varier en dehors de l'intervalle [0,1], ce qui les rend peu commodes pour la définition des validités. C'est pour ces raisons que dans la plupart des cas on adopte plutôt la première méthode de normalisation (I.45).

Il est légitime que plus le modèle se rapproche du système (une valeur de l'erreur ou du résidu faible), plus il est apte à mieux représenter le processus dans le domaine de fonctionnement en question. On constate alors que les validités et les résidus varient en sens inverse. Une expression simple des validités peut, par exemple, être donnée par :

$$v_i(k) = 1 - r_i^{norm}(k) \quad i = 1, \ldots, K. \tag{I.47}$$

Dans la littérature, d'autres expressions de la validité ont été proposées. Par exemple, par l'introduction de deux paramètres $v_{\min}$ et $err_{\max}$, les validités peuvent être écrites comme suit [Delmotte, 97]:

$$v_i(k) = \max\left(v_{\min}, 1 - r_i^{norm}(k) / err_{\max}\right); \tag{I.48}$$

$v_{\min}$ est un simple paramètre permettant d'éviter la division par zéro lorsque tous les modèles sont mauvais. $err_{\max}$ est un paramètre qui permet de moduler la vitesse de décroissance de validité d'un modèle en fonction de ses erreurs. Cependant, pour être efficace, le paramètre $err_{\max}$ doit être bien initialisé. En effet, lorsqu'il est mal conditionné, il peut induire en erreur tout le système multimodèle. Si par exemple, $err_{\max}$ est initialisé à une trop petite valeur, tous les modèles auront le même degré de validité égal à $v_{\min}$. Dans le cas

contraire (err_{max} assigné à une très grande valeur), tous les modèles auront un indice de validité proche de 1. Dans les deux cas, la fusion sera équivalente à une simple moyenne des modèles influant ainsi les résultats qui sont loin d'être satisfaisants. Mais une fois le paramètre err_{max} bien adapté, l'approche des résidus utilisant l'expression (I.48) se révèlera très intéressante et performante [Delmotte, 97].

D'une façon générale, l'approche des résidus s'avère très intéressante par rapport aux autres approches de calcul des validités notamment dans le cas où on ne dispose d'aucune connaissance sur le système à part les mesures de ses variables. Elle repose seulement sur des informations obtenues en ligne (a posteriori). C'est pour ces raisons que l'approche des résidus paraît bien adaptée à la modélisation générique (paragraphe I.5.3)

Dans le deuxième chapitre, nous allons présenter les détails de l'approche des résidus en évoquant certaines variantes des validités qui peuvent en être déduites telles que les validités simples et les validités renforcées avec différentes méthodes de renforcement.

I.7. Modélisation de processus complexes – Techniques neuro-floues – Position du problème

L'approche multimodèle a été récemment développée pour pallier aux difficultés rencontrées lors de la modélisation et la commande de processus complexes et/ou incertains. On rappelle que le principe consiste à remplacer le modèle unique de la modélisation classique par un ensemble de modèles plus simples linéaires ou non, formant ce qu'on a appelé base de modèles. Chacun de ces modèles représente le processus dans un domaine particulier de son espace de fonctionnement. Plusieurs systèmes multimodèles ont été envisagés dans différents contextes pour la représentation et la commande de processus complexes [Johansen et al, 99]. La principale différence entre ces systèmes porte sur la procédure de construction de la base, la méthode adoptée pour l'identification des modèles ainsi que leur type.

I.7.1. Synthèse des systèmes multimodèles à base de linéarisation

En 1985, Takagi & Sugeno ont proposé une représentation floue du processus par une combinaison de modèles linéaires locaux obtenus par application de méthodes de linéarisation [Takagi et al, 85]. Cette approche a suscité l'intérêt de plusieurs chercheurs [Angelov et al,

04] [Feng, 06] et a été appliquée dans plusieurs domaines mais elle est souvent confrontée à des difficultés de manque d'information sur le processus étudié ou une incertitude paramétrique importante. Elle peut nécessiter parfois la mise en œuvre d'un grand nombre de modèles alourdissant ainsi la tâche de commande. Ceci a incité un grand nombre de chercheurs à proposer d'autres systèmes multimodèles afin de remédier à ces difficultés.

Ainsi, des approches de détermination de la base de modèles, basées sur le théorème de Kharitonov, ont été développées pour la modélisation de systèmes complexes incertains (continus et discrets) à paramètres bornés [Kardous Khaldi, 04] [Ksouri-Lahmari et al, 04] [Mezghani, 00]. Les approches citées précédemment requièrent la connaissance des limites des variations des paramètres du système, de telles informations ne sont pas toujours disponibles notamment pour les systèmes incertains dont les domaines de variations des paramètres sont souvent inconnus ou mal connus. D'autres travaux [Leith et al, 99] [Schorten et al, 99] proposent de décrire le système par un ensemble de modèles locaux définis en général à partir d'un modèle de référence linéarisé autour de certains points de fonctionnement.

Cependant, la représentation multimodèle devient plus difficile à concevoir dans le cas où les sous-modèles sont non-linéaires ou doivent être déterminés à partir des mesures de type entrée/sortie, les seules informations disponibles sur le processus à étudier (système de type « boîte noire »). Quelques résultats ont été présentés dans [Borne et al, 08]. Dans ce cas, les approches multimodèles précédemment citées ne sont plus valables compte tenu du manque de connaissances a priori. On parle alors d'une modélisation générique (paragraphe I.5.3) du système. Le système étant de type « boîte noire », les modèles de la base sont obtenus par identification paramétrique (paragraphe I.4.2). Dans cette situation, il faut élaborer des méthodes de construction de la base de modèles plus sophistiquées dont la mise en œuvre est basée sur l'exploitation des données entrée/sortie.

I.7.2. Structures multimodèles à base de réseaux de neurones

Dans ce contexte, il a été prouvé que les réseaux de neurones artificiels avec des architectures et des algorithmes d'apprentissage très variés [Haykin, 94], sont de plus en plus utilisés par les approches multimodèles. Grâce à leur capacité d'approximation et d'apprentissage, ils ont été adoptés dans plusieurs études pour la modélisation, l'identification, la prédiction et le contrôle de processus dynamiques [Hunt et al, 92]

[Narendra et al, 90] [Omatu et al, 95] [Pham et al, 95] [Sastry et al, 94] [Yip et al, 94]. Partant du principe que le réseau de neurones est considéré comme un modèle non linéaire de par les relations qui lient les neurones entre eux, plusieurs chercheurs ont pensé à s'en servir pour la l'identification de processus complexes (en général fortement non linéaires). Dans ce contexte, certaines études [Manioudakis et al, 01] [Selmic et al, 01] [Vieira et al, 04] proposent de concevoir un seul réseau de neurones en utilisant une représentation directe des dynamiques non linéaires à travers les neurones. D'autres suggèrent de mettre en œuvre plusieurs réseaux de neurones, chacun correspondant à un domaine de fonctionnement bien particulier [Fu et al, 07] [Yu, 06]. L'inconvénient majeur de ces approches est que le modèle neuronal d'identification n'est pas paramétrique, ce qui ne permet pas d'utiliser l'information obtenue directement lors de la conception de la commande.

I.7.3. Aide à la conception de structures multimodèles par des méthodes neuro-floues

Certains types de réseaux de neurones sont aussi connus pour être de bons outils de classification. De ce fait, plusieurs chercheurs se sont intéressés à l'idée de les utiliser pour la construction de la base de données du système multimodèle à partir des données entrée/sortie. Généralement, il s'agit de la carte de Kohonen utilisée comme technique de classification de données [Ben Aoun Talmoudi, 05] [Cho et al, 06] [Kukolj et al, 04]. Le principe de cette classe d'approches multimodèles consiste tout d'abord à classifier les mesures, calculer le nombre de modèles de la base et ensuite estimer leurs paramètres. La dernière étape consiste à établir les fonctions permettant de calculer en ligne les degrés de validité des différents modèles. Toujours dans le but de l'identification des modèles de la base, une autre possibilité consiste à recourir à des méthodes neuro-floues, combinant les réseaux de neurones avec des techniques floues (logique floue ou les K-moyennes floues par exemple) [Cheng et al, 06] [Gegúndez et al, 08] [Ronen et al, 02].

Qu'il s'agisse de techniques purement neuronales ou neuro-floues utilisées pour la détermination de la base de modèles, la classe des systèmes multimodèles, précédemment décrite, semble très intéressante. Ainsi, les différentes approches proposées dans ce contexte ont été appliquées dans plusieurs domaines (ingénierie, chimie, biologie, etc.) surtout lorsqu'il s'agit de systèmes incertains ou fortement non linéaires. Cependant, ces approches présentent quelques limitations. Tout d'abord, il s'agit généralement de systèmes multimodèles conçus

pour des applications bien spécifiques, ce qui restreint leur champ d'utilisation. D'autre part, les méthodes proposées sont, dans la plupart des cas, difficiles à mettre en œuvre et conduisent parfois à une base de modèles assez volumineuse (tel est les cas des travaux de [Cho et al, 07] dans lesquels un bio-réacteur a été représenté par 196 modèles), ce qui va à l'encontre du principe des multimodèles (qui repose sur la simplicité) et complique la tâche de conception de la commande. D'autres problèmes peuvent être évoqués tels que la détermination du nombre de modèles ou bien le choix de la méthode de calcul des validités la plus adéquate.

En considérant les limitations des méthodes proposées pour la conception de systèmes multimodèles, notre objectif était de développer une approche multimodèle pour la représentation et la commande de systèmes complexes qui soit la plus générale possible, efficace, facile à mettre en œuvre et qui n'exige pas de connaissances a priori (les seules mesures des variables du système suffisent pour sa mise en œuvre). Contrairement à la plupart des travaux proposés dans le domaine qui utilisent les algorithmes de classification pour l'identification de modèles de type Takagi-Sugeno [Kukolj et al, 04] [Vernieuwe et al, 06], notre approche vise plutôt une représentation multimodèle classique en se basant sur diverses techniques de classification pour établir le nombre de modèles et l'étendue des domaines de fonctionnement (réseaux de neurones et algorithmes des K-moyennes) et en utilisant seulement les données de type entrée/sortie. Dans notre étude, nous proposons une solution pour le problème d'estimation du nombre de modèles. D'autre part, on a essayé d'améliorer la procédure de classification, qui vise à construire la base de modèles à partir de mesures d'entrée sortie, en proposant d'appliquer trois algorithmes : carte de Kohonen, K-moyennes et K-moyennes floues. Ceci nous a permis d'établir quelques interprétations sur le choix de la méthode de classification la plus appropriée et de distinguer deux types de modèles, suivant que les domaines de fonctionnement correspondants présentent un faible ou un important chevauchement. La procédure d'estimation des degrés de pertinence des modèles a été également abordée en proposant deux méthodes de calcul de validités et quelques interprétations sur le choix de celle qui convient le mieux.

I.8. Conclusion

Partant du principe que l'approche multimodèle permet de remplacer un modèle unique par une bibliothèque de modèles plus simples, les systèmes multimodèles, d'une façon

générale, permettent de faciliter l'étude et la synthèse de processus complexes tout en évitant les difficultés rencontrées lors de la conception de la commande. Ces dernières années, cette approche a connu un grand succès et a suscité l'intérêt de plusieurs chercheurs, ce qui a donné lieu à plusieurs types de systèmes multimodèles. Certains de ces systèmes ont été destinés à l'étude de processus complexes bien particuliers tels que les systèmes incertains à paramètres variables et bornés [Kardous Khaldi, 04] [Ksouri-Lahmari, 99] [Mezghani, 00], d'autres exigent la présence d'un modèle de référence (par exemple [Leith et al, 99] [Schorten et al, 99]). Moins nombreuses sont les approches multimodèles qui ne nécessitent pas de connaissances a priori sur le processus à étudier [Baruch et al, 08] [Cho et al, 07] [Principe et al, 98].

Bien qu'elles partent du même principe, les approches multimodèles proposées diffèrent sur plusieurs points tels que la procédure de la construction de la base de modèles, les méthodes de calcul des indices de validité ou bien aussi la stratégie adoptée pour la conception de la commande globale. Dans le deuxième chapitre, nous allons présenter l'approche multimodèle proposée dans le cadre de notre étude pour la représentation de processus complexes en détaillant son principe et les différentes méthodes adoptées pour sa mise en œuvre.

Chapitre II

Méthodes de classification pour l'aide à la conception d'une représentation multimodèle de processus complexes

II.1. Introduction

II.2. Démarche envisagée pour la détermination de la structure multimodèle

II.3. Identification paramétrique des modèles de la base

II.4. Estimation des degrés de pertinence des modèles et validation de la modélisation proposée

II.5. Exemples de modélisation de processus complexes

II.6. Interprétations des résultats de la mise en œuvre de l'approche proposée

II.7. Conclusion

II.1. Introduction

Dans le premier chapitre, nous avons introduit la notion multimodèle qui consiste à représenter un système complexe par un ensemble de modèles simples constituant la base de modèles. A travers la conception d'un mécanisme de décision adéquat, ces modèles sont capables de reproduire le comportement du processus dans l'ensemble de son espace de fonctionnement. Ainsi, nous avons présenté les différentes classes des multimodèles, les méthodes de leur obtention ainsi que les concepts de modélisation qui dépendent des connaissances a priori disponibles sur le système.

L'approche multimodèle a fait le sujet de plusieurs travaux de recherche et plusieurs applications ont été envisagées dans différents domaines tels que la chimie ou la biologie [Böling et al, 07] [Liu et al, 08], l'ingénierie ou l'industrie [Cho et al, 06] [Salgado et al, 05].

Les structures multimodèles proposées sont généralement basées sur des connaissances a priori sur les systèmes étudiés et les modèles générés sont obtenus soit par linéarisation autour de certains points de fonctionnement à partir d'un modèle non linéaire de base, soit par transformation polytopique des termes non linéaires et/ou non constants en utilisant les bornitudes des domaines de variation.

Par contre, la modélisation devient beaucoup plus compliquée lorsqu'on ne dispose d'aucune information a priori sur le processus qui peut être qualifié de « boîte noire ». Dans ce cas, la seule possibilité est de se servir des données de type entrée/sortie afin de construire la base de modèles en faisant appel à des techniques d'identification. Ces méthodes sont généralement basées sur des techniques neuro-floues pour la classification non supervisée, parmi lesquelles on peut citer les réseaux de neurones artificiels qui ont été utilisés, soit en tant que modèle non linéaire, soit comme une technique permettant la sélection des différents domaines de fonctionnement. Dans ce contexte, l'élaboration d'une structure multimodèle évoque plusieurs problèmes avec de nombreux degrés de liberté tels que la détermination du nombre de modèles, la procédure à adopter pour la construction de la base, la structure et les paramètres des différents modèles ainsi que la définition des méthodes permettant d'évaluer la contribution de chacun des modèles à la représentation du système dans chaque point de fonctionnement.

Dans ce chapitre, nous proposons une approche pratique pour la représentation multimodèle de processus complexes. On exclura de notre étude les systèmes chaotiques ou avec une sensibilité paramétrique trop importante. En se basant seulement sur les données de type entrée/sortie collectées sur le système, l'approche envisagée consiste tout d'abord à déterminer le nombre de modèles de la base en proposant une nouvelle technique basée sur un apprentissage compétitif pénalisant le rival. Ensuite, l'utilisation d'un algorithme de classification adéquat permet la sélection des différents domaines de fonctionnement à partir desquels les modèles sont déduits en faisant appel à des méthodes classiques d'identification structurelle et paramétrique. Et enfin, nous définissons deux méthodes à mettre en œuvre pour le calcul des validités. L'approche proposée présente la particularité de pouvoir distinguer entre deux classes de systèmes et d'adapter en conséquence le traitement au niveau du choix de l'algorithme de classification le plus approprié d'une part et de la méthode de calcul des degrés de pertinence des modèles d'autre part. L'étude de quelques systèmes dynamiques complexes permet d'illustrer l'intérêt et l'efficacité de l'approche suggérée.

II.2. Démarche envisagée pour la détermination de la structure multimodèle

Pour la détermination d'une structure multimodèle efficace, nous proposons une approche basée sur des techniques de classification neuro-floues. Cette dernière permet la représentation multimodèle de systèmes complexes incertains et fortement non linéaires. Pour sa mise en œuvre, les besoins en matière d'informations a priori sur le processus sont minimes. En effet, elle ne nécessite ni la définition d'un modèle global ni la connaissance des domaines de variation des paramètres du système, les seules mesures collectées sur les variables d'entrée/sortie suffisent, permettant ainsi de surmonter le problème du manque d'information. En étant assez souple et flexible, notre approche s'inscrit dans le cadre d'une modélisation générique. La particularité de l'ensemble de nos travaux [Elfelly et al, 10a] réside essentiellement dans la démarche de construction de la base de données qui sera détaillée dans les paragraphes suivants.

II.2.1. Procédure de construction de la base de modèles

Partant des seules connaissances disponibles sur le processus à étudier, l'approche envisagée permet la construction de la base de modèles en utilisant des algorithmes de classification [Du, 10]. La mise en œuvre de cette approche nécessite tout d'abord la détermination de la taille de la base de la structure multimodèle. A cet effet, nous proposons une méthode neuronale, permettant de résoudre le problème du nombre de modèles rencontré dans de nombreux travaux élaborés dans le domaine ([Cho et al, 07] par exemple). En se servant d'un ensemble de données entrée/sortie (décrivant au mieux la dynamique du système dans les différents domaines de fonctionnement), l'application d'un réseau de neurones artificiels à apprentissage compétitif pénalisant le rival [Xu et al, 92] permet une bonne estimation de la taille de la base.

Une fois le nombre de modèles déterminé, l'étape suivante consiste à répartir les données en classes en utilisant un algorithme de classification bien adéquat. En considérant plusieurs critères (efficacité, simplicité, temps d'exécution, etc.), nous avons choisi de nous intéresser à trois méthodes : les cartes de Kohonen, les K-moyennes et les K-moyennes floues. Suivant certains aspects du processus, on fait appel à la méthode la plus convenable (une hypothèse sera présentée dans les paragraphes suivant portant sur le choix de la méthode de classification). Le résultat de cette classification se présente sous forme d'une certaine répartition des données entrée/sortie dans l'ensemble de l'espace du fonctionnement du système. Chacune des classes de données générées correspond à un aspect particulier de la dynamique du système étudié.

Les résultats de la classification de données sont ensuite exploités pour l'identification des différents modèles de la base. En effet, chaque classe correspond à un domaine de fonctionnement particulier. En faisant appel à des méthodes d'identification paramétrique, on fait correspondre un modèle linéaire à chacune de ces classes de données. Les modèles ainsi obtenus sont assez simples surtout lorsqu'on les compare au système initial. Cette simplicité justifie l'intérêt et l'efficacité de l'approche multimodèle notamment lors de la conception de la commande.

Une fois la base de modèles construite, il convient donc de valider la structure multimodèle ainsi obtenue. Pour cela, la stratégie de fusion des sorties partielles, correspondant aux différents modèles de la base, pondérées par leurs degrés de validité

respectifs, est adoptée afin de générer la sortie multimodèle. Etant donné qu'on ne dispose d'aucune connaissance a priori sur le système, l'approche des résidus (cf. paragraphe I.6.2.5) section paraît la plus adaptée pour le calcul des degrés de pertinence des modèles.

II.2.2. Détermination du nombre de modèles

Le sujet de l'estimation de la taille de la base a toujours constitué un problème pour la représentation de systèmes complexes de type « boîte noire » pour lesquels on ne dispose d'aucune connaissance a priori ni sur l'étendu des domaines de fonctionnement ni sur leur nombre. Par exemple, les études basées sur l'approche de Takagi-Sugeno ne définissent pas généralement une méthode de détermination du nombre de modèles et peuvent aboutir à une base assez volumineuse (un grand nombre de modèles) alourdissant ainsi les tâches d'analyse de commande. Par ailleurs, certaines méthodes heuristiques ont été mises en œuvre à cet effet. Dans ce contexte, on peut citer par exemple les travaux de Talmoudi [Ben Aoun Talmoudi, 05] où la méthode proposée est seulement basée sur le nombre de données par classe, ou bien aussi les travaux de [Cho et al, 07] dans lesquels on présente une technique basée sur plusieurs simulations de la carte de Kohonen avec différentes tailles. Cette dernière méthode conduit souvent à une dimension élevée du système multimodèle dont le nombre de modèles atteint 196 dans un exemple donné dans [Cho et al, 07], ce qui va à l'encontre du principe de l'approche multimodèle basée sur la simplicité.

A propos de ce sujet, nous avons pensé à faire appel à une méthode permettant d'estimer le nombre de modèles de la base seulement à partir de la répartition des données entrée/sortie dans l'espace de fonctionnement. En n'ayant aucune autre information sur le système, l'idée consiste à appliquer un algorithme de classification avec un apprentissage non supervisé. Dans ce contexte, la plupart des algorithmes existants ne permettent pas la sélection du nombre adéquat de classes. Par ailleurs, plusieurs résultats expérimentaux ont montré qu'un algorithme à apprentissage compétitif pénalisant le rival est capable d'estimer le nombre de classes en considérant un ensemble de données dans le but de les répartir en domaines [Borne et al, 07].

Ainsi, la première étape de l'approche proposée consiste à mettre en œuvre un algorithme de classification à apprentissage compétitif pénalisant le rival pour l'estimation du nombre de modèles de la base. Pour cela, il faut tout d'abord disposer d'ensemble de données

de type entrée/sortie qui décrit au mieux la dynamique du système complexe à étudier dans l'ensemble de ses domaines de fonctionnement, d'où l'importance de l'étape de collecte de mesures. En effet, cette dernière doit être bien menée puisqu'elle constitue la base de notre étude.

II.2.2.1. Collecte de données et type entrée/sortie du système

Comme il a été déjà mentionné, l'approche envisagée dans le cadre de nos travaux permet de modéliser des systèmes complexes de type « boîte noire ». La difficulté de la modélisation de ce genre de systèmes réside dans le manque et parfois l'absence de connaissances a priori sur la dynamique du système ou bien sur l'ensemble de ses domaines de fonctionnement. Par conséquent, la seule solution est de se servir des mesures des différentes variables d'entrée/sortie. Ces mesures sont ainsi obtenues à l'issue d'une étape appelée collecte de données qui constitue la base des approches multimodèles visant les systèmes de type « boîte noire ». C'est pour cette raison qu'une importance considérable doit être accordée à cette étape. En effet, pour qu'elles soient assez efficaces et précises, de telles approches doivent partir d'un ensemble de données entrée/sortie pouvant décrire le processus dans tous ses domaines de fonctionnement.

Pour sa mise en œuvre, l'étape de collecte de données nécessite tout d'abord que le système soit excité afin de récupérer les données utiles de type sortie ou entrée/sortie. A ce niveau, deux questions se posent :

- Quel signal utiliser pour l'excitation du système ?
- Quelles sont les variables mesurables, à considérer pour les étapes de classification, afin de mieux décrire le fonctionnement du processus ?

Pour répondre à ces questions, il n'existe pas ou peu de règles précises établies. Dans notre étude [Elfelly et al, 08], le signal d'excitation, le nombre et le type de variables mesurables à considérer dépendent surtout de la complexité du système étudié. D'une façon générale, le signal d'excitation doit être assez riche avec des paramètres bien choisis permettant ainsi une meilleure description du processus dans tous ses domaines de fonctionnement et une prise en compte de l'aspect complexe et non linéaire du système. A ce propos, certains signaux aléatoires, telles que les séquences binaires pseudo-aléatoires, sont connus pour être efficaces pour forcer le système à réagir dans tous ses domaines de

fonctionnement. Il est tout de même important de bien choisir certains paramètres du signal d'excitation tels que la fréquence et l'amplitude.

En ce qui concerne le nombre et le type de variables mesurables à considérer pour la détermination des domaines de fonctionnement à travers les techniques de classification, nous allons montrer à travers des exemples que parfois seulement les variables de sortie suffisent pour décrire la dynamique du système. Cependant, dans certaines situations, ces variables s'avèrent insuffisantes et il devient donc intéressant de considérer les variables d'entrée aussi notamment dans le cas de systèmes fortement non linéaires et/ou dont les paramètres dépendent de la sortie et/ou de l'entrée.

Sachant que l'approche envisagée est applicable aussi bien aux systèmes discrets qu'aux systèmes continus, il est indispensable, dans le deuxième cas, de passer par une étape de discrétisation des signaux d'entrée et de sortie, en considérant une période d'échantillonnage adéquate [Borne et al, 93].

Une fois les mesures des variables de sortie ou d'entrée/sortie générées, l'étape suivante consiste à estimer le nombre de modèles de la base en mettant en œuvre un réseau de neurones artificiels et un apprentissage compétitif pénalisant le rival.

II.2.2.2. Apprentissage compétitif pénalisant le rival pour la sélection du nombre de modèles

Afin d'identifier les différents domaines de fonctionnement du système, les données de type sortie ou entrée/sortie doivent être réparties en un certain nombre de classes. A ce niveau, le principe de notre approche consiste à appliquer un algorithme de classification à apprentissage non supervisé (cf. paragraphe II.2.3.1).

La plupart des techniques de classification existantes [Jain et al, 88] [Mirkin, 96] ne permettent pas la sélection du nombre approprié de classes et exigent qu'il soit connu ou déterminé a priori. Ce nombre constitue un critère déterminant pour la performance de la classification dans l'approche multimodèle surtout lorsqu'on ne dispose d'aucune information sur les domaines de fonctionnement et leur nombre. L'apprentissage compétitif pénalisant le rival permet de résoudre ce problème. En effet, il a été prouvé qu'un algorithme utilisant cet apprentissage permet d'affecter le nombre adéquat de groupes à un ensemble de données

quand elles sont utilisées pour une classification. Ainsi, pour la résolution du problème concernant la taille de la base dans la structure multimodèle, on se propose d'appliquer cet algorithme.

- **Apprentissage compétitif pénalisant le rival**

L'apprentissage compétitif pénalisant le rival est une stratégie d'apprentissage non supervisé (proposé par Xu en 1993 [Xu et al, 93] et révisée par Tambe en 1996 [Tambe et al, 96]) qui génère automatiquement le nombre optimal de classes. Le principe soulignant cet apprentissage pourrait être considéré comme une extension de l'apprentissage compétitif basé sur la règle de Kohonen (paragraphe II.2.3.2). La particularité de l'apprentissage compétitif pénalisant le rival réside dans le fait qu'à la présentation d'une observation *x*, non seulement le vecteur poids du neurone gagnant est déplacé dans le sens de l'observation mais aussi son rival (deuxième gagnant) est repoussé ou pénalisé. Le coefficient suivant lequel le rival est pénalisé doit être très inférieur au coefficient d'apprentissage [Borne et al, 07].

Considérons un réseau de neurones à apprentissage compétitif, c'est-à-dire une couche de neurones avec une sortie u_i pour chaque neurone et son vecteur poids w_i, pour $i = 1, \ldots, N$; N étant le nombre de neurones de la couche de sortie.

L'algorithme d'apprentissage compétitif pénalisant le rival peut être décrit par les étapes suivantes.

1. *Initialiser les vecteurs poids w_i aléatoirement.*
2. *Prendre un exemple x de l'ensemble de données, et pour $i = 1, \ldots, N$, soit*

$$u_i = \begin{cases} 1 & \text{si } i = g \\ -1 & \text{si } i = r \\ 0 & \text{sinon} \end{cases} ; \qquad \text{(II.1)}$$

avec :

$$\gamma_g \left\| x - w_g \right\|^2 = \min_j \gamma_j \left\| x - w_j \right\|^2 ; \qquad \text{(II.2)}$$

$$\gamma_r \left\| x - w_r \right\|^2 = \min_{j \neq g} \gamma_j \left\| x - w_j \right\|^2 ; \qquad \text{(II.3)}$$

- ||*|| : *distance euclidienne* ;
- *g* : *indice du neurone gagnant* ;
- w_g : *vecteur poids du neurone qui gagne la compétition* ;
- *r : indice du deuxième gagnant* (*rival*) ;
- w_r : *vecteur poids du rival* ;
- γ_j : *facteur caractéristique de la fréquence relative de gain, utilisé pour réduire le taux de gain des neurones qui remportent la compétition fréquemment. Il est utilisé pour développer un ensemble de prototypes équiprobables représentant les données d'entrée.* γ_j *est calculé comme suit* [Murlidharan Nair et al, 03] :

$$\gamma_j = \frac{n_j}{\sum_{i=1}^{N} n_i} \ ; \tag{II.4}$$

sachant que n_j *représente le nombre cumulatif des occurrences pour lesquelles le neurone j gagne la compétition* ($u_j = 1$).

3. *Mettre à jour les vecteurs poids comme suit* :

$$w_j(t+1) = w_j(t) + \Delta w_j \ ; \tag{II.5}$$

avec :

$$\Delta w_j = \begin{cases} \alpha_g(t)\left(x - w_j(t)\right) & \text{si } u_j = 1 \\ -\alpha_r(t)\left(x - w_j(t)\right) & \text{si } u_j = -1 \\ 0 & \text{sinon} \end{cases} ; \tag{II.6}$$

$0 \le \alpha_g(t)$ *et* $\alpha_r(t) \le 1$ *sont respectivement le coefficient d'apprentissage du gagnant et celui de son rival. En pratique, les coefficients sont fixés à de petites valeurs ou bien dépendent du temps* (*partant de valeurs pas assez basses et ensuite réduits à zéro d'une certaine façon*) *avec* $\alpha_g(t) \gg \alpha_r(t)$ *à chaque étape.*

4. *Répéter les étapes* 2 *et* 3 *jusqu'à la convergence du processus d'apprentissage.*

Quelques méthodes ont été proposées pour la mise à jour des coefficients d'apprentissage $\alpha_g(t)$ et $\alpha_r(t)$ [Chen et al, 92] [King et al, 98] [Murlidharan Nair et al, 03].

- **Méthodes de calcul des coefficients d'apprentissage**

  Le principe consiste toujours à initialiser les deux coefficients α_g et α_r à de petites valeurs tout en respectant la contrainte $\alpha_g^i \gg \alpha_r^i$. Ensuite, ces coefficients sont recalculés à chaque itération en considérant une fonction décroissante.

  Par exemple, les coefficients d'apprentissage peuvent être calculés comme suit [Chen et al, 92] [Murlidharan Nair et al, 03] :

$$\alpha_g(t) = \frac{\alpha_g(t-1)}{\sqrt{1+\text{int}\left[t/N\right]}} ; \alpha_r(t) = \frac{\alpha_r(t-1)}{\sqrt{1+\text{int}\left[t/N\right]}} \ ; \qquad \text{(II.7)}$$

  où $\text{int}\left[\cdot\right]$ représente la partie entière de l'argument et *N* le nombre de neurones de la couche de sortie.

  Une autre méthode de mise à jour des coefficients pourrait être considérée. Celle-ci suppose la définition des valeurs finales α_g^f et α_r^f. A chaque itération, les coefficients peuvent être alors définis comme suit [King et al, 98] :

$$\alpha_g(t) = \alpha_g^i \left(\frac{\alpha_g^f}{\alpha_g^i}\right)^{t/t_{\max}} \ ; \ \alpha_r(t) = \alpha_r^i \left(\frac{\alpha_r^f}{\alpha_r^i}\right)^{t/t_{\max}} \ ; \qquad \text{(II.8)}$$

  sachant que $t_{\max}$ représente l'itération finale.

Ainsi, pour l'estimation du nombre de modèles de la base, le principe consiste tout d'abord à considérer un réseau de neurones compétitif à deux couches (une couche d'entrée et une couche de sortie). Le nombre de neurones de la couche d'entrée correspond au nombre de variables du système à considérer. Le nombre de neurones de la couche de sortie doit être initialisé à une valeur *N* assez élevée. On applique ensuite l'algorithme de l'apprentissage compétitif pénalisant le rival en présentant à chaque itération l'ensemble des mesures

collectées sur les variables mesurables du système. A la fin de l'apprentissage, deux cas de figure sont envisageables.

- Si le nombre *N* de neurones de sortie est supérieur au nombre réel de classes en présence dans l'ensemble des données, les neurones gagnants évoluent vers les centres des classes et les autres s'éloignent de l'ensemble des observations. Dans ce cas, seulement les centres entourés de données doivent être considérés et le nombre de classes correspond au nombre de centres retenus.
- Si au contraire le nombre *N* de neurones de sortie est inférieur au nombre réel de classes en présence, il y a oscillation des vecteurs poids entre les classes durant l'apprentissage, ce qui indique la nécessité d'ajouter un ou plusieurs neurones à la couche de sortie [Borne et al, 07].

II.2.3. Application de techniques de classification pour la détermination des classes de fonctionnement du système

Comme il a été déjà mentionné, lorsqu'on dispose uniquement de données de type entrée/sortie sur le système (de type « boîte noire »), une représentation multimodèle est envisageable en faisant appel à des techniques de classification et d'identification pour la construction de la base de modèles. Parmi l'éventail des techniques disponibles, nous avons choisi de nous intéresser à des méthodes neuro-floues permettant la détection de la base de modèles. En se basant sur ces méthodes, peu de travaux proposent une représentation multimodèle d'un système ou d'un modèle non linéaire par une combinaison variable de plusieurs sous-modèles linéaires ou non [Cho et al, 07] [Gegúndez et al, 08] [Principe et al, 98] [Ronen et al, 02] [Talmoudi et al, 08] [Xue et al, 06].

L'approche envisagée dans le cadre de nos travaux propose d'appliquer des techniques de classification neuronales et floues pour l'identification de systèmes complexes par un ensemble de sous-modèles linéaires [Elfelly et al, 08] [Elfelly et al, 10a]. En effet, après l'estimation de la taille de base de modèles par l'intermédiaire d'un réseau de neurones et apprentissage compétitif pénalisant adéquats, l'étape suivante consiste à répartir les mesures des variables (de type sortie ou entrée/sortie) du système en classes afin de générer les différents domaines de fonctionnement à partir desquels les modèles sont identifiés.

Etant donné qu'il s'agit d'une étape déterminante dans la construction de la base de modèles, une grande partie de notre étude a porté sur la phase de classification de données. Le premier objectif était alors de sélectionner les méthodes qui peuvent être utilisées à cet effet en considérant plusieurs critères tels que la simplicité, l'efficacité et la rapidité. La première contrainte réside dans le fait qu'on ne dispose que de données collectées sur le système et aucune ou très peu d'informations (généralement très vagues) sur les domaines de fonctionnement, ce qui nous oriente vers la classification non supervisée (paragraphe II.2.3.1). Dans un premier temps, notre choix s'est directement posé sur les réseaux de neurones et en particulier les cartes de Kohonen [Elfelly et al, 08]. Celles-ci font partie de la classe des réseaux de neurones les plus utilisés dans la classification non supervisée. Néanmoins, ces « petits cerveaux » peuvent paraître parfois assez complexes de point de vue nombre d'opérations de calcul et demandent souvent une configuration très particulière en considérant plusieurs paramètres.

Par ailleurs, il a été démontré que les algorithmes des K-moyennes (K-moyennes et K-moyennes floues) sont assez efficaces pour la classification de données grâce à leur simplicité et leur capacité de générer une bonne répartition de mesures en un certain nombre de classes [Cheung, 03] [Dembélé et al, 03] [Hore et al, 08] [Kanzawa et al, 08].

Ainsi, pour la répartition des données issues du système en un certain nombre de groupes représentants les domaines de fonctionnement, nous avons retenu trois techniques de classification : les cartes de Kohonen, l'algorithme des K-moyennes et celui des K-moyennes floues. Il est à noter qu'il s'agit d'un choix délibéré et que d'autres méthodes de classification non supervisée peuvent être utilisées telles que les SVM (Machines à Support de Vecteurs) [Cristianini et al, 00] ou l'algorithme EM (Espérance-Maximisation ou Expectation-Maximisation en anglais) [Celeux et al, 91] et donc une comparaison peut être envisagée.

Dans ce qui suit, on introduit tout d'abord les notions de réseau de neurones artificiels et d'apprentissage. Ensuite, nous présentons les trois algorithmes de classification retenus et nous émettons une hypothèse sur le choix de la méthode de classification la plus appropriée. Ce choix va dépendre de certaines caractéristiques du système étudié et donc de la répartition de l'ensemble des mesures de ses variables dans l'espace de fonctionnement. Cette hypothèse sera traitée et vérifiée à travers des exemples de simulation (cf. paragraphe II.5).

II.2.3.1. Réseaux de neurones artificiels et apprentissage

Un réseau de neurones artificiels est un modèle de calcul dont la conception est très schématiquement inspirée du fonctionnement des neurones biologiques.

Les réseaux de neurones sont généralement optimisés par des méthodes d'apprentissage de type statistique, si bien qu'ils sont placés d'une part dans la famille des applications statistiques, qu'ils enrichissent avec un ensemble de paradigmes permettant de générer de vastes espaces fonctionnels, souples et partiellement structurés, et d'autre part dans la famille des méthodes de l'intelligence artificielle qu'ils enrichissent en permettant de prendre des décisions s'appuyant davantage sur la perception que sur le raisonnement logique formel.[1]

Basés sur un mécanisme d'adaptation, il a été prouvé que les réseaux de neurones artificiels sont capables d'apprendre et de généraliser le comportement d'un système donné à travers un ensemble de données. Le principe consiste à extraire et mémoriser certaines caractéristiques du système à partir des données qui leur sont présentées, c'est ce qu'on appelle apprentissage d'un réseau de neurones.

Un réseau de neurones artificiels est une structure de traitement de l'information parallèle et distribuée, constituée d'unités de calcul (les neurones) interconnectées par des réseaux unidirectionnels, appelés connexions. Chaque unité de calcul n'a qu'une seule connexion de sortie qui peut être dupliquée en autant d'exemplaires que désirés, les duplicata transportant le même signal.

L'apprentissage peut alors être considéré comme étant la procédure de mise à jour des poids des connexions au sein du réseau jusqu'à obtention du comportement désiré. Il s'agit d'une caractéristique principale des réseaux de neurones qui peut se faire de différentes manières et selon différentes règles.

Les réseaux de neurones artificiels peuvent être classés en deux grandes catégories [Borne et al, 07] :

- les réseaux de neurones à apprentissage supervisé,
- les réseaux de neurones à apprentissage non supervisé,

[1] http://fr.wikipedia.org/wiki/R%C3%A9seau_de_neurones_artificiels

- **Apprentissage supervisé**

L'apprentissage supervisé est assimilé à un procédé d'induction où l'on cherche à produire automatiquement des règles à partir d'une base de données (appelée aussi base d'apprentissage) contenant des exemples (entrées + résultats). Dans ce type d'apprentissage, le réseau s'adapte par comparaison entre le résultat qu'il a calculé, en fonction des entrées fournies, et la réponse attendue en sortie. Ainsi, les poids des connexions du réseau vont être modifiées jusqu'à atteindre la bonne sortie, c'est-à-dire celle attendue, correspondant à une entrée donnée.

- **Apprentissage non supervisé**

L'apprentissage non supervisé est une méthode d'apprentissage automatique. Il s'agit de diviser un groupe hétérogène de données en sous-groupes de manière à ce que les données considérées comme les plus similaires soient associées au sein d'un groupe homogène et qu'au contraire les données considérées comme différentes se retrouvent dans d'autres groupes distincts, l'objectif étant de permettre une extraction de connaissance organisée à partir de ces données [Borne et al, 07].

Cette méthode se distingue de l'apprentissage supervisé par le fait qu'il n'existe pas de sortie a priori. L'entrée de l'apprentissage non supervisé est un ensemble de données collectées. Le principe consiste à considérer et traiter ces données comme des variables aléatoires et à construire un modèle de « densités jointes » pour cet ensemble de données.

Les réseaux de neurones à apprentissage compétitif, dont fait partie les cartes de Kohonen, constituent une classe particulière des réseaux à apprentissage non supervisé. Les cartes de Kohonen peuvent être utilisées dans plusieurs applications telles que la réduction de dimension, l'analyse ou la classification de données.

II.2.3.2. Classification par les cartes de Kohonen

Les cartes de Kohonen (ou cartes auto-adaptative de Kohonen) appelées aussi SOM (Self-Organizing Maps) font partie de la classe des réseaux de neurones artificiels à apprentissage compétitif non supervisé.

- **Principe**

Les cartes de Kohonen peuvent être considérées comme des structures intelligentes de représentation de données qui sont inspirées, comme beaucoup d'autres créations de l'intelligence artificielle, par la biologie. Il s'agit de reproduire le principe neuronal du cerveau des vertébrés : des stimuli de même nature excitent une région du cerveau bien particulière. Les neurones sont organisés dans le cortex de façon à interpréter tous les types de stimuli imaginables. De la même manière, la carte auto adaptative se déploie de façon à représenter un ensemble de données, et chaque neurone se spécialise pour représenter un groupe bien particulier de données selon les points communs qui les rassemblent. Elle permet une visualisation en dimension multiple de données[2].

Le principe de la classification par les cartes de Kohonen consiste alors à regrouper les données en un certain nombre de catégories. Les patrons similaires vont donc être rangés dans une même classe, en se basant sur les corrélations des données, et seront représentés par un seul neurone ; on parle de la règle « winner-takes-all » ou règle de Kohonen. Cette dernière possède la propriété d'auto-organisation qui lui permet de regrouper un ensemble de données, présentés séquentiellement au réseau, autour d'un certain nombre de centres représentatifs de classes.

- **Structure de la carte de Kohonen**

D'un point de vue architectural, les cartes auto-adaptatives de Kohonen sont constituées, généralement, de deux couches : une couche d'entrée et une couche de sortie. Cette dernière se présente sous forme d'une grille qui peut avoir une structure unidimensionnelle (figure II.1) ou bidimensionnelle (figure II.2). Dans chaque nœud de la grille se trouve un neurone. C'est par rapport à la couche de sortie que le terme « carte » a été attribué à ce type de réseau. Chaque nœud de la grille ou de la carte est relié à chaque neurone de la couche d'entrée à travers une connexion.

[2] http://fr.wikipedia.org/wiki/Carte_auto_adaptative

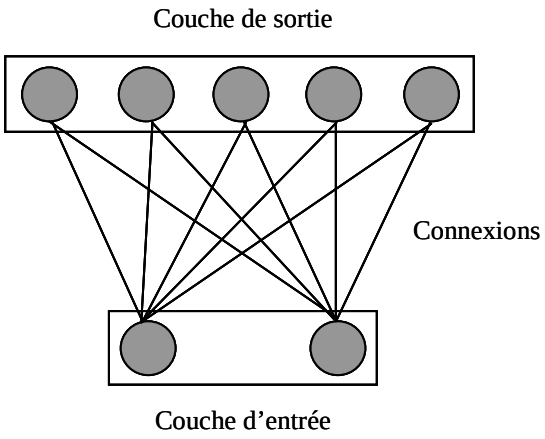


Figure II.1. Exemple d'une carte de Kohonen à structure unidimensionnelle

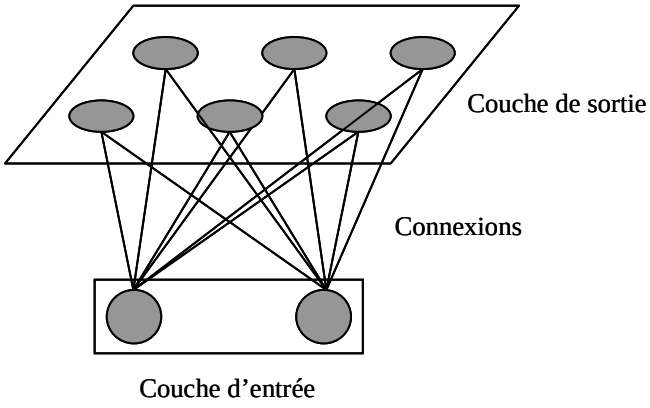


Figure II.2. Exemple d'une carte de Kohonen à structure bidimensionnelle

- **Apprentissage**

Pour la description de la procédure d'apprentissage, on s'intéresse à la classe des réseaux de Kohonen unidimensionnelles.

D'une façon générale, les cartes de Kohonen [Kohonen, 90] sont fondées sur les méthodes d'apprentissage compétitif. Il s'agit, en effet, d'une compétition qui a lieu, à chaque itération, avant la modification des poids du réseau ; et ce sont seulement les neurones qui gagnent la compétition, qui ont le droit de changer leurs poids de connexion. L'apprentissage d'un réseau auto-adaptatif (carte de Kohonen) consiste à appliquer la règle de Kohonen (ou règle « winner-takes-all »). La particularité de cette règle, c'est que l'apprentissage, à un instant donné, ne concerne qu'un seul neurone.

En effet, après une initialisation des poids de connexion à des valeurs aléatoires, l'apprentissage est déclenché en présentant une donnée *x* à l'entrée du réseau. Les neurones entrent alors en compétition pour être actifs. Ils sont à sortie binaire (0 ou 1) et, à un instant donné, un seul neurone est actif et sa sortie vaut 1. Il s'agit du neurone gagnant (qui a remporté la compétition) qui est le plus proche de l'entrée présentée en considérant la distance entre le vecteur d'entrée et les poids de connexion de chaque neurone. Les neurones perdants auront donc des sorties égales à 0.

Le neurone vainqueur est alors gratifié d'un changement de ses poids de connexions, on dit qu'il apprend en déplaçant ses poids vers les valeurs des entrées qui l'activent, afin d'augmenter ses chances de gagner lors de la présentation d'entrées similaires. Si un neurone ne répond pas à une entrée (perd la compétition), aucun ajustement de ses poids de connexion n'intervient. Ainsi, chaque neurone de sortie est spécialisé pour « détecter » une suite de formes similaires et devient alors un détecteur de caractéristiques.

L'apprentissage des cartes de Kohonen peut être résumé par l'algorithme suivant :

1. *Les poids des connexions* w_j ($j = 1, \ldots, K$) *sont initialisés à des valeurs aléatoires* ; *K étant le nombre de neurones de la couche de sortie.*
2. *Un vecteur d'entrée x est présenté au réseau.*
3. *La distance entre le vecteur d'entrée x et les poids* w_j*, reliant l'entrée à chaque neurone de sortie, est calculée.*
4. *Le neurone g correspondant à la plus petite distance et donc le plus proche du vecteur d'entrée gagne la compétition* ($u_g = 1$). *Les poids* w_g *reliant l'entrée à ce neurone sont alors actualisés.*

$$g = \arg\min_j \left\| x - w_j \right\| ;\ j = 1, \ldots, K ; \tag{II.9}$$

avec ||*|| : *distance euclidienne* ;

5. *La mise à jour des poids des connexions est donnée par* :

$$\begin{cases} w_g(t+1) = w_g(t) + \alpha \left(x(t) - w_g(t) \right) \\ w_j(t+1) = w_j(t) \quad \forall\, j \neq g,\ j = 1, \ldots, K \end{cases} ; \tag{II.10}$$

avec :

- g : *indice du neurone gagnant* ;
- w_g : *vecteur poids du neurone qui gagne la compétition* ;
- α : coefficient d'apprentissage ; $0 < \alpha \leq 1$.

Ainsi, à chaque itération, on présente toutes les entrées de la base d'apprentissage une à une. L'apprentissage s'arrête lorsque, entre deux itérations successives, les neurones ne bougent plus, ou très peu, c'est-à-dire que les poids de connexion se stabilisent.

- **Sélection des classes de fonctionnement du système par les cartes de Kohonen**

Dans le contexte des systèmes multimodèles, les cartes de Kohonen ont été utilisées dans certains travaux [Ben Aoun Talmoudi, 05] [Cho et al, 07] [Kukolj et al, 04] [Principe et al, 98] pour la classification de données dans le but de générer les différentes classes de fonctionnement du système.

Ayant déjà déterminé le nombre de classes ou de modèles de la base par la méthode d'apprentissage compétitif pénalisant le rival, on procède alors à la classification des données collectées sur les variables mesurables du système. Pour ce faire, on considère un réseau de Kohonen unidimensionnel pour lequel le nombre de neurones de la couche d'entrée est égal au nombre de variables mesurables (de sortie ou d'entrée/sortie) à considérer et le nombre de neurones de la couche de sortie est égal au nombre de classes déjà estimé [Elfelly et al, 08].

L'apprentissage résulte alors en une certaine répartition de données en K classes, K étant le nombre de neurones de la couche de sortie du réseau et qui correspond au nombre de modèles déjà estimé. Chaque classe est représentée par une partie de l'ensemble de données et un centre.

Malgré le succès de leur utilisation dans plusieurs domaines, les réseaux de neurones s'avèrent parfois difficiles à mettre en œuvre de part le choix de leurs paramètres (coefficient d'apprentissage, nombre d'itérations, initialisation des poids, fonction d'activation, etc). D'autre part, lorsqu'il s'agit d'une grande base de données très hétérogène et d'un nombre élevé de classes, l'algorithme peut devenir assez complexe et présenter un temps d'exécution rédhibitoire. Ceci nous a amené à penser à d'autres algorithmes de classifications plus simples tels les algorithmes des K-moyennes.

II.2.3.3. Classification par l'algorithme des K-moyennes

L'algorithme des K-moyennes [Forgy, 65] [MacQueen, 67], également appelé algorithme des nuées dynamiques, (en anglais K-means) est l'un des algorithmes de classification non supervisés les plus simples. Il couramment utilisé en analyse et classification de données. Il permet la détection automatique de classes en regroupant les données en fonction de leurs similitudes.

- **Principe**

L'algorithme des K-moyennes permet de partitionner une collection de données en *K* classes, *K* étant un nombre fixé par l'utilisateur. La procédure de mise en œuvre est assez simple et efficace.

Il a été montré que l'algorithme des K-moyennes vise à minimiser une fonction objectif *J*. D'une façon générale, cette fonction permet de mesurer le degré de cohésion des classes obtenues et peut être, par exemple, exprimée sous la forme suivante :

$$J = \sum_{j=1}^{K} \sum_{i=1}^{M} u_{ij} \left\| x_i - c_j \right\|^2 \ ; \qquad \text{(II.11)}$$

avec :

- ||*|| : une mesure de distance ;
- x_i : $i^{\text{ème}}$ donnée ;
- c_j : vecteur centre de la classe j ;
- u_{ij} : degré d'appartenance de x_i à la classe j tel que $u_{ij} \in \{0,1\}$ et

$$u_{ij} = \begin{cases} 1 & \text{si } x_i \text{ appartient à la classe } j \\ 0 & \text{sinon} \end{cases}$$

- M : nombre de données ;
- K : nombre de classes ($2 \leq K < M$).

- **Algorithme**

L'algorithme des K-moyennes consiste à définir *K* centres, un pour chaque classe. Les positions de ces centres doivent, tout d'abord, être initialisées d'une façon aléatoire. Ensuite, chaque point de la base de données doit être affecté au centre le plus proche. Une première itération est achevée résultant à une classification préliminaire lorsque toutes les données se trouvent affectées. A ce niveau, on procède à une mise à jour des positions des centres des nouvelles classes résultantes de l'itération précédente, et ensuite à une réaffectation des données. La procédure est répétée jusqu'à stabilisation des centres.

L'algorithme des K-moyennes peut être alors décrit par les étapes suivantes :

1. *Localiser K points dans l'espace des données qui vont être classifiées. Soient* $(c_1,\ldots,c_K)$ *les points ainsi obtenus. Ces points sont les représentants des K classes* $(C_1,\ldots,C_K)$*, qui sont pour l'instant vides. C'est l'étape de l'initialisation des positions des centres des classes.*
2. *Affecter chaque point de l'ensemble de données à l'une des classes en fonction du centre le plus proche* :

$$\arg\min_j \left\| x_i - c_j \right\|; \quad j = 1,\ldots,K.$$

3. *Quand toutes les données sont affectées, recalculer les positions des centres. Ces nouveaux correspondent à la moyenne des points de la classe* :

$$c_j = \frac{1}{\left|C_j\right|} \sum_{i, x_i \in C_j} \left\| x_i - c_j \right\|; \quad j = 1,\ldots,K; \tag{II.12}$$

où $\left|C_j\right|$ *représente le cardinal de la classe* C_j *c'est-à-dire le nombre de points affectés à cette classe.*

4. *Répéter les étapes* 2 et 3 *jusqu'à ce que les centres ne bougent plus ou très peu c'est-à-dire que la différence entre les anciens et les nouveaux centres est inférieur à un certain seuil fixé* (*arbitrairement petit*) $\varepsilon \in [0,1]$.

Malgré qu'on puisse prouver que son exécution finit toujours par donner un résultat, l'algorithme des K-moyennes n'aboutit pas nécessairement à la configuration optimale qui

correspond au minimum global de la fonction objectif *J* (expression II.11). En effet, la partition obtenue dépend de l'emplacement des centres initialement choisis, on dit que l'algorithme est sensible aux conditions initiales. Une solution de s'affranchir en partie de cette dépendance consiste à exécuter l'algorithme des K-moyennes avec des initialisations différentes, et on retient la meilleure partition sachant que la qualité de la partition est mesurée par la valeur de la fonction objectif *J*. Dans notre cas, l'initialisation des centres peut s'effectuer en partant des résultats obtenus lors de la détermination du nombre de classes par la méthode d'apprentissage pénalisant le rival.

- **Sélection des classes de fonctionnement du système par l'algorithme des K-moyennes**

En fournissant simplement le nombre de classes, qui dans notre étude, correspond au nombre déjà estimé par l'apprentissage compétitif pénalisant le rival, l'algorithme des K-moyennes permet une détection automatique des différentes classes de fonctionnement du système complexe étudié [Elfelly et al, 10a].

L'algorithme des K-moyennes présente l'avantage de la simplicité, ce qui favorise, dans certaines situations, son utilisation par rapport à la carte de Kohonen. En effet, dans le cas où on dispose par exemple d'un grand volume de données collectées sur le système et/ou d'un nombre assez élevé de classes, la classification par les K-moyennes s'avère plus efficace de point de vue complexité et temps d'exécution de l'algorithme dans le sens où l'apprentissage des cartes de Kohonen peut exiger de nombreuses itérations avant d'aboutir aux résultats attendus. Néanmoins, l'algorithme des K-moyennes présente l'inconvénient de la dépendance des conditions initiales, ce qui exige son exécution plusieurs fois afin d'aboutir à de meilleurs résultats.

Par ailleurs, la nature de données manipulées nous a amené à réfléchir à l'utilisation d'autres méthodes de classification. En effet, il s'agit dans la plupart des cas de données incertaines et/ou bruitées dans le sens où des erreurs peuvent s'introduire lors de la collecte de mesures, des erreurs qui peuvent être dues au matériel utilisé ou bien tout simplement des erreurs de mesure. D'autre part, il est très difficile de pouvoir travailler dans un environnement dépourvu de bruit. Sa présence (sous forme de perturbations par exemple) se répercute sur la nature de données qui peuvent être parfois très bruitées, ce qui nécessite un filtrage sachant qu'il y a un risque de perte d'information. L'influence du bruit sur les

données dépend de la sensibilité du système étudié. Ces constatations nous ont fait penser à la logique floue [Zadeh 73] vu qu'il s'agit d'une technique permettant la prise en compte d'ensembles définis d'une façon imprécise. Combinant la classification non supervisée et la logique floue, l'algorithme des K-moyennes floues constitue une bonne méthode qui peut être utilisée dans le cadre de notre étude.

II.2.3.4. Classification par l'algorithme des K-moyennes floues

L'algorithme des K-moyennes floues (en anglais fuzzy K-means) (développé par Dunn en 1973 [Dunn, 73] et amélioré par Bezdek en 1981 [Bezdek, 81]) est un algorithme de classification non supervisée floue. Inspiré de l'algorithme des K-moyennes (K-means), il introduit la notion d'ensemble flou dans la définition des classes dans la mesure où chaque point de l'ensemble des données peut appartenir à plus d'une classe avec des degrés différents (compris entre 0 et 1). Ainsi, l'algorithme des K-moyennes floues permet d'obtenir une partition floue de l'ensemble des données en définissant des frontières approximatives et floues entre les classes.

- **Principe**

Comme les autres techniques de classification non supervisée, l'algorithme des K-moyennes floues utilise un critère de minimisation des distances intra-classe et de maximisation des distances inter-classe, mais en donnant un certain degré d'appartenance à chaque classe pour chaque point de l'ensemble de données. Cet algorithme nécessite la connaissance préalable du nombre de classes. Ces dernières sont générées par un processus itératif en minimisant une certaine fonction objectif qui peut être définie comme suit :

$$J_m = \sum_{i=1}^{M}\sum_{j=1}^{K} u_{ij}^m \left\| x_i - c_j \right\|^2 ; \quad 1 \le m < \infty \ ; \tag{II.13}$$

avec :

- ||*|| : une norme de mesure de distance ;
- x_i : $i^{\text{ème}}$ donnée mesurée ;
- c_j : vecteur centre de la classe j ;

- u_{ij} : degré d'appartenance de x_i à la classe j tel que $u_{ij} \in [0,1]$; $\sum_{j=1}^{K} u_{ij} = 1$ et $0 < \sum_{i=1}^{M} u_{ij} < M$;
- m : constante représentant l'exposant de poids (nombre réel supérieur à 1) qui influence les degrés d'appartenance ;
- M : nombre d'observations ;
- K : nombre de classes ($2 \leq K <$ M).

- **Algorithme**

L'agorithme des K-moyennes floues est basé sur une optimisation itérative de la fonction objectif J_m décrite par l'équation (II.13) ; avec la mise à jour, à chaque itération, des degrés d'appartenance u_{ij} et des vecteurs centres c_j [Nascimento et al, 00]. Cette procédure s'arrête quand $\max_{ij}\left\{\left|u_{ij}(k+1) - u_{ij}(k)\right|\right\} < \varepsilon$; tel que ε définit un critère d'arrêt (arbitrairement petit) appartenant à l'intervalle [0,1] et k représente le numéro d'itération. Ainsi, l'algorithme converge vers un minimum local de J_m.

L'algorithme des K-moyennes floues se résume alors dans les étapes suivantes :

1. *Initialiser la matrice des degrés d'appartenance* $U = \left[u_{ij}\right]$, $U(0)$.
2. *A l'itération k, calculer les vecteurs centres* c_j ; $j = 1, \ldots, K$:

$$c_j = \frac{\sum_{i=1}^{M} u_{ij}^m \cdot x_i}{\sum_{i=1}^{M} u_{ij}^m}. \qquad \text{(II.14)}$$

3. *Actualiser la matrice des degrés d'appartenance* $U(k)$ *suivant les nouvelles positions des centres,* $U(k+1)$:

$$u_{ij} = \left[\sum_{l=1}^{K}\left(\frac{\left\|x_i - c_j\right\|}{\left\|x_i - c_l\right\|}\right)^{\frac{2}{m-1}}\right]^{-1}. \qquad \text{(II.15)}$$

4. *Retourner à l'étape* 2 *tant que* $\max_{ij}\left\{\left|u_{ij}(k+1) - u_{ij}(k)\right|\right\} \geq \varepsilon$.

- **Sélection des classes de fonctionnement du système par l'algorithme des K-moyennes floues**

Le principe de la classification des données par l'algorithme des K-moyennes floues est similaire à celui des K-moyennes sauf qu'avant de générer les différents domaines de fonctionnement, il faut affecter chaque observation à la classe pour laquelle il présente le degré d'appartenance le plus élevé.

L'application de l'algorithme des K-moyennes floues, pour la répartition des données collectées sur le système, s'avère très intéressante dans le cas d'une dynamique caractérisée par un recouvrement entre les domaines de fonctionnement.

II.2.3.5. Hypothèse sur la sélection de la méthode de classification

Dans les paragraphes précédents, on a présenté trois algorithmes (la carte de Kohonen, les K-moyennes et les K-moyennes floues) permettant la classification des données décrivant la dynamique du système étudié. L'objectif est le même pour ces trois méthodes et consiste à générer les différentes classes de fonctionnement du système.

Le but de cette diversité est de pouvoir sélectionner la technique de classification la plus appropriée en considérant certaines caractéristiques de processus qui peuvent être déduites à partir de la façon avec laquelle les données sont réparties dans l'ensemble de l'espace de fonctionnement, ce qui constitue une particularité de notre étude et permet d'étendre son application pour la modélisation de plusieurs classes de systèmes complexes.

En effet, en se référant aux résultats de l'application de l'algorithme d'apprentissage compétitif pénalisant le rival, qui constitue la première étape de notre approche et vise à estimer le nombre de modèles de la base, si on remarque un recouvrement entre les classes de données avec l'absence de séparation nette entre ces classes, on va privilégier l'algorithme des K-moyennes floues dans la mesure où ce dernier permet la prise en compte d'ensembles définis d'une façon imprécise. Dans le cas contraire, on fait appel à une classification par l'algorithme des K-moyennes ou les cartes de Kohonen suivant qu'il s'agit d'un grand volume de données (et éventuellement un nombre assez élevé de classes) ou non.

L'hypothèse ainsi présentée, sera traitée et vérifiée à travers une comparaison entre les résultats donnés par les algorithmes de classification, ici présentés, suite à leur mise en œuvre pour la modélisation de quelques exemples de systèmes complexes.

II.3. Identification paramétrique des modèles de la base

Après la détermination du nombre de modèles et la classification des données système, vient l'étape d'identification des différents modèles de la base. En effet, L'application de l'algorithme de classification le plus approprié (carte de Kohonen, K-moyennes ou K-moyennes floues) aboutit à une certaine répartition de l'ensemble de données. Chaque classe de fonctionnement est représentée par un ensemble de mesures d'entrée/sortie. Ces mesures sont alors exploitées pour l'identification des différents modèles. Parmi l'éventail des méthodes proposées à cet effet, nous avons choisi de faire appel à des méthodes classiques d'identification en définissant des modèles linéaires simples, ce qui permet de préserver les avantages de l'approche multimodèle qui sont essentiellement basés sur la simplicité.

Dans l'approche envisagée, on propose alors une structure multimodèle basée sur des modèles linéaires décrits par des équations affines. Pour ce faire, deux étapes sont requises pour l'identification de chaque modèle de la base : l'estimation structurelle (ou l'estimation de l'ordre) et l'estimation paramétrique permettant de définir les paramètres du modèle, en se basant sur les données relatives à chaque classe de fonctionnement résultant de l'étape de classification de données.

II.3.1. Estimation des ordres des modèles

La méthode retenue pour l'estimation des ordres des modèles est celle du test du Rapport des Déterminants Instrumentaux [Ben Abdennour et al, 01]. Dans ce qui suit, nous décrivons le principe de cette méthode.

Soit une représentation discrète de type ARX, décrite par l'équation linéaire de la forme suivante :

$$A\left(q^{-1}\right) y(k) = q^{-d} B\left(q^{-1}\right) u(k)\,; \tag{II.16}$$

avec :

- $u(k)$: signal d'entrée ;
- $y(k)$: signal de sortie ;
- q^{-1} : opérateur retard ;
- d : retard du processus ;

- $A\left(q^{-1}\right)$ et $B\left(q^{-1}\right)$: deux polynômes qui dépendent de l'opérateur retard $\left(q^{-1}\right)$ et qui sont définis comme suit :

$$A\left(q^{-1}\right)=1+a_1q^{-1}+a_2q^{-2}+\ldots+a_nq^{-n}; \tag{II.17}$$

$$B\left(q^{-1}\right)=b_1q^{-1}+b_2q^{-2}+\ldots+b_nq^{-n}; \tag{II.18}$$

où *n* représente l'ordre du modèle.

Pour l'estimation de la valeur de *n*, la méthode du test du Rapport des Déterminants Instrumentaux est essentiellement basée sur les conditions de rang concernant une matrice appelée « matrice d'information » qui contient les couples de mesures entrée/sortie. Cette matrice Q_m est définie comme suit :

$$Q_m=\frac{1}{N_{ob}}\sum_{k=1}^{N_{ob}}\begin{bmatrix} u(k) \\ u(k+1) \\ u(k-1) \\ u(k+2) \\ . \\ . \\ . \\ u(k-m+1) \\ u(k+m) \end{bmatrix}\begin{bmatrix} y(k+1) \\ u(k+1) \\ y(k+2) \\ u(k+2) \\ . \\ . \\ . \\ y(k+m) \\ u(k+m) \end{bmatrix}^T; \tag{II.19}$$

sachant que N_{ob} représente le nombre d'observations.

Le Rapport des Déterminants Instrumentaux *RDI*(*m*) est donné par la relation suivante :

$$RDI(m)=\left|\frac{\det\left(Q_m\right)}{\det\left(Q_{m+1}\right)}\right|. \tag{II.20}$$

Pour chaque valeur de *m* ($m\geq 1$), la procédure de détermination de l'ordre construit les matrices Q_m et Q_{m+1} et évalue le rapport $RDI(m)$, l'ordre du modèle correspond alors à la valeur de *m* pour laquelle on observe pour la première fois une augmentation significative du rapport $RDI(m)$.

Connaissant les ordres des différents modèles de la base, la deuxième phase d'identification consiste à estimer les paramètres de ces modèles.

II.3.2. Estimation des paramètres des modèles

Toujours en se basant sur les mesures entrée/sortie relatives à chaque ensemble de données résultant de la classification de données, on procède à l'estimation des paramètres des différents modèles de la base en considérant leurs ordres respectifs, déjà estimés par la méthode du test du Rapport des Déterminants Instrumentaux. Il s'agit de la dernière étape de la construction de la base de modèles.

Le problème de l'identification paramétrique consiste à déterminer, à partir de la connaissance de plusieurs mesures expérimentales caractérisant le fonctionnement dynamique du système, les valeurs des paramètres intervenant dans l'équation du modèle correspondant. A cet effet, nous avons sélectionné la méthode des moindres carrés récursifs [Ben Abdennour et al, 01] [Borne et al, 92a].

On considère le modèle linéaire décrit par l'équation (II.16). Celle-ci peut être réécrite de la façon suivante :

$$y(k) = \theta^T \varphi(k) ; \tag{II.21}$$

avec :

$$\varphi^T(k) = \left[-y(k-1),\ldots,-y(k-n),u(k-d-1),\ldots,u(k-d-n)\right] ; \tag{II.22}$$

$$\theta^T = \left[a_1, a_2,\ldots,a_n,b_1,b_{2,},\ldots,b_n\right]. \tag{II.23}$$

La méthode des moindres carrés récursifs consiste donc à estimer les paramètres du modèle en déterminant le vecteur θ par un algorithme récursif. A l'itération *k*, cet algorithme se résume par les instructions suivantes :

$$\begin{cases} \theta(k) = \theta(k-1) + P(k)\,\varphi(k)\,e(k) \\ P(k) = P(k-1) - \dfrac{P(k-1)\,\varphi(k)\,\varphi^T(k)\,P(k-1)}{1+\varphi^T(k)\,P(k-1)\,\varphi(k)} \\ e(k) = y(k) - \theta^T(k-1)\,\varphi(k) \end{cases} ; \tag{II.24}$$

où *P* représente la matrice de covariance et *e* l'erreur a priori entre la sortie réelle du système et la sortie obtenue en utilisant les paramètres estimés.

Une fois les modèles identifiés, on peut alors en déduire notre structure multimodèle. Celle-ci pourrait exploitée pour différents objectifs tels que la commande du processus en question. Mais avant de procéder à la commande, il faut tout d'abord définir une méthode permettant le calcul des degrés de pertinence (ou degrés de fiabilité) des différents modèles à chaque instant.

II.4. Estimation des degrés de pertinence des modèles et validation de la modélisation proposée

Lorsqu'il s'agit d'une représentation multimodèle du système, il est indispensable de pouvoir estimer, à chaque instant, le degré de justesse ou de fiabilité de chaque modèle pour savoir dans quelle mesure il faudra l'utiliser pour décrire le système global.

Contrairement aux différentes étapes de construction de la base précédemment décrites qui sont élaborées hors ligne, le calcul des degrés de pertinence des modèles (appelés aussi validités) s'effectue en ligne dans le sens où les validités sont calculées à chaque instant et dépendent de l'évolution du processus étudié.

Le problème de choix de la méthode de calcul des degrés de pertinence des modèles a souvent été évoqué dans les travaux portant sur des approches multimodèles. Dans ce mémoire, nous traitons ce sujet en proposant deux méthodes efficaces pour l'estimation des validités des modèles [Elfelly et al, 08]. Les méthodes choisies découlent de l'approche des résidus (évoquée dans le chapitre I) et sont bien adaptées aux systèmes étudiés qui ont comme particularité d'être des « boîtes noires ».

Dans ce qui suit, nous allons commencer par présenter les deux méthodes que nous avons sélectionnées pour le calcul des degrés de pertinence des modèles. Ensuite, nous proposons une hypothèse sur le choix de la méthode la plus appropriée en considérant certaines caractéristiques des classes de fonctionnement du système étudié. Et enfin, nous évoquons une procédure permettant la validation de la structure multimodèle proposée, une étape à ne pas négliger et à effectuer avant d'entamer toute exploitation de la base de modèles.

II.4.1. Méthodes adoptées pour le calcul des validités

Dans le chapitre I (section I.6), nous avons introduit la notion de validité et évoqué les différentes approches de calcul. Ces approches diffèrent selon que les validités sont calculées hors ligne en fonction des informations disponibles a priori, ou bien en ligne en fonction de mesures effectuées a posteriori [Delmotte, 97].

Dans l'ensemble de nos travaux, nous nous sommes intéressés aux systèmes incertains et/ou fortement non linéaires qui sont souvent considérés comme des « boîtes noires » c'est-à-dire que les connaissances a priori sont très rares et parfois absentes. Pour la représentation de ce genre de processus, nous avons opté pour une modélisation générique et des modèles obtenus par identification. Il s'avère alors indispensable de faire appel à une méthode assez puissante pour le calcul des validités, telle que l'approche des résidus, qui pourrait bien répondre à nos besoins dans la mesure où elle ne nécessite aucune connaissance a priori mais se base plutôt sur des informations qui sont obtenues en ligne et qui concernent les modèles et les sorties du système.

Qu'il s'agisse d'un simple calcul de la sortie multimodèle ou bien de la conception d'une commande globale, la définition d'une fonction permettant le calcul des validités en ligne est donc indispensable. Il s'agit d'une étape déterminante puisque les coefficients calculés vont avoir une influence directe sur la contribution de chaque modèle soit à la génération de la sortie multimodèle, soit au calcul de la commande globale. C'est pour ces raisons là qu'il faut bien choisir les techniques de calcul des validités. En se basent sur l'approche des résidus, nous présentons deux méthodes permettant d'estimer deux types de validités : validités simples et validités renforcées.

II.4.1.1. Résidus et validités

Comme son nom l'indique, il s'agit d'une approche de calcul des validités basée sur les résidus. La notion de résidu repose essentiellement sur le calcul de la distance entre le processus et le modèle considéré.

D'une façon générale, les résidus utilisés doivent satisfaire certaines conditions.

- Ils doivent être calculés seulement à partir des informations accessibles par mesure ou par estimation en ligne avec l'évolution du processus, telles que les entrées, les états ou bien aussi les sorties issues du processus et des différents modèles de la base.
- Les expressions de définition des résidus doivent être assez simples afin de faciliter leur implémentation et respecter un temps de calcul raisonnable dans la mesure où il s'agit d'un procédé en ligne.
- Les résidus doivent être normalisés de telle sorte qu'ils évoluent dans l'intervalle [0,1] et qu'ils vérifient la propriété de somme convexe :

$$\begin{cases} r_i^{norm}(k) \in [0,1] \quad i = 1,\ldots,K \\ \sum_{i=1}^{K} r_i^{norm}(k) = 1 \end{cases} ; \qquad \text{(II.25)}$$

où K représente le nombre de modèles de la base et k les différents instants de calcul.

Pour les mêmes raisons, les validités doivent aussi satisfaire ces conditions. Elles doivent en plus évoluer inversement à l'évolution des résidus.

Une expression simple de résidu peut être obtenue en considérant, à chaque instant, l'erreur entre la sortie du processus y et la sortie relative à chaque modèle y_i ($i = 1,\ldots,K$), comme le montre l'expression suivante :

$$r_i(k) = \left\| y(k) - y_i(k) \right\| \quad i = 1,\ldots,K \ ; \qquad \text{(II.26)}$$

sachant que :

$\left\| y(k) - y_i(k) \right\| = \left| y(k) - y_i(k) \right|$ si le système est mono-sortie ;

$\left\| y(k) - y_i(k) \right\| = \sum_{j=1}^{m} \left| y_j(k) - y_{i_j}(k) \right|$ si le système est multi-sorties ($y(k) \in \mathbb{R}^m$).

Dans le deuxième cas, d'autres types de normes peuvent être utilisés, tels que la norme euclidienne suivante :

$$\left\| y(k) - y_i(k) \right\| = \left(\sum_{j=1}^{m} \left(y_j(k) - y_{i_j}(k) \right)^2 \right)^{\frac{1}{2}}$$

Pour définir les résidus normalisés, il suffit de diviser l'expression (II.26) par la somme des résidus des *K* modèles :

$$r_i^{norm} = \frac{r_i}{\sum_{j=1}^{K} r_j} \quad i = 1, \ldots, K.. \tag{II.27}$$

Cette dernière expression vérifie bien les conditions sur les résidus, précédemment évoquées, et sera donc adoptée pour le calcul des validités.

En se basant sur l'approche des résidus, plusieurs méthodes ont été proposées pour le calcul des validités [Delmotte, 97] [Gasso, 00] [Leith et al, 99] [Narendra et al, 97]. Dans la cadre de nos travaux, nous allons définir deux méthodes de calcul des validités, en se basant sur l'expression (II.27) des résidus.

II.4.1.2. Première méthode de calcul : validités simples

En partant du principe que les validités et les résidus évoluent en sens inverse (voir chapitre I section I.6), une expression simple des validités peut être alors considérée :

$$v_i = 1 - r_i^{norm} \quad i = 1, \ldots, K. \tag{II.28}$$

Afin de satisfaire la propriété de somme convexe tout en gardant toujours les mêmes propriétés, les validités v_i peuvent être normalisées en divisant l'expression (II.28) par la somme des validités des *K* modèles, c'est ce qu'on va appeler validités simples :

$$v_i^{simp} = \frac{v_i}{\sum_{j=1}^{K} v_j} \quad i = 1, \ldots, K\ . \tag{II.29}$$

En tenant compte de la propriété de somme convexe des résidus (II.25), l'expression des validités simples (II.29) devient :

$$v_i^{simp} = \frac{v_i}{K-1} \quad i = 1, \ldots, K \,. \tag{II.29}$$

La méthode de calcul de validités, ainsi évoquée, présente l'avantage de la simplicité. Elle se montre particulièrement efficace dans les cas où les domaines de fonctionnement sont assez mélangés (présentant plusieurs zones communes). Dans ce cas, à un instant donné, le système se trouve généralement décrit par plusieurs modèles en même temps et aucun modèle ne peut représenter idéalement le système. Les résidus associés aux différents modèles sont alors plus au moins comparables. Par conséquent, l'utilisation des validités simples pour estimer, à chaque instant, le degré de pertinence des différents modèles, paraît bien adaptée à ce genre de situations dans la mesure où l'application de l'expression (II.29), telle qu'elle est définie, permet de déduire des validités aussi comparables que les résidus et permet ainsi de tenir compte de la contribution de chacun des modèles de la base.

II.4.1.3. Deuxième méthode de calcul : validités renforcées

Dans le cas où le système présente des classes de fonctionnement bien séparées c'est-à-dire qu'à un instant donné, très peu de modèles (souvent un seul modèle) sont valides (ou représentatifs) et tous les autres sont pratiquement faux. Ainsi, les résidus des premiers vont être très petits devant ceux des derniers. Afin de bien traduire cette situation, il faut que les mauvais modèles aient des validités assez faibles pour ne pas induire des phénomènes de perturbation sur les bons modèles qui doivent avoir en revanche des validités importantes. Dans ce genre de situation, les validités simples paraissent insuffisantes et inefficaces, d'où le besoin de définir d'autres méthodes de calcul plus sophistiquées.

Pour atténuer ce phénomène de perturbation causé par les mauvais modèles, une solution consiste à renforcer les validités des bons modèles au détriment des autres en faisant intervenir des opérations supplémentaires sur l'expression des validités. Dans ce qui suit, nous allons présenter une méthode de renforcement, parmi celles qui ont été proposées dans la littérature : Annexe B).

Le principe du renforcement consiste à multiplier la validité normalisée (donnée par l'expression (II.29)) de chaque modèle par le produit des valeurs complémentaires à 1 des validités des autres modèles [Delmotte, 97] [Kardous Khaldi, 04]. En appliquant le renforcement, l'expression (II.29) s'écrit alors comme suit :

$$v_i'^{renf} = v_i^{simp} \prod_{\substack{j=1 \\ j \neq i}}^{K} \left(1 - v_j^{simp}\right) \quad i = 1, \ldots, K \,. \tag{II.30}$$

En appliquant la méthode de renforcement aux validités simples, ces dernières perdent la propriété de somme convexe. C'est pour cela qu'une étape de normalisation est nécessaire. Les validités renforcées normalisées sont alors définies comme suit :

$$v_i^{renf} = \frac{v_i'^{renf}}{\sum_{j=1}^{K} v_j'^{renf}} \quad i = 1, \ldots, K \,. \tag{II.31}$$

En considérant l'expression (II.30), si à un instant *k*, un modèle est parfait (avec un résidu nul), sa validité normalisée sera donc égale à 1 et son complémentaire est nul. Ainsi, toutes les validités des autres modèles seront nulles, ce qui est tout à fait logique. En revanche, si deux modèles ou plus ont des résidus nuls (ce qui est envisageable dans le cas où les domaines de fonctionnement se chevauchent), toutes les validités renforcées correspondants aux différents modèles de la base, seront alors nulles. Ceci permet de déduire que les validités renforcées ne conviennent pas aux processus présentant un ou plusieurs recouvrements entre les classes de fonctionnement.

II.4.2. Hypothèse sur le choix de la méthode de calcul des validités

Après la détermination du nombre de modèles et la classification des données système, on a présenté deux méthodes de calcul des validités. Dérivées de l'approche des résidus, les deux techniques évoquées sont bien adaptées aux classes de systèmes étudiés.

Rappelons que l'approche des résidus est seulement basée sur des estimations faites en ligne et qui portent généralement sur les mesures des sorties du système d'une part et des différents modèles d'autre part. Cette précision impose que les méthodes utilisées doivent être décrites par des expressions simples nécessitant le moins d'opérations de calcul possibles afin de limiter le temps de leur exécution, d'où l'importance du choix de la technique d'estimation

des validités en faisant un compromis simplicité, efficacité. En effet, pour avoir une bonne modélisation du système qui soit la plus proche de la réalité, la méthode utilisée doit bien traduire le comportement du système dans les différents domaines de fonctionnement en offrant une meilleure estimation de la contribution de chaque modèle à chaque instant.

Dans un premier temps, une expression simple (II.29) a été présentée donnant lieu à ce qu'on a appelé validités simples. On a démontré que cette méthode convient aux systèmes dont la dynamique est décrite par des domaines de fonctionnement présentant une ou plusieurs zones de recouvrement. Elle devient, en revanche, moins efficace et peut induire des phénomènes de perturbation dans le cas où les classes de fonctionnement sont bien séparées avec, en général, un seul modèle valide à chaque instant et dont le résidu est très faible par rapport aux ceux des autres modèles. Afin de remédier à ce problème, on a introduit une opération supplémentaire permettant de renforcer les validités des modèles les plus pertinents à chaque instant, d'où le nom de validité renforcée.

Ainsi, le choix de la méthode de calcul des validités dépend de la dynamique du système en termes de dispersion des domaines de fonctionnement. Partant du fait qu'aucune information a priori n'est disponible sur ces domaines ni sur leur répartition, une solution consiste à se référer aux résultats de la classification des données [Elfelly et al, 08]. En effet, le but de cette dernière étape est de répartir les mesures collectées selon un certain nombre de groupes qui correspondent aux classes ou domaines de fonctionnement du système.

En se basant d'une part sur les constatations faites quant à l'utilisation des deux types de validités (simples et renforcées) précédemment définies, et sur l'observation des résultats de la classification en termes de structure et répartition des classes d'autre part, nous avons émis une hypothèse sur le choix de la méthode la plus convenable pour l'estimation des degrés de pertinence des différents modèles de la base. En effet, quand on constate un chevauchement entre les différentes classes, la méthode des validités simples semble être plus efficace puisqu'elle permet de tenir compte de la contribution des différents modèles. Dans ce cas, aucun modèle ne peut représenter idéalement le système à aucun instant. Mais lorsque les classes présentent peu de variations et sont bien séparées, la méthode des validités renforcées est mieux adaptée. Grâce à l'expression de renforcement, l'application de cette méthode renforce la contribution du modèle dominant qui représente au mieux le comportement du processus.

A travers l'étude de certains processus complexes, nous allons essayer de vérifier et de confirmer cette hypothèse en procédant par comparaison des résultats donnés par l'application des deux méthodes de calcul des validités.

II.4.3. Test et validation de la structure multimodèle proposée

Les différentes étapes, précédemment décrites, à savoir la détermination du nombre de modèles, la classification des données et la définition de la méthode d'estimation des validités, permettent de définir la structure multimodèle du système complexe considéré. Avant toute utilisation de cette structure que ce soit dans le but de l'étude ou la commande, il est indispensable de passer par une étape de test permettant d'évaluer l'efficacité et la précision de la modélisation ainsi proposée. Le principe consiste à vérifier que les modèles identifiés par l'approche envisagée, commandés par un mécanisme de décision, sont capables de reproduire le comportement du processus dans l'ensemble de ses domaines de fonctionnement. Pour ce faire, on va procéder à une comparaison entre la sortie du système réel et celle générée par la structure multimodèle en considérant plusieurs entrées différentes de celle qui a été utilisée pour la collecte de mesures. La figure suivante illustre le principe de la procédure de test et d'évaluation du schéma multimodèle élaboré dans le but de la modélisation ou la représentation d'un processus complexe.

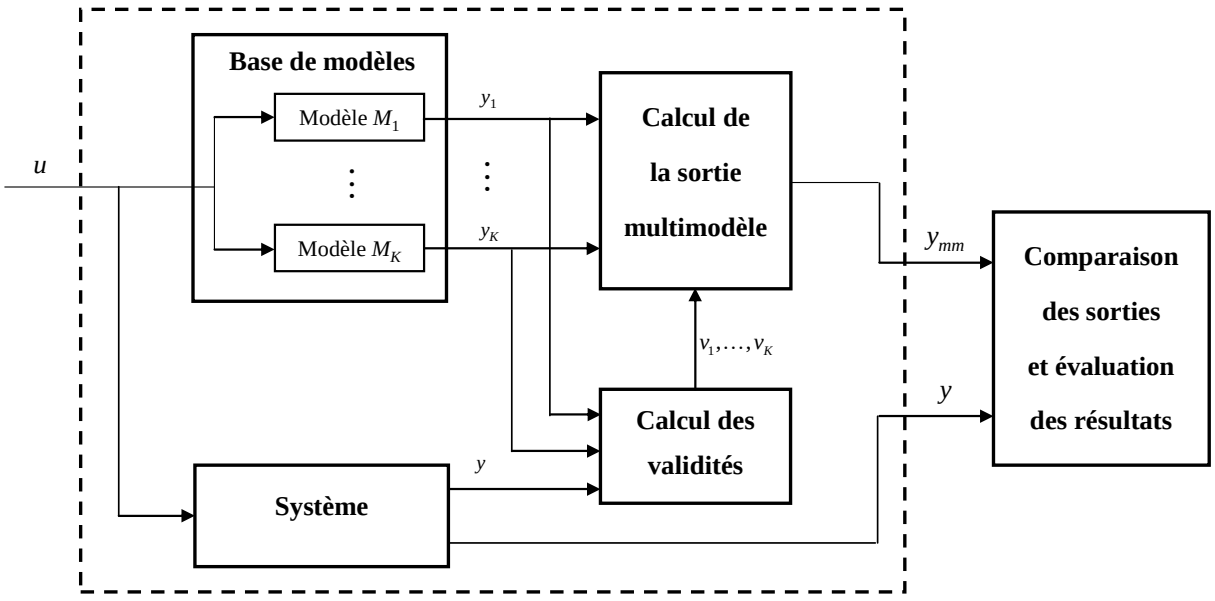


Figure II.3. Principe de l'évaluation de la structure multimodèle

Pour le calcul de la sortie multimodèle y_{mm}, on a opté pour une fusion entre les sorties partielles y_i ($i = 1,\ldots,K$), générées par les différents modèles de la base, pondérées par leurs coefficients de validité respectifs v_i, comme l'illustre la relation suivante :

$$y_{mm}(k) = \sum_{i=1}^{K} y_i(k)\, v_i(k)\,. \tag{II.32}$$

Ainsi, cette étape d'évaluation s'avère très intéressante dans le sens où elle permet de vérifier si la structure multimodèle élaborée répond bien à nos attentes en termes d'erreur tolérable par exemple notamment lorsqu'il s'agit de systèmes réclamant une grande précision dans leur modélisation.

II.5. Exemples de modélisation de processus complexes

Afin de souligner l'intérêt et la performance de l'approche de modélisation décrite dans ce chapitre, quelques exemples de simulation ont été considérés.

II.5.1. Exemple 1 : Système discret du second ordre avec des paramètres dépendant du temps

Le premier exemple est un système discret complexe dont l'évolution est décrite par les équations suivantes :

$$y(k) = -a_1(k)\, y(k-1) - a_2(k)\, y(k-2) + b_1(k)\, u(k-1) + b_2(k)\, u(k-2) \tag{II.33}$$

Les lois de variation des paramètres du processus en fonction du temps sont données par :

$$a_1(k) = 0.04 \sin(0.035k) - 0.8\ ; \tag{II.34a}$$

$$a_2(k) = 0.005 \sin(0.03k) + 0.1\ ; \tag{II.34b}$$

$$b_1(k) = 0.02 \sin(0.03k) + 0.5\ ; \tag{II.34c}$$

$$b_2(k) = 0.01 \sin(0.035k) + 0.2\,. \tag{II.34d}$$

- **Collecte de données système**

Avant de procéder à tout traitement, la première étape consiste à collecter l'ensemble des données systèmes qui constituent la base de notre approche de modélisation. Ces données doivent permettre de décrire la dynamique du processus dans l'ensemble de son espace de fonctionnement. Pour ce faire, nous avons sélectionné une entrée aléatoire uniforme (figure II.4) comme signal d'excitation.

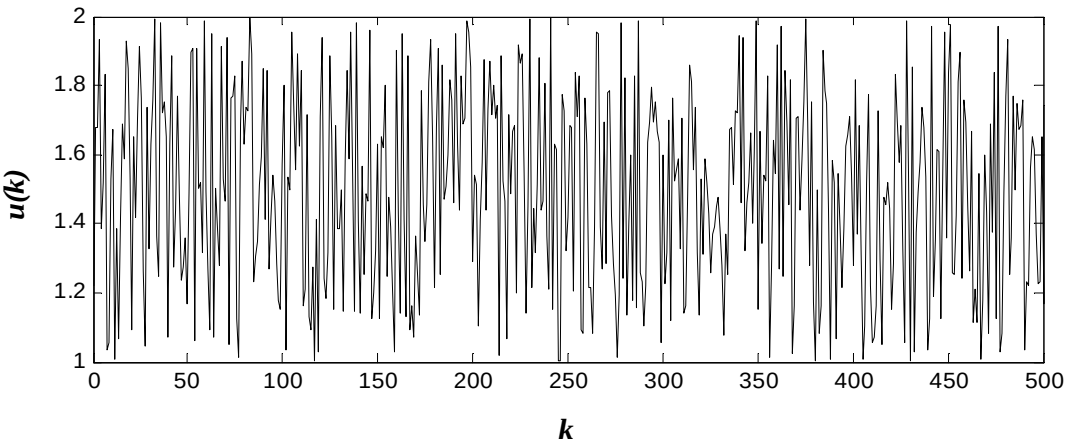


Figure II.4. Exemple 1 : Signal d'excitation

Quant aux variables mesurables, nous nous sommes limités, pour ce système, aux variables de sortie $y(k)$ et $y(k\text{-}1)$ qui se sont montrées suffisantes pour la description de l'évolution du processus. Après avoir défini une période d'échantillonnage convenable, on récupère l'ensemble des mesures des variables considérées à différents instants.

- **Détermination du nombre de modèles de la base**

L'ensemble des données numériques collectées sur le système va tout d'abord servir pour la détermination du nombre de modèles de la base, qui correspond au nombre de classes de fonctionnement, en se servant d'un réseau de neurones à apprentissage compétitif pénalisant le rival. Le nombre de neurones de la couche d'entrée du réseau est égal à deux, ce qui représente le nombre de variables mesurables considérées. La figure II.5 donne les résultats de cet apprentissage. En effet, avec cinq neurones dans la couche de sortie, deux centres se sont éloignés de l'ensemble des observations, ce qui permet de conclure que le nombre adéquat de classes est égal à trois.

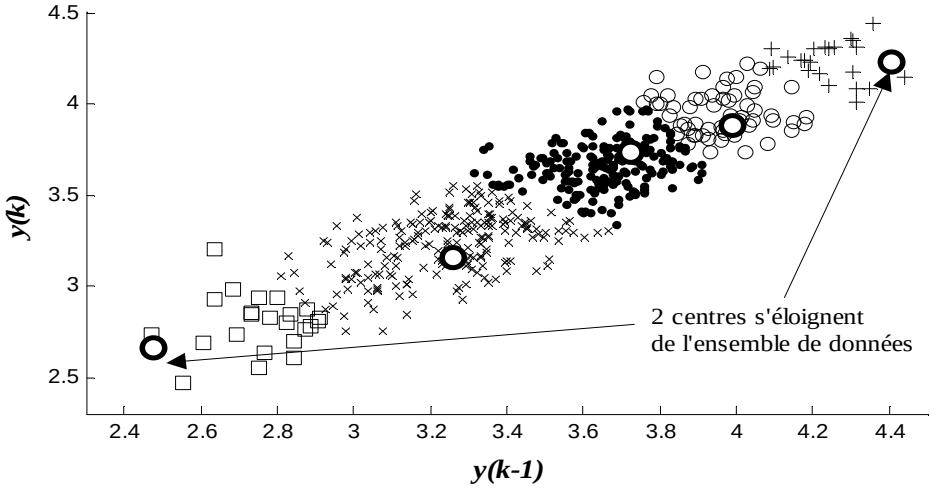


Figure II.5. Exemple 1 : Détermination du nombre de classes

- **Classification des données systèmes pour la sélection des domaines de fonctionnement**

Une fois le nombre de classes déterminé, on procède à une répartition des mesures par domaines de fonctionnement en mettant en œuvre les techniques de classification (carte de Kohonen, K-moyennes, K-moyennes floues) décrites dans le paragraphe II.2.3. La différence entre ces trois techniques n'est pas suffisamment visible sur les résultats de la classification. En considérant un réseau de Kohonen avec deux neurones pour la couche d'entrée et trois neurones pour la couche de sortie (ce qui correspond au nombre de classes identifié par l'apprentissage compétitif pénalisant le rival), les résultats de la classification sont donnés par la figure II.6.

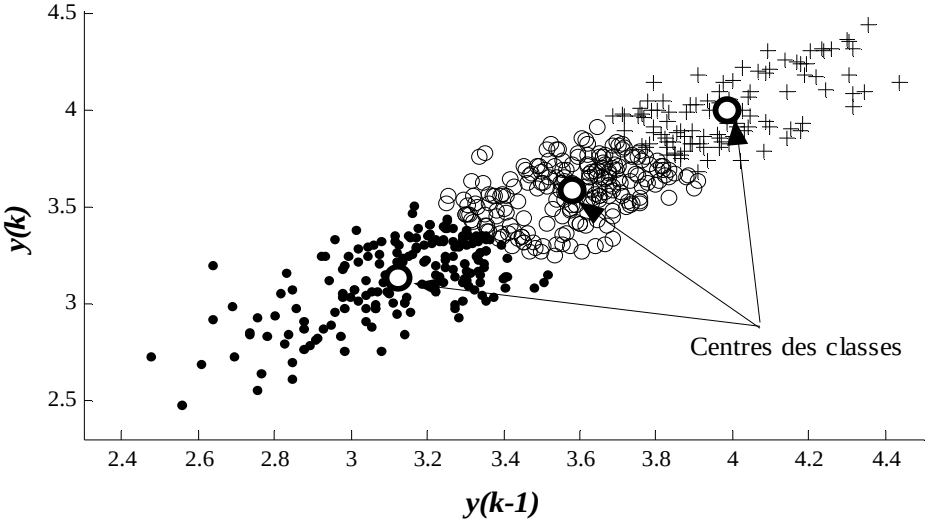


Figure II.6. Exemple 1 : Résultats de la classification (carte de Kohonen)

- **Identification structurelle et paramétrique des modèles**

A partir des mesures relatives aux différentes classes résultant de l'implémentation de chacune des techniques de classification, on procède à chaque fois à l'estimation des ordres et des paramètres de chacun des trois modèles de la base. Pour une classification par la carte de Kohonen par exemple, les ordres des trois modèles, estimés par la méthode du test du rapport des déterminants instrumentaux, sont égaux à deux. Ce résultat est donc en accord avec l'ordre réel du système qui lui aussi est égal à deux. Ensuite, l'application de la méthode des moindres carrés récursifs nous a permis d'écrire les fonctions de transfert des trois modèles de la base :

$$F_1(z^{-1}) = \frac{0.33549\, z^{-1} - 0.047793\, z^{-2}}{1 - 0.83697\, z^{-1} - 0.039754\, z^{-2}} \ ; \tag{II.35}$$

$$F_2(z^{-1}) = \frac{0.41086\, z^{-1} - 0.021659\, z^{-2}}{1 - 0.86336\, z^{-1} - 0.055068\, z^{-2}} \ ; \tag{II.36}$$

$$F_3(z^{-1}) = \frac{0.36985\, z^{-1} - 0.0079934\, z^{-2}}{1 - 0.9234\, z^{-1} - 0.060727\, z^{-2}} \ ; \tag{II.37}$$

- **Sélection de la méthode de classification**

Dans le paragraphe II.2.3.5, nous avons fait une hypothèse sur le choix de la méthode la plus convenable à utiliser pour la classification de données système. Dans le but de confirmer cette hypothèse, nous allons procéder à une comparaison des résultats obtenus suite à l'application de chacune des trois techniques de classification proposées. Pour ce faire, nous allons considérer la méthode des validités renforcées pour l'estimation de la contribution de chaque modèle à la génération de la sortie multimodèle. Ainsi, pour chaque algorithme de classification, en se référent à la base de modèles construite, la sortie multimodèle est calculée (par fusion entre les sorties des modèles pondérées par les validités renforcées) et ensuite comparée à la sortie réelle du système.

Les simulations ont montré qu'il n'y a pas de différence remarquable entre les résultats obtenus en utilisant la carte de Kohonen et ceux donnés par l'application de l'algorithme des K-moyennes. Il reste donc à comparer l'efficacité de l'une de ces deux méthodes à l'algorithme des K-moyennes floues.

En considérant par exemple la séquence d'entrée suivante :

$$u(k) = 1 + \sin(0.06k), \tag{II.38}$$

les résultats de la comparaison sont donnés par la figure II.7 où :

- y : sortie réelle du système ;
- y_{koh}, y_{kmf} : sorties multimodèles obtenues par application des techniques de classification respectives : carte de Kohonen et K-moyennes floues ;
- e_{koh}, e_{kmf} : erreurs absolues entre la sortie réelle et les sorties multimodèles calculées en utilisant respectivement le réseau de Kohonen et les K-moyennes floues comme technique de classification ;

$$e_{koh}(k) = \left| y(k) - y_{koh}(k) \right| \ ; \tag{II.39a}$$

$$e_{kmf}(k) = \left| y(k) - y_{kmf}(k) \right|. \tag{II.39b}$$

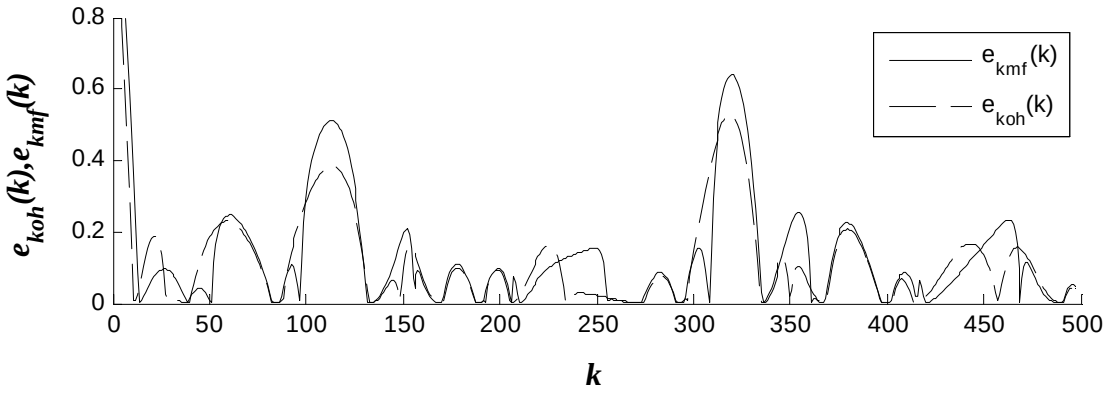


Figure II.7. Exemple 1 : Erreurs absolues entre la sortie réelle et les sorties multimodèles (sélection de la méthode de classification)

Il apparaît clairement que la courbe e_{kmf} présente les pics les plus élevés. Par conséquent, on peut conclure que les résultats obtenus par application d'une classification par la carte de Kohonen ou les K-moyennes sont meilleurs que ceux résultants de la mise en œuvre de l'algorithme des K-moyennes floues, ce qui permet de confirmer une partie de l'hypothèse émise dans le paragraphe II.2.3.5 affirmant qu'en présence d'un faible recouvrement entre les classes (ce qui le cas des résultats données par la figure II.5), les cartes de Kohonen ou les K-moyennes floues paraissent plus efficaces en termes de classification de

données. Entre ces deux techniques, nous avons choisi, pour ce système, de faire appel à une répartition des données système par un réseau de Kohonen étant donné que son utilisation reste préférable par rapport à l'algorithme des K-moyennes (qui présente l'inconvénient de la dépendance aux conditions initiales) dans le cas où on dispose d'un volume raisonnable de données collectées sur le système et d'un nombre peu élevé de classes, ce qui correspond aux observations faites lors des deux premières étapes (collecte de données et détermination du nombre de classes).

- **Sélection de la méthode de calcul des validités**

Afin de générer la sortie multimodèle, une fusion entre les sorties des différents modèles pondérées par les coefficients de validité respectifs, a été envisagée. A cet effet, deux méthodes permettant l'estimation, à chaque instant, de la contribution de chaque modèle, ont été présentées (section II.4.1). A travers l'observation de la dispersion des domaines de fonctionnement après l'étape de classification, une hypothèse sur le choix de la méthode la plus appropriée a été proposée dans la section II.4.2.

Après la classification des données systèmes par la carte de Kohonen et l'identification des différents modèles de la base, nous procédons au calcul de la sortie multimodèle en mettant en œuvre les deux méthodes de calcul des validités. Dans le but de choisir la méthode la plus appropriée et vérifier l'hypothèse émise à ce propos, nous allons comparer les résultats donnés par chacune des deux méthodes. Pour ce faire, on considère par exemple la séquence d'entrée suivante :

$$u(k) = 1 + \sin(0.08k). \tag{II.40}$$

Les résultats sont donnés par les figures II.8, II.9 et II.10 avec :

- y_{simp}, y_{renf} : sorties multimodèles obtenues en utilisant respectivement la méthode des validités simples et celle des validités renforcées :

$$y_{simp}(k) = \sum_{i=1}^{3} y_i(k)\, v_i^{simp}(k) \ ; \tag{II.41a}$$

$$y_{renf}(k) = \sum_{i=1}^{3} y_i(k)\, v_i^{renf}(k) \ ; \tag{II.41b}$$

- y_i : sortie du modèle M_i ;

- e_{simp}, e_{renf} : erreurs absolues entre la sortie réelle et les sorties multimodèles :

$$e_{simp}(k) = \left| y(k) - y_{simp}(k) \right|; \tag{II.42a}$$

$$e_{renf}(k) = \left| y(k) - y_{renf}(k) \right|. \tag{II.42b}$$

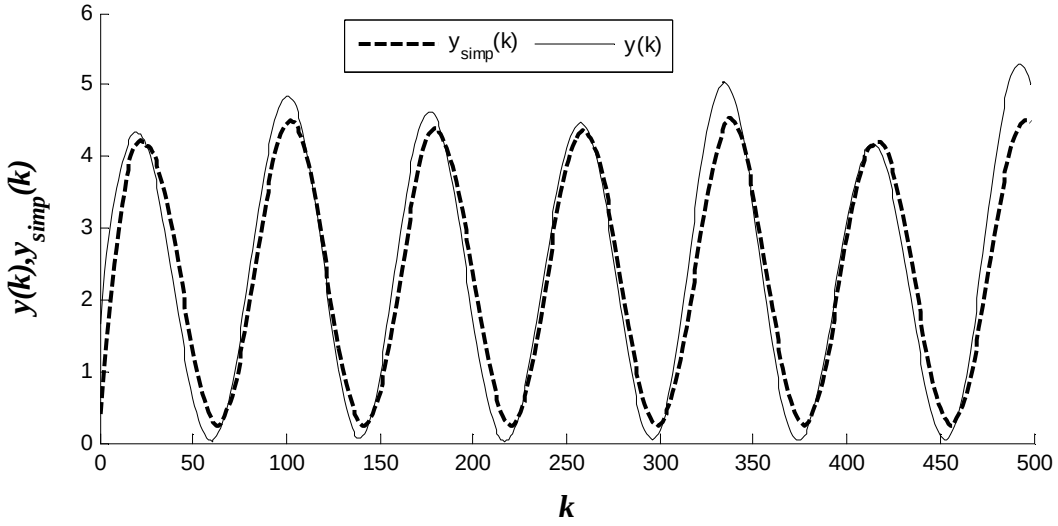


Figure II.8. Exemple 1 : Sorties réelle et multimodèle (validités simples)

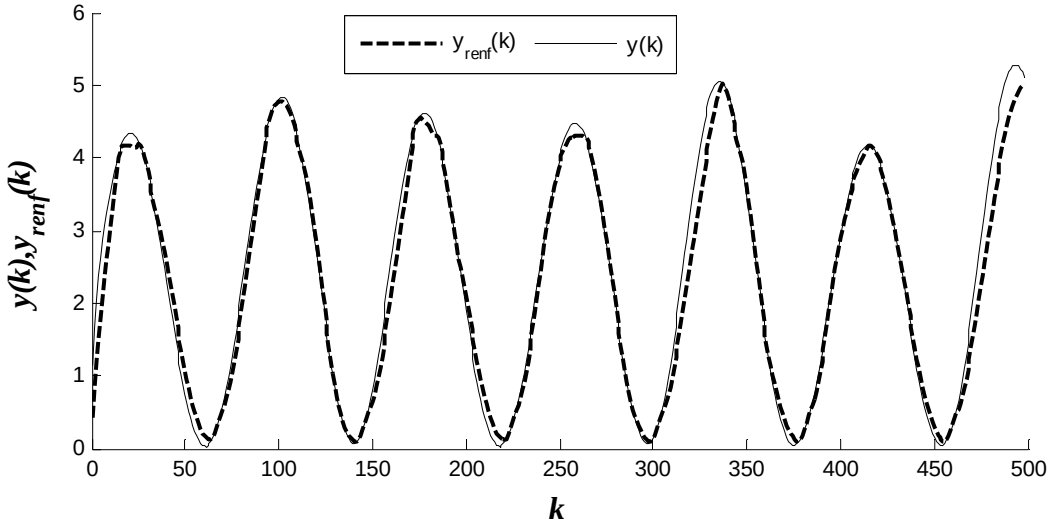


Figure II.9. Exemple 1 : Sorties réelle et multimodèle (validités renforcées)

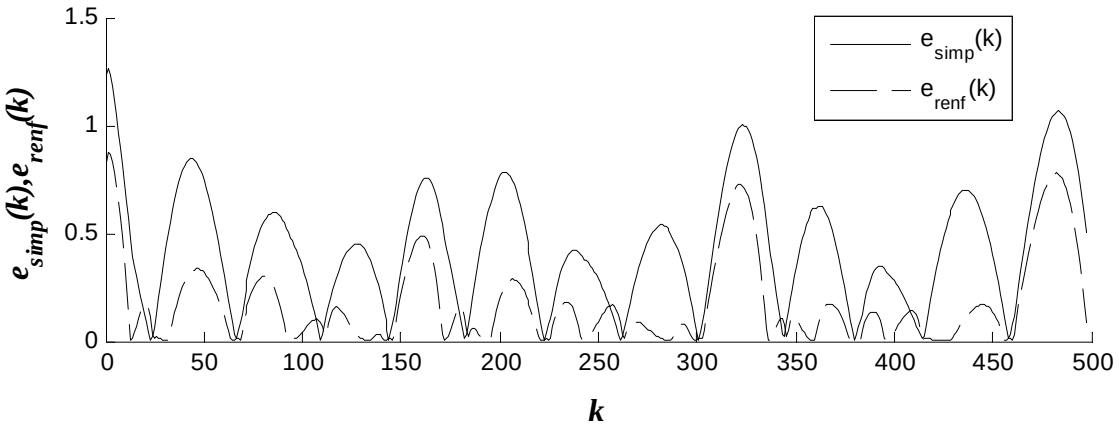


Figure II.10. Exemple 1 : Erreurs absolues entre la sortie réelle et les sorties multimodèles (sélection de la méthode de calcul des validités)

Ces résultats montrent que la méthode des validités renforcées est mieux adaptée pour ce système, ce qui va dans le sens de l'hypothèse émise dans la mesure où les résultats de la classification (figure II.6) présentent des classes bien séparées avec très peu de recouvrement. Ainsi, la méthode des validités renforcées permet d'accentuer, à chaque instant, la contribution du ou des modèles les plus pertinents.

- **Validation de la structure multimodèle proposée**

Afin d'évaluer l'efficacité de la structure multimodèle ainsi établie, on se propose d'appliquer au système des entrées différentes de celle qui a servi pour la collecte de données. Ensuite, la sortie multimodèle est calculée à chaque instant en mettant en œuvre la méthode des validités renforcées puis est comparée à la sortie réelle. En considérant par exemple la séquence d'entrée de l'expression (II.38), les résultats sont donnés par la figure II.11 où y représente la sortie réelle et y_{mm} la sortie multimodèle obtenue en utilisant la carte de Kohonen comme algorithme de classification et les validités renforcées comme méthode d'estimation des degrés de pertinence des modèles.

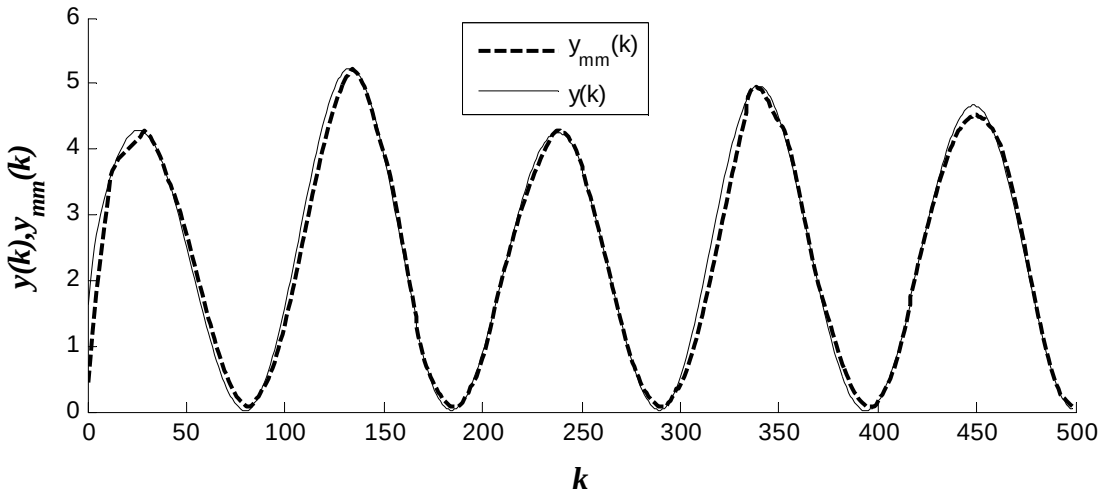


Figure II.11. Exemple 1 : Sorties réelle et multimodèle

A travers les résultats obtenus, on constate que l'implémentation de l'approche proposée permet une bonne modélisation du système considéré. En effet, sur les figures II.9 et II.11, la sortie multimodèle suit la sortie réelle avec une très faible erreur.

II.5.2. Exemple 2 : Système du second ordre avec des paramètres variant en fonction de l'entrée et de la sortie

Afin de démontre que l'approche proposée est applicable aussi bien aux systèmes continus qu'aux systèmes discrets, on considère, comme deuxième exemple de simulation, un système continu dont l'évolution est régie par les équations suivantes :

$$a_0 y + a_1 \dot{y} + \ddot{y} = u + b\dot{u} \ ; \tag{II.43}$$

avec :

$$a_0(y) = 0.3 + sigm(y-2); \tag{II.44a}$$

$$a_1(y) = sat(y^2); \tag{II.44b}$$

$$b(u) = sat(u); \tag{II.44c}$$

sigm est la fonction sigmoïde ; *sat* est la fonction saturation.

Le système considéré est complexe et fortement non linéaire avec des paramètres variant en fonction de l'entrée et de la sortie, ce qui fait de la modélisation une tâche difficile.

- **Collecte de données système**

Le signal retenu pour l'excitation du système (figure II.12) est une sinusoïde perturbée par un signal aléatoire uniforme, « plus riche » qu'un simple signal aléatoire, ce qui permet une meilleure prise en compte de l'aspect complexe et non linéaire du système. Il est à signaler qu'il a fallu aussi bien ajuster les paramètres du signal d'excitation (fréquence, amplitude) pour aboutir à de bons résultats.

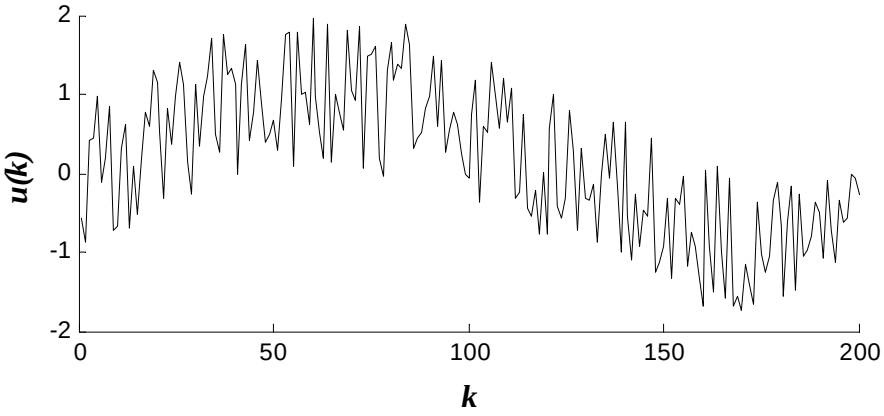


Figure II.12. Exemple 2 : Signal d'excitation

D'autre part, afin de décrire correctement les dynamiques de fonctionnement du système, on a dû augmenter le nombre de variables servant pour la classification par rapport au premier exemple. En effet, il a fallu considérer, non seulement les mesures de la sortie $y(k)$ et de la sortie retardée $y(k\text{-}1)$, mais aussi les mesures de l'entrée $u(k\text{-}1)$. Ceci peut bien s'expliquer par le fait que le système considéré est complexe et fortement non linéaire et surtout présentant des paramètres dépendant de l'entrée et de la sortie. Une discrétisation des signaux d'entrée et de sortie, avec une période d'échantillonnage adéquate, était nécessaire pour la collecte des mesures des variables considérées ($u(k\text{-}1)$, $y(k\text{-}1)$ et $y(k)$).

- **Détermination du nombre de modèles de la base**

En se servant de l'ensemble de mesures collectées sur le système, le nombre de classes de fonctionnement est déterminé en utilisant un réseau de neurones à apprentissage compétitif pénalisant le rival avec trois neurones pour la couche d'entrée (ce qui correspond au nombre de variables considérées pour la collecte de mesures) et K neurones pour la couche de sortie. En choisissant n'importe quelle valeur de K supérieure ou égale à quatre, le nombre de classes

déterminé reste toujours égal à quatre (figure II.13), l'augmentation de *K* ne modifiant donc pas le nombre de classes obtenues.

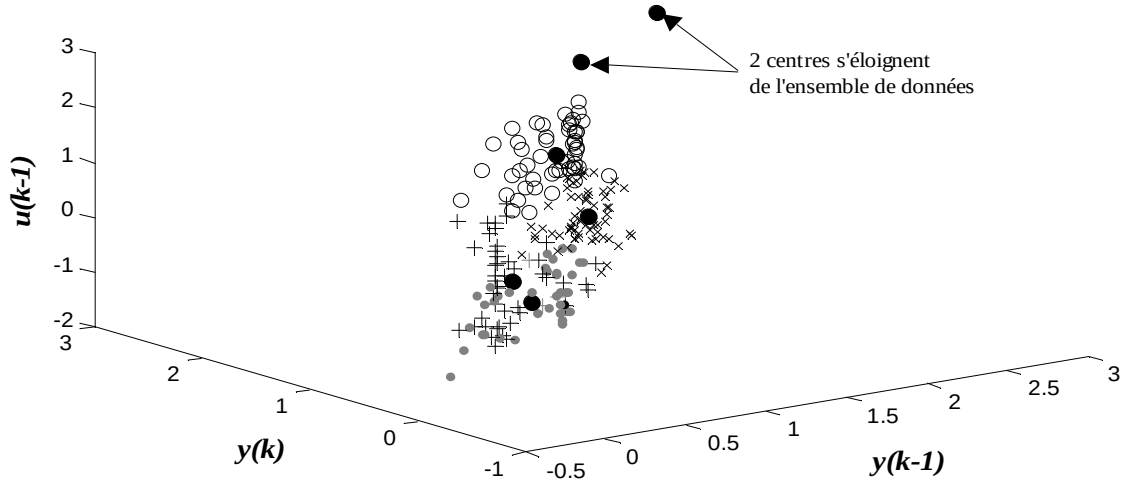


Figure II.13. Exemple 2 : Détermination du nombre de classes

- **Classification des données systèmes et identification des modèles de la base**

Après la détermination du nombre de modèles, les données système sont réparties en classes en faisant appel à chaque fois à l'une des trois techniques de classification : carte de Kohonen, K-moyennes et K-moyennes floues. Pour le cas de l'algorithme des K-moyennes floues, les résultats sont donnés par la figure II.14 sur laquelle on distingue effectivement quatre ensembles de données.

Pour chacune des techniques de classification et en partant des différents ensembles de données, l'application de la méthode des RDI pour l'estimation des ordres des modèles et celle des moindres carrés récursifs pour leur identification a permis d'écrire quatre fonctions de transfert ayant des ordres différents. Cet aspect est tout à fait remarquable dans une perspective de commande multimodèle puisque la structure du modèle – et donc du contrôleur – pourra varier selon la zone considérée.

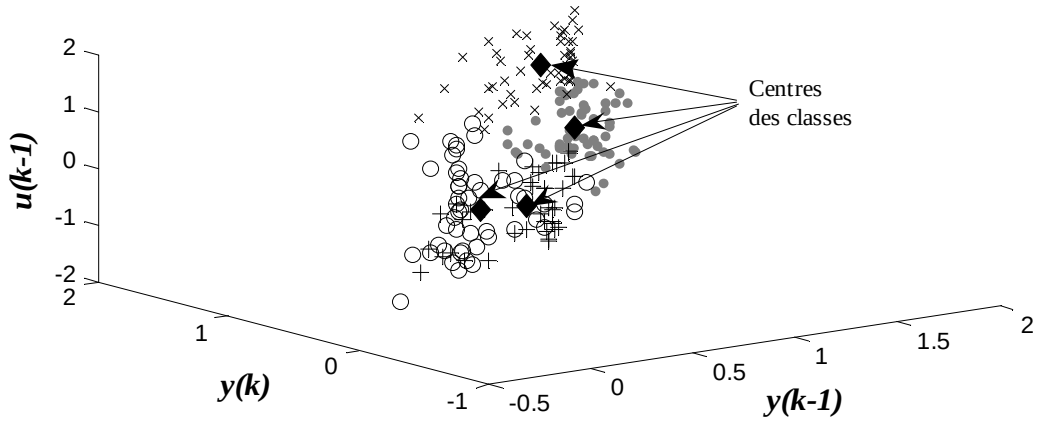


Figure II.14. Exemple 2 : Résultats de la classification (K-moyennes floues)

- **Sélection de la méthode de classification**

Afin de sélectionner la méthode de classification la plus appropriée pour le système considéré, nous allons procéder à la comparaison des résultats obtenus suite à l'application de chacune des trois méthodes. Dans un premier temps, nous allons opter pour la méthode des validités simples pour l'estimation de la contribution, à chaque instant, des différents modèles de la base à la génération de la sortie multimodèle.

Ainsi, pour chaque technique de classification, la base de modèles est élaborée et la sortie multimodèle est calculée à travers une fusion des sorties des modèles pondérées par leurs validités simples respectives. Comme pour le premier exemple, on n'a pas constaté une différence remarquable entre les résultats obtenus suite à l'application de l'algorithme des K-moyennes et ceux donnés par l'utilisation des cartes de Kohonen. On s'est contenté alors de comparer les résultats fournis par les méthodes des K-moyennes et des K-moyennes floues.

En considérant la séquence d'entrée suivante :

$$u(k) = 1.2 + 1.5\sin(2k), \tag{II.45}$$

la figure II.15 donne les erreurs absolues e_{km} et e_{kmf} entre la sortie réelle du système et les sorties multimodèles calculées en utilisant respectivement les K-moyennes et les K-moyennes floues :

$$e_{km}(k) = \left| y(k) - y_{km}(k) \right| \ ; \tag{II.46a}$$

$$e_{kmf}(k) = \left| y(k) - y_{kmf}(k) \right| \ ; \qquad \text{(II.46b)}$$

avec :

- y : sortie réelle du système ;
- y_{km}, y_{kmf} : sorties multimodèles obtenues par application des techniques de classification respectives : K-moyennes et K-moyennes floues.

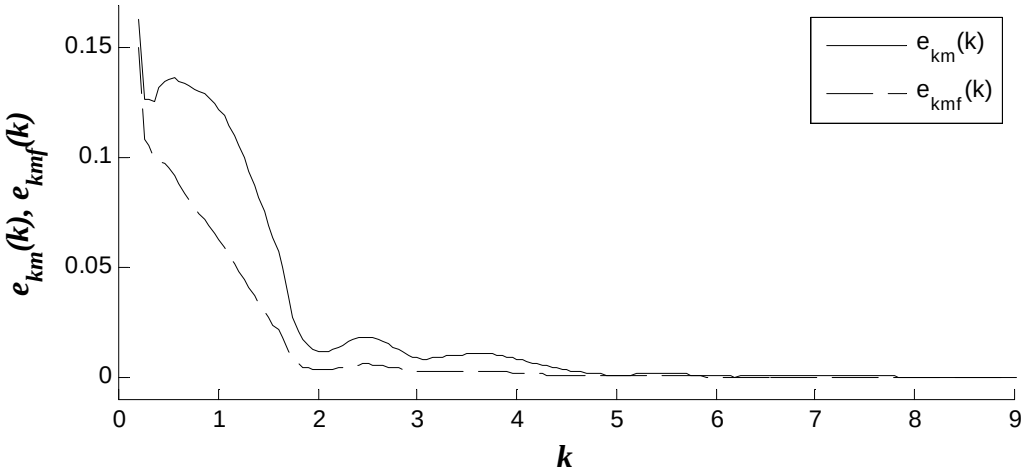


Figure II.15. Exemple 2 : Erreurs absolues entre la sortie réelle et les sorties multimodèles (sélection de la méthode de classification)

On constate alors que l'erreur e_{kmf} est inférieure à l'erreur e_{km}. Par conséquent, on peut conclure que, contrairement au premier exemple, la classification des mesures par l'algorithme des K-moyennes floues donne les meilleurs résultats, ce qui permet de valider la deuxième partie de notre hypothèse (paragraphe II.2.3.5) qui affirme que lorsqu'il s'agit d'un recouvrement assez important entre les classes (ce qui est le cas de la figure II.13), l'algorithme des K-moyennes floues est le plus convenable pour la classification de données systèmes.

- **Sélection de la méthode de calcul des validités**

En retenant l'algorithme des K-moyennes floues comme technique de classification de données système, la base de modèles est construite. Il reste donc à sélectionner la méthode adéquate de calcul des validités. Pour cela, nous proposons de comparer les résultats donnés

par les deux méthodes (validités simples et validités renforcées). A cet effet, nous considérons par exemple le signal d'entrée suivant :

$$u(k) = 1 + \sin(k)\,. \tag{II.47}$$

Les résultats de la comparaison sont donnés par les figures II.16 et II.17 avec :

- y_{simp}, y_{renf} : sorties multimodèles obtenues en utilisant respectivement la méthode des validités simples et celle des validités renforcées :

$$y_{simp}(k) = \sum_{i=1}^{4} y_i(k)\, v_i^{simp}(k)\ ; \tag{II.48a}$$

$$y_{renf}(k) = \sum_{i=1}^{4} y_i(k)\, v_i^{renf}(k)\ ; \tag{II.48b}$$

- y_i : sortie du modèle M_i ;
- e_{simp}, e_{renf} : erreurs absolues entre la sortie réelle et les sorties multimodèles données par les expressions (II.42a) et (II.42b).

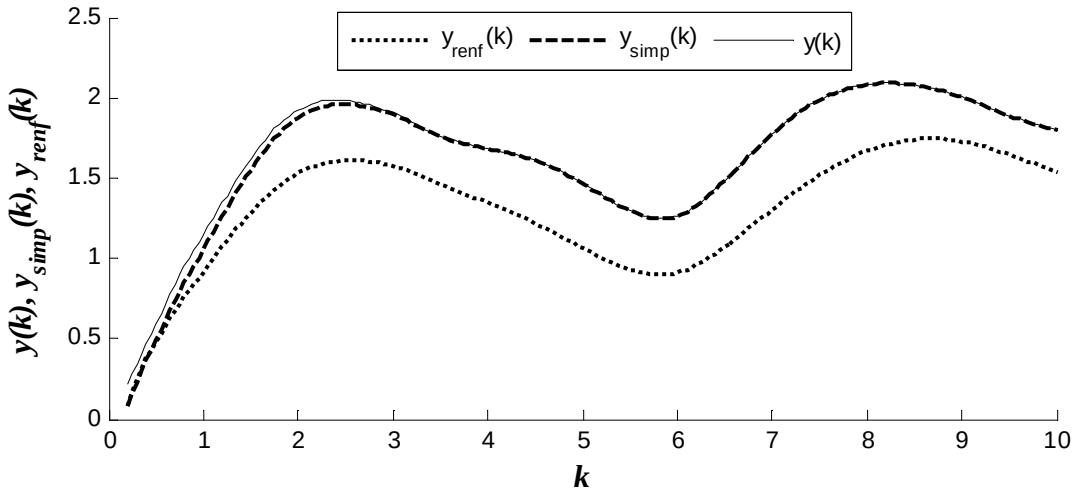


Figure II.16. Exemple 2 : Sorties réelle et multimodèles (validités simples/renforcées)

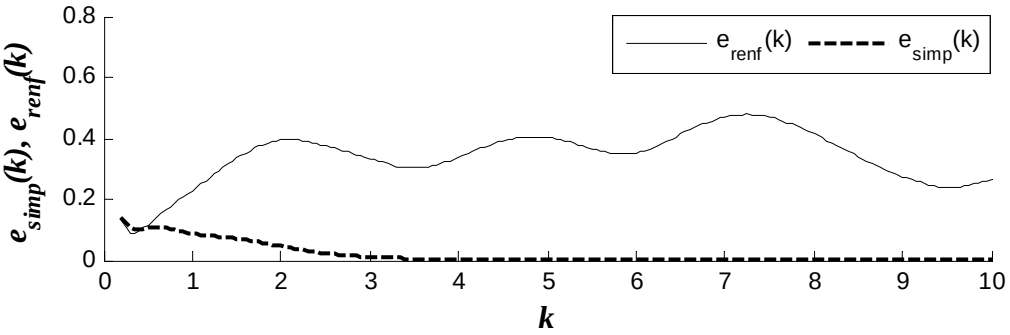


Figure II.17. Exemple 2 : Erreurs absolues entre la sortie réelle et les sorties multimodèles (sélection de la méthode de calcul des validités)

Contrairement au premier exemple, on constate ici que la méthode des validités simples est plus performante que celle des validités renforcées. En effet, les figures II.16 et II.17 montrent que la sortie multimodèle y_{simp}, déduite de la fusion des sorties partielles pondérées par les validités simples, suit presque parfaitement la sortie réelle avec une erreur e_{simp} presque nulle comparée à l'erreur e_{renf} qui est relativement importante. Par conséquent, la deuxième partie de l'hypothèse, portant sur le choix des validités à utiliser (section II.4.2), est vérifiée, car on peut constater que les résultats de la classification (figure II.14) se présentent sous forme d'une répartition des données en classes présentant un chevauchement assez important. Cet aspect favorise l'utilisation de la méthode des validités simples qui permet de tenir compte, à chaque instant, de la contribution de tous les modèles.

- **Validation de la structure multimodèle proposée**

Une fois les choix des méthodes adéquates de classification et de calcul des validités faits, la structure multimodèle élaborée est testée à travers le calcul de la sortie multimodèle pour différentes séquences d'entrée. En appliquant par exemple le signal d'entrée donné par l'expression (II.45), les évolutions des sorties réelle y et multimodèle y_{mm} sont données par la figure II.18.

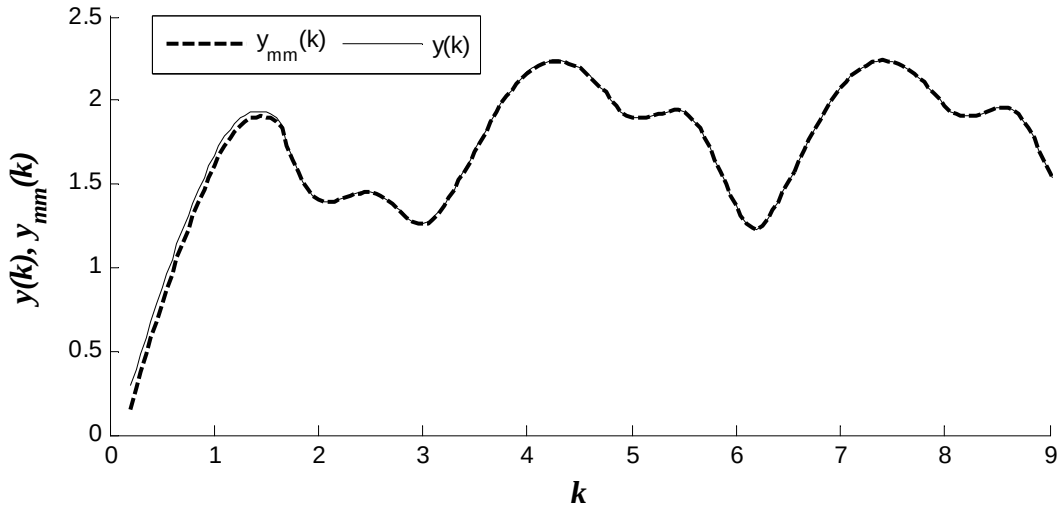


Figure II.18. Exemple 2 : Sorties réelle et multimodèle

Les résultats ainsi obtenus témoignent de la précision et de la performance de la structure multimodèle proposée pour la représentation du système considéré.

D'autre part, afin de mettre en évidence l'intérêt et la robustesse de l'approche multimodèle proposée, nous avons procédé à la mise en œuvre de l'approche classique de modélisation par identification d'un modèle global unique et ensuite à la comparaison des résultats. En effet, en se servant de l'ensemble des mesures collectées sur le système et par application de la méthode des RDI et celle des moindres carrés récursifs pour l'identification paramétrique, la fonction du transfert du modèle global est la suivante :

$$F(z^{-1}) = \frac{0.03601\, z^{-1} + 0.0035236\, z^{-2}}{1 - 0.55726\, z^{-1} - 0.42595\, z^{-2}}. \qquad \text{(II.49)}$$

En considérant le modèle global identifié d'une part et la base de modèles élaborée par l'implémentation de l'approche décrite dans ce chapitre d'autre part, leurs réponses respectives y_g et y_{mm} à la séquence d'entrée de l'expression (II.47) sont présentées sur la figure II.19.

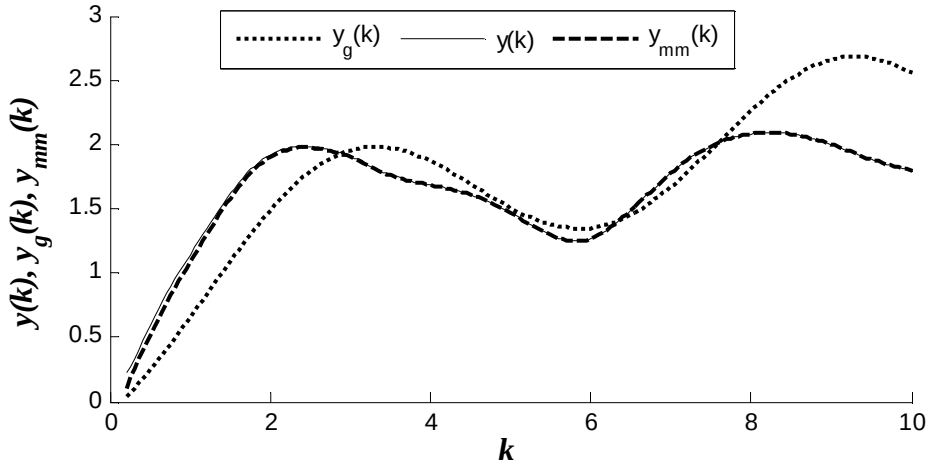


Figure II.19. Exemple 2 : Comparaison entre l'approche multimodèle proposée et l'approche classique de modélisation

A travers cette étude, on déduit que l'approche multimodèle proposée pour la représentation de systèmes complexes, comparée à l'approche classique de modélisation, offre une meilleure précision avec des résultats assez satisfaisants.

II.5.3. Exemple 3 : Modélisation d'un système mécanique : bras manipulateur

Afin de souligner l'intérêt de l'approche proposée dans ce chapitre, nous avons essayé de l'appliquer pour la modélisation d'un système mécanique. Il s'agit d'un bras manipulateur constitué de deux bras en rotation (figure II.20) [Chen, 06] [Elfelly et al, 10b] [Salcedo et al, 08] qui peut être décrit par le système suivant :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\dfrac{Mgl}{I}\sin x_1 - \dfrac{k}{I}(x_1 - x_3) \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \dfrac{k}{J}(x_1 - x_3) + \dfrac{1}{J}u \end{cases} \qquad \text{(II.50)}$$

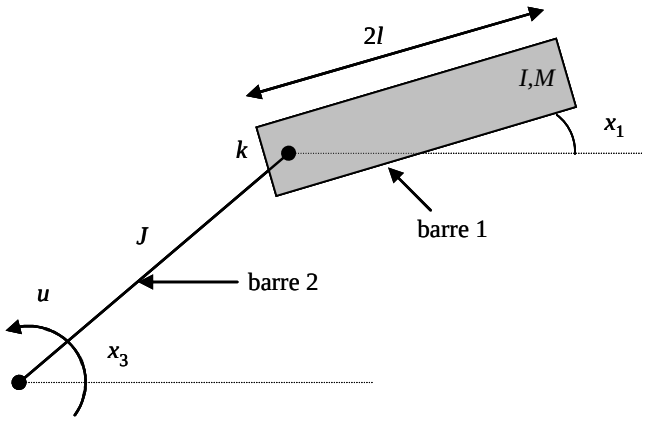


Figure II.20. Exemple 3 : Système mécanique (bras manipulateur)

Le système considéré est composé de deux barres en rotation avec :

- x_1, x_2 : position angulaire et vélocité angulaire de la barre 1 ;
- x_3, x_4 : position angulaire et vélocité angulaire de la barre 2 ;
- u : couple appliqué à la barre 2 ;
- $g = 9.8$ m·s^{-2} : constante de gravité ;
- $I = 1$ kg·m^2 : moment d'inertie de la barre 1 ;
- $J = 1$ kg·m^2 : moment d'inertie de la barre 2 ;
- $l = 1$ m : moitié de la longueur de la barre 1 ;
- $M = 1$ kg : masse de la barre 1 ;
- $k = 1$ Nm·rad^{-1} : rigidité élastique au niveau du lien entre les barres 1 et 2.

La forme normale du modèle non linéaire du système peut s'écrire comme suit :

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = z_3 \\ \dot{z}_3 = z_4 \\ \dot{z}_4 = a(z) + b(z)u \end{cases} ; \qquad \text{(II.51)}$$

avec :

$$a(z) = \frac{Mgl}{I}\sin(z_1)z_2^2 - \frac{Mgl}{I}\cos(z_1)z_3 - \left(\frac{K}{I} + \frac{K}{J}\right)z_3 - \frac{KMgl}{IJ}\sin(z_1) \; ; \qquad \text{(II.52a)}$$

$$b(z) = \frac{K}{IJ} . \qquad \text{(II.52b)}$$

Les variables z_i sont liées à aux variables x_i à travers les équations suivantes :

$$\begin{cases} x_1 = z_1 \\ x_2 = z_2 \\ x_3 = z_1 + \dfrac{I}{K} z_3 + \dfrac{Mgl}{K} \sin(z_1) \\ x_4 = z_2 + \dfrac{I}{K} z_4 + \dfrac{Mgl}{K} \cos(z_1) z_2 \end{cases} \quad \text{(II.53)}$$

Dans le reste de l'étude, u va être considérée comme l'entrée et x_1 la sortie du système.

La première étape de l'implémentation de l'approche multimodèle proposée consiste à appliquer au système un signal d'excitation adéquat afin de collecter les mesures des variables $x_1(k)$ et $x_1(k\text{-}1)$ à différents instants. Ces données ainsi que les mesures de l'entrée $u(k\text{-}1)$ sont tout d'abord utilisées pour la détermination du nombre de classes de fonctionnement en utilisant un réseau de neurones à apprentissage compétitif pénalisant le rival. La figure II.21, qui présente les résultats de l'apprentissage, montre qu'en partant de six neurones dans la couche de sortie, deux centres se trouvent repoussés de l'espace d'observation, ce qui permet de conclure que le nombre adéquat de classes est égal à quatre.

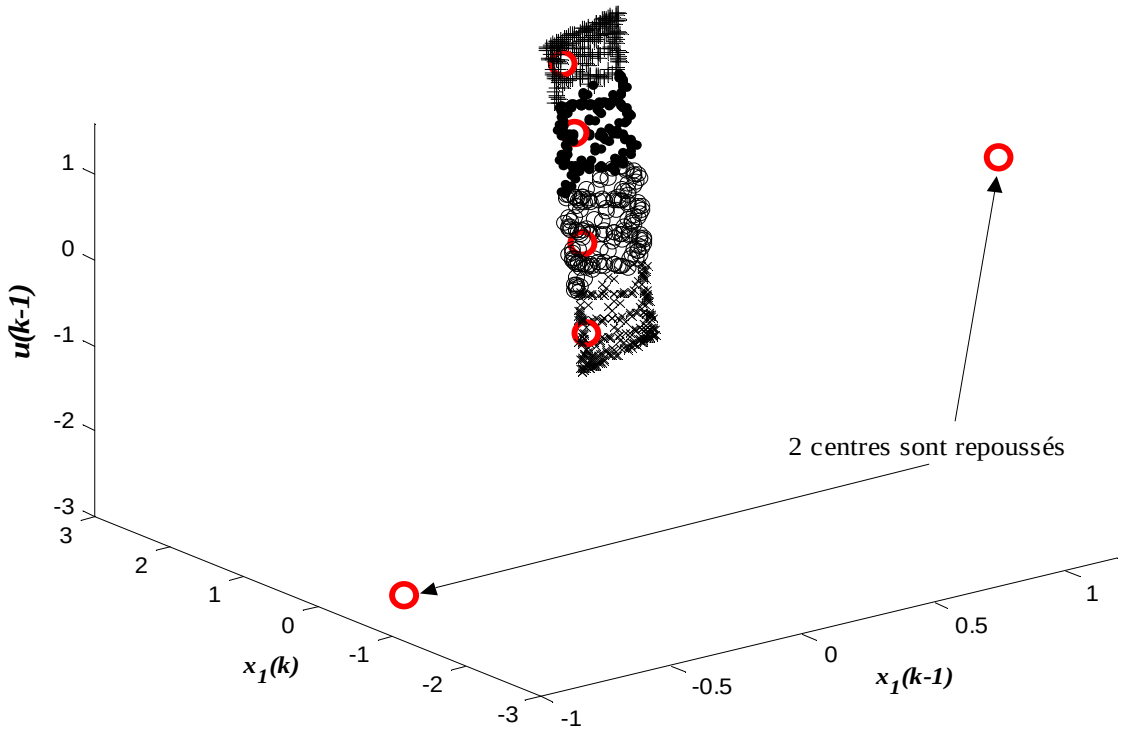


Figure II.21. Exemple 3 : Détermination du nombre de classes

Une fois le nombre de modèles déterminé, une procédure de comparaison des résultats donnés par les trois techniques de classification a permis de sélectionner l'algorithme des K-moyennes, ce qui permet encore une fois de confirmer l'hypothèse émise à cet effet dans la mesure où on a constaté, à travers les résultats de la première étape (classification par l'apprentissage compétitif pénalisant le rival), que les groupes de données générés sont bien séparés et présentent peu de recouvrement. D'autre part, contrairement au premier exemple de simulation, pour des raisons de simplicité, nous avons privilégié la méthode des K-moyennes par rapport aux cartes de Kohonen puisque l'on se retrouve en présence d'un grand volume de données (comme on peut le voir sur la figure II.21) et que l'on a constaté que l'utilisation d'une carte de Kohonen, dans ce cas, nécessite un paramétrage minutieux et plusieurs itérations et donc un temps d'exécution assez long avant d'aboutir aux résultats. La figure II.22 donne les résultats de la classification des données systèmes par l'algorithme des K-moyennes.

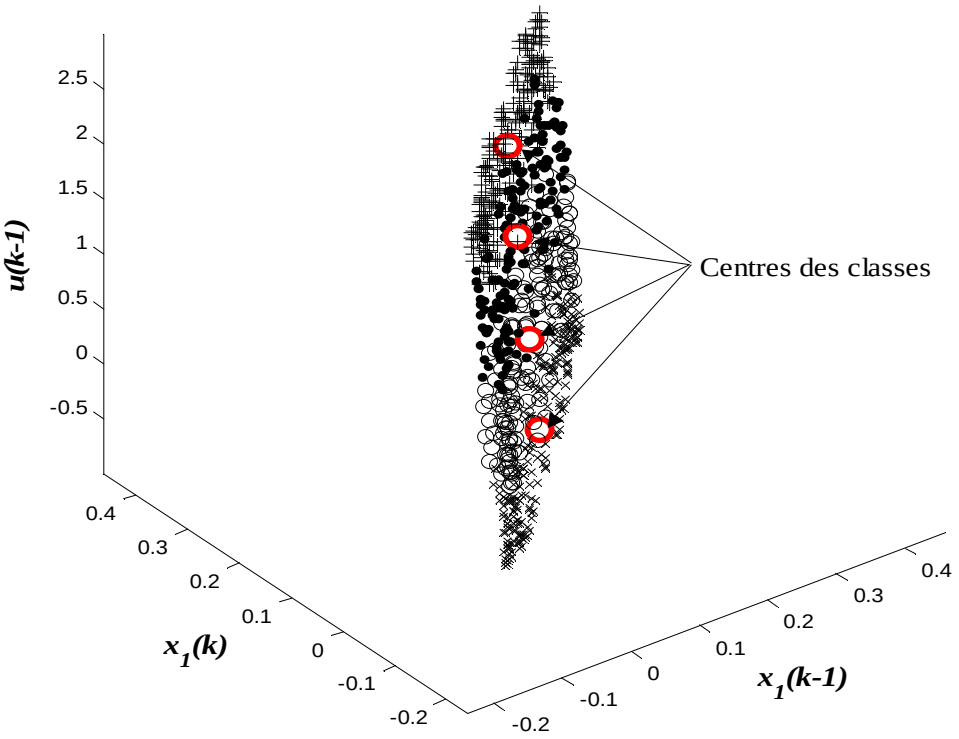


Figure II.22. Exemple 3 : Résultats de la classification (K-moyennes)

En se référant à chacun des groupes de données relatifs aux différentes classes générées, les ordres et les paramètres des quatre modèles de la base sont estimés en utilisant respectivement la méthode des RDI et celle des moindres carrés récursifs.

En adoptant la même démarche que dans les deux premiers exemples, nous avons sélectionné la méthode des validités renforcées comme étant la mieux adaptée pour l'estimation, à chaque instant, de la pertinence de chacun des modèles de la base, ce qui coïncide avec les propos de l'hypothèse portant sur la sélection de la méthode de calcul des validités partant du fait que les résultats de la classification (figure II.22) présentent des groupes assez bien séparés avec peu de variations sur les frontières.

Afin de mettre en évidence la capacité des modèles identifiés de reproduire le fonctionnement du système dans différents domaines, plusieurs séquences d'entrée ont été considérées, parmi lesquelles deux sont ici présentées :

$$u_1(k) = 0.5\sin(0.5k)\ ; \tag{II.54}$$

$$u_2(k) = 1 + 0.5\sin(0.6k)\,. \tag{II.55}$$

En se servant de la base de modèles élaborée, la sortie multimodèle est calculée pour chaque séquence d'entrée (par fusion entre les sorties des modèles pondérées par les validités renforcées) et ensuite comparée à la sortie réelle du système. Les résultats sont donnés par les figures II.23 et II.24 où x_1 représente la sortie réelle et x_{1mm} la sortie multimodèle :

$$x_{1mm}(k) = \sum_{i=1}^{4} x_{1i}(k)\, v_i^{renf}(k)\,. \tag{II.56}$$

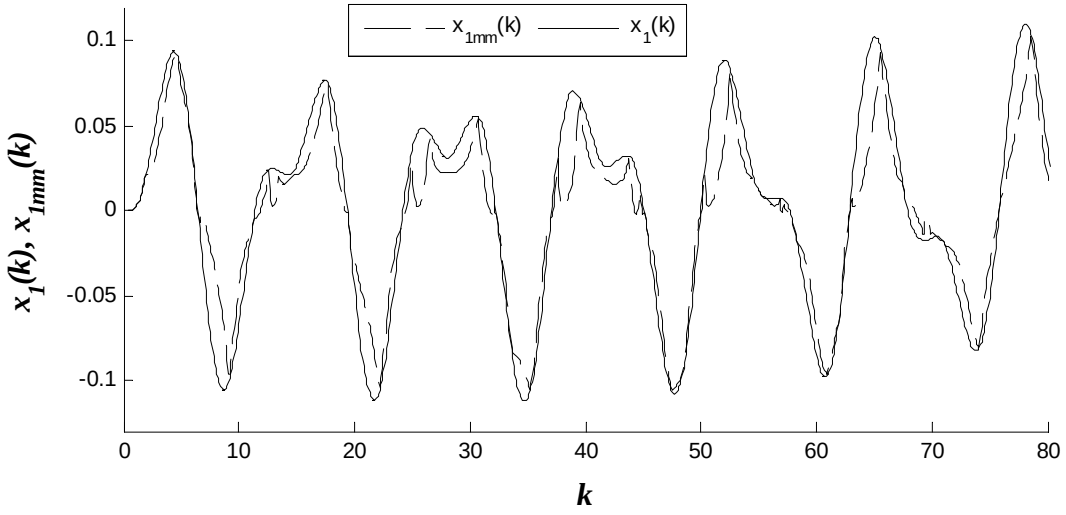


Figure II.23. Exemple 3 : Sorties réelle et multimodèle (séquence d'entrée u_1)

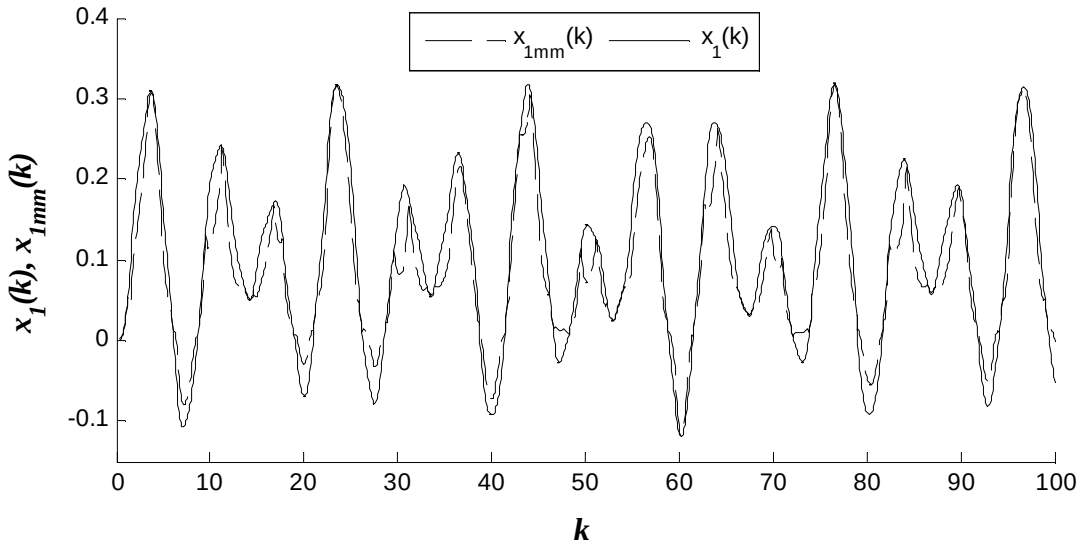


Figure II.24. Exemple 3 : Sorties réelle et multimodèle (séquence d'entrée u_2)

D'après les résultats obtenus, on constate que l'implémentation de l'approche multimodèle proposée offre une modélisation acceptable du système étudié vu sa complexité. En effet, en considérant la première séquence d'entrée u_1, la sortie multimodèle suit la sortie réelle et l'erreur entre les deux signaux diminue au cours du temps. D'autre part, même avec une sortie du système plus complexe (obtenue en appliquant la séquence d'entrée u_2) avec d'importantes oscillations et sur un intervalle de temps plus étendu, les résultats donnés par la l'approche de modélisation proposée (figure II.24) sont aussi satisfaisants.

II.5.4. Exemple 4 : Modélisation d'un réacteur biologique

Afin de mettre en évidence l'intérêt et les apports de notre approche de modélisation, en se basant sur un même système qui est le bio-réacteur, nous avons procédé à une comparaison de nos résultats avec ceux donnés par les travaux de [Cho et al, 07] dans lesquels on propose une approche multimodèle pour les systèmes non linéaires de type « boîte noire » basée sur plusieurs modèles linéaires. En étant un bon exemple académique de système non linéaire, le réacteur biologique a été traité dans certains travaux pour des fins d'illustration de différentes approches de modélisation et de commande de systèmes non linéaires [Bastin et al, 90] [Cho et al, 07] [Gauthier et al, 92].

Il s'agit en effet d'un bio-réacteur dans lequel les micro-organismes se développent en consommant un substrat. Le réacteur, supposé infiniment mélangé et de volume V constant, est alimenté en substrat (concentration constante) avec un débit volumique F. Ces deux

composants, sont supposés être présents avec de faibles concentrations. En désignant par x_1 et x_2 les concentrations des micro-organismes et du substrat, respectivement, et en supposant que le taux spécifique de croissance est donné par le « modèle de Contois » [Contois, 59], on obtient le modèle discret suivant :

$$\begin{cases} x_1(k+1) = x_1(k) + 0.5\dfrac{x_1(k)\,x_2(k)}{x_1(k)+x_2(k)} - 0.5u(k)\,x_1(k) \\ x_2(k+1) = x_2(k) - 0.5\dfrac{x_1(k)\,x_2(k)}{x_1(k)+x_2(k)} - 0.5u(k)\,x_2(k) + 0.05u(k) \\ y(k) = x_1(k) \end{cases} ; \qquad \text{(II.51)}$$

avec $u = \dfrac{F}{V}$ l'entrée qui représente le taux de dilution normalisé (compris entre 0 et 1). Comme pour les autres exemples, on suppose qu'on dispose d'aucune information a priori et on procède à la stimulation du système à l'aide d'une entrée d'excitation adéquate afin de générer l'ensemble de mesures entrée/sortie permettant de décrire la dynamique du processus et qui représentent la base de notre étude.

En considérant l'ensemble de données collectées, on applique l'apprentissage compétitif pénalisant le rival qui permet de déterminer le nombre de modèles de la base, qui pour cet exemple, est égal à dix. En se basant sur les résultats de cet apprentissage qui présentent peu de recouvrement entre les classes et en procédant à plusieurs simulations, nous avons sélectionné l'algorithme des K-moyennes pour la classification et la méthode des validités renforcées pour l'estimation de la contribution de chaque modèle.

Ainsi, avec seulement 10 modèles linéaires contre 196 modèles obtenus par l'application de l'approche de modélisation proposée dans [Cho et al, 07], nous avons réussi à concevoir une structure multimodèle pour le système dont les performances sont comparables à celles résultant de la modélisation proposée dans [Cho et al, 07]. Cette différence au niveau de la taille de la base permet de mettre en valeur notre approche qui garantit une représentativité satisfaisante tout en étant facile à implémenter et à mettre en œuvre et en préservant la simplicité de la structure multimodèle avec un nombre de modèles qui reste raisonnable même face à un système assez complexe sachant qu'une base de modèles volumineuse risque de constituer un handicap en terme de calcul des commandes et/ou de l'analyse et l'étude du système.

Afin d'illustrer les résultats de l'implémentation de l'approche proposée pour la modélisation du bio-réacteur, on considère la séquence d'entrée suivante :

$$u_1(k) = 0.5 + 0.1\sin(0.2k)\,. \tag{II.52}$$

La figure II.25 donne les évolutions des sorties réelle y et multimodèle y_{mm}.

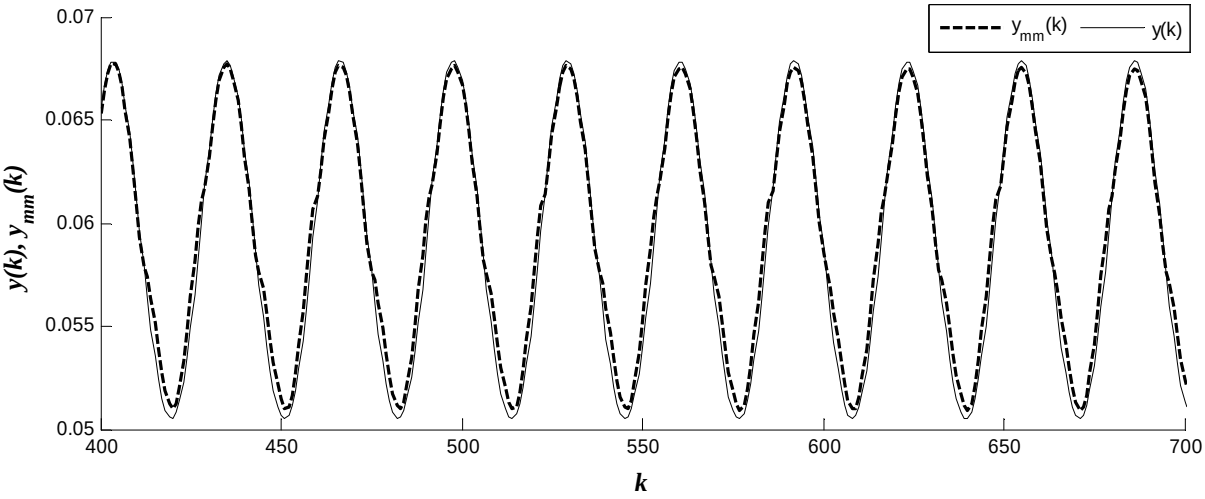


Figure II.25. Exemple 4 : Sorties réelle et multimodèle (séquence d'entrée u_1)

En appliquant la séquence d'entrée u_2 donnée par la figure II.26, on calcule la sortie multimodèle qu'on compare à la sortie réelle (figure II.27).

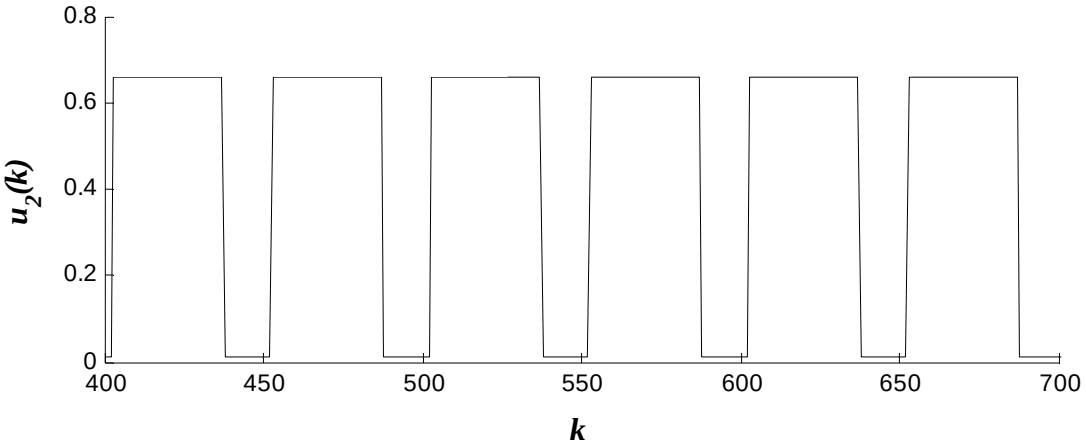


Figure II.26. Exemple 4 : Séquence d'entrée u_2

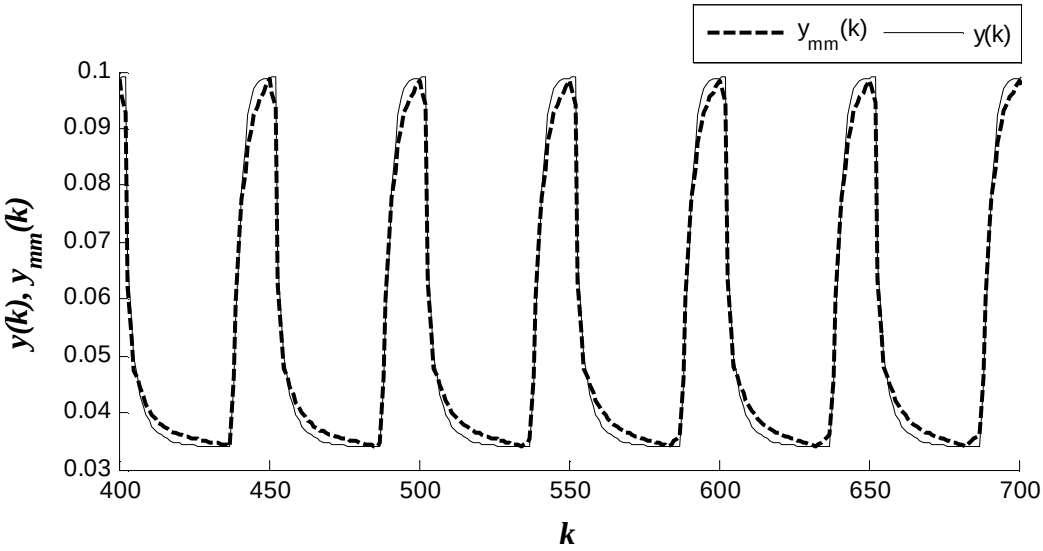


Figure II.27. Exemple 4 : Sorties réelle et multimodèle (séquence d'entrée u_2)

II.6. Interprétations des résultats de la mise en œuvre de l'approche proposée

Les résultats obtenus suite à l'étude de différents types de systèmes dynamiques ont permis de souligner l'intérêt et la précision de l'approche de modélisation proposée et de mettre en évidence l'ensemble de nos contributions au niveau des différentes étapes de la conception des structures multimodèles pour les processus complexes notamment de type « boîte noire » et qui est basée sur un ensemble de mesures entrée/sortie.

- **Contribution au niveau de la détermination du nombre de modèles de la base**

La détermination du nombre de modèles a souvent constitué un problème dans les approches multimodèles proposées pour la représentation de systèmes complexes sur lesquels on dispose de peu ou pas d'information a priori. Dans le but d'apporter une solution à ce problème tout en évitant les inconvénients des méthodes existantes qui peuvent être heuristiques et peu précises (exemple [Ben Aoun Talmoudi, 05]) ou complexes, difficiles à mettre en œuvre et aboutissant parfois à un grand nombre de modèles (exemple [Cho et al, 07]), nous avons proposé une nouvelle méthode basée sur l'apprentissage compétitif pénalisant le rival. A part la simplicité et la facilité d'implémentation de cette technique, nous avons constaté à travers les exemples de simulation que le nombre de modèles sélectionné reste raisonnable même lorsqu'il s'agit de systèmes assez complexes (comme l'exemple 4

traité dans la section II.5.4), permettant ainsi de garantir la facilité et la flexibilité de l'exploitation de la structure multimodèle résultante pour différents objectifs tel que la conception d'une commande multimodèle par exemple.

- **Contribution au niveau de la génération des classes de fonctionnement du processus**

Ayant pour but l'étude de faisabilité de la détermination des domaines de fonctionnement d'un système de type « boîte noire » à partir de la répartition de ses données entrée/sortie, nous avons sélectionné trois méthodes de classification à apprentissage non supervisé qu'on a jugées simples et faciles à implémenter. Notre contribution consiste à distinguer deux classes de systèmes et à adapter en conséquence le choix de la méthode la plus appropriée, ce qui n'a pas été déjà traité dans des travaux antérieurs qui proposent généralement une seule méthode de classification (carte de Kohonen dans la plupart des cas) [Ben Aoun Talmoudi, 05] [Cho et al, 06] [Kukolj et al, 04]. A cet effet, nous avons émis une hypothèse affirmant que si les résultats de l'apprentissage compétitif pénalisant le rival présentent un recouvrement important entre les groupes de données, l'algorithme des K-moyennes floues est privilégiée, sinon on utilise soit la carte de Kohonen, soit l'algorithme des K-moyennes qui est favorisée dans le cas d'un grand nombre de classes et/ou de données système à classifier. En considérant différents systèmes dynamiques, les comparaisons des résultats donnés par les trois méthodes ont permis de confirmer l'hypothèse présentée.

- **Contribution au niveau du mécanisme de décision**

Ayant opté pour l'approche des résidus, nous avons cherché à étudier différentes expressions de calcul des validités et on a fini par sélectionner deux méthodes en favorisant toujours la simplicité pour une mise en œuvre facile et rapide vu qu'il s'agit d'un traitement en ligne. En se basant sur le même raisonnement adopté pour la sélection de l'algorithme de classification le mieux adapté et par simple observations des expressions des validités sélectionnées, nous avons présenté une deuxième hypothèse portant sur le choix de la méthode de calcul des validités qui donne les meilleurs résultats. Les résultats donnés par les différentes simulations ont permis de confirmer cette hypothèse.

Contrairement à la plupart des études proposées dans le domaine qui définissent une seule procédure pour l'estimation des contributions des différents modèles, notre contribution

réside dans l'adaptation du mécanisme de décision suivant que les classes de fonctionnement du processus considéré présentent des recouvrements importants ou non.

Pour conclure, les résultats de l'application de l'approche présentée dans ce chapitre pour la modélisation de différents processus complexes démontrent les performances et l'efficacité des structures multimodèles résultantes qui permettent une représentation assez satisfaisante.

L'organigramme donné par la figure suivante permet de résumer les différentes étapes d'implémentation de l'approche proposée pour la représentation multimodèle.

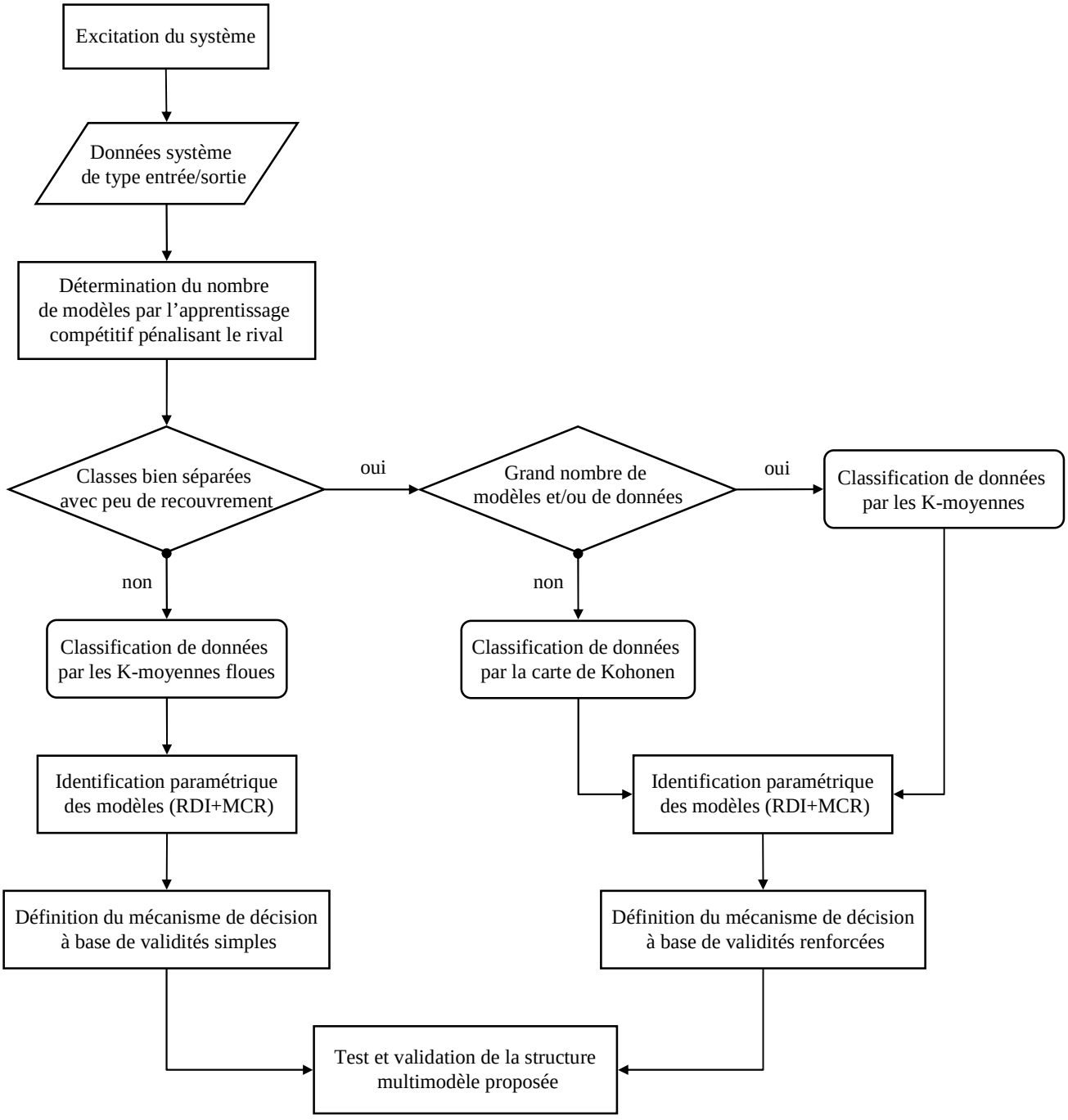


Figure II.28. Organigramme résumant les étapes d'implémentation de l'approche proposée pour la représentation multimodèle

II.7. Conclusion

Dans ce chapitre, une nouvelle approche multimodèle basée sur des algorithmes de classification neuro-floue, a été proposée. Cette approche peut être appliquée pour la représentation de systèmes complexes, fortement non linéaires et/ou incertains. Elle permet la construction de la base en utilisant différentes techniques de classification et deux méthodes d'identification structurelle et paramétrique. Le problème de la détermination du nombre de modèles a été résolu par l'utilisation d'un réseau de neurones et un apprentissage compétitif pénalisant le rival. Une fois ce nombre calculé, les mesures collectées sur le système sont réparties en utilisant un algorithme de classification bien adapté (carte de Kohonen, K-moyennes ou K-moyennes floues) afin de générer les différents domaines de fonctionnement. Les résultats de la classification sont ensuite exploités pour l'identification des modèles de la base. Une étape de validation permet de démontrer la capacité de la structure multimodèle proposée d'approximer la réponse du système dans différents points de fonctionnement. L'approche envisagée a été implémentée et testée pour différents systèmes complexes. Les résultats de simulation se révèlent intéressants et témoignent de l'efficacité et la précision de la stratégie de modélisation ainsi proposée.

Sachant que l'objectif principal de la modélisation des systèmes est de pouvoir les commander, l'adoption d'une approche multimodèle entraîne naturellement la conception d'une commande multimodèle. Cette dernière fait l'objet du troisième chapitre.

Chapitre III

Commande multimodèle de processus complexes

III.1. Introduction

III.2. Principe de la commande multimodèle

III.3. Exemples de systèmes de commande

III.4. Stratégies de la commande multimodèle

III.5. Description de l'approche envisagée pour la commande multimodèle de processus complexes

III.6. Mise en œuvre sur des exemples de systèmes complexes

III.7. Conclusion

III.1. Introduction

Dans le chapitre précédent, nous avons présenté une nouvelle approche pour la représentation multimodèle de différentes classes de processus complexes. Cette approche consiste à définir un ensemble de modèles simples constituant la base de modèles en se basant essentiellement sur des algorithmes de classification et des méthodes d'identification structurelle et paramétrique. Par ailleurs, un mécanisme de décision en ligne, ayant pour fonction d'estimer, à chaque instant, la qualité de chaque modèle en fonction de sa pertinence (indice de validité), a été mis en place. A cet effet, deux méthodes de calcul des validités des modèles ont été présentées.

En automatique, la modélisation d'un système est souvent considérée comme une étape préliminaire pour la conception d'une commande. Lorsqu'il s'agit de processus complexes et fortement non linéaires, l'élaboration d'une structure multimodèle s'avère une méthode assez puissante et efficace permettant de pallier aux difficultés rencontrées lors de la conception d'une commande dans la mesure où le modèle global du système, qui peut être très complexe, et parfois même absent, est remplacé par un ensemble de modèles simples décrivant chacun le processus dans un domaine de fonctionnement particulier. Dans ce contexte, on parle d'une approche multicommande qui consiste à définir une stratégie de commande multimodèle en se basant sur la bibliothèque de modèles. Cette dernière approche a fait l'objet de plusieurs travaux de recherche récents ([Al-Akhras et al, 96] [Elfelly et al, 10b] [Kardous Khaldi, 04] [Mezghani, 00] [Raissi et al, 01] [Sadati et al, 04], etc) et a été adoptée pour des applications dans des domaines divers comme par exemple la chimie et la biologie ([Böling et al, 07] [Liu et al, 08] [Özkan et al, 03]) ou bien aussi l'ingénierie et l'industrie ([Baruch et al, 04] [Cho et al, 06] [Cutululis et al, 06]).

Dans ce chapitre, nous commençons tout d'abord par présenter le principe et les différentes stratégies de commande multimodèle. Ensuite, nous décrivons la méthode envisagée pour la conception d'une commande globale pour les processus complexes, qui prolonge les travaux de chapitre II en mettant en évidence le lien entre la structure multimodèle élaborée par utilisation d'algorithmes de classification, les méthodes/paramètres de synthèse de la commande multimodèle et le niveau des performances obtenues. L'étude de deux exemples de simulation permet d'illustrer l'efficacité et la précision de l'approche ainsi proposée.

III.2. Principe de la commande multimodèle

L'objectif principal est de déterminer une commande globale pour le système considéré en se servant de la base de modèles déterminée par l'approche de la représentation multimodèle. Cette commande est obtenue à partir des commandes partielles issues des différents modèles et est appelée « commande multimodèle ».

La notion de bibliothèque ou base de modèles ayant été introduite, il est naturel de définir une base de lois de commandes ou contrôleurs partiels. En effet, à chaque modèle de la base est associé un contrôleur qui prend en compte la consigne et/ou l'objectif à atteindre. Comme pour la représentation multimodèle, un mécanisme de décision est mis en place afin d'estimer à chaque instant la contribution de chaque modèle au calcul de la commande globale. Cette contribution se présente sous forme d'un indice de validité que l'on attribue à chaque contrôleur.

Pour la conception de la bibliothèque des contrôleurs associés chacun à un modèle de la base, on peut opter pour une structure commune ou pour des structures différentes selon les modèles. Ensuite, la commande globale est déduite à partir des différentes commandes partielles et des indices de validité qui y sont associés. A ce sujet, plusieurs stratégies de commande multimodèle ont été évoquées dans la littérature qui dépendent de plusieurs paramètres (structure du système multimodèle, performances souhaitées, nature et caractéristiques du processus étudié, etc). On revient sur les détails dans la section III.4 en présentant deux grandes classes de stratégies de la commande multimodèle : stratégie par commutation et stratégie par fusion.

Ainsi, la commande globale appliquée au système est considérée comme une commande ajustable (ou adaptative), dans le sens où les paramètres du contrôleur changent à chaque instant et dépendent surtout du mécanisme de décision. La figure III.1 présente le schéma bloc illustrant le principe de la commande multimodèle, avec :

- K : le nombre de modèles de la base ;
- y_c : la consigne à suivre ;
- u : la commande globale appliquée au système obtenue à partir des commandes partielles et des indices de validité générés par le bloc de décision selon une certaine stratégie ;

- y : la sortie du système par application de la commande u ;
- $y_1,\ldots,y_K$: les sorties des différents modèles de la base, obtenues par application des commandes partielles ;
- $v_1,\ldots,v_K$: les indices de validité calculés par l'application de l'une des approches présentées dans le premier chapitre (section I.6.2).

Les validités v_i sont calculées de la même façon que dans la représentation multimodèle à ceci près qu'elles vont servir au calcul du contrôleur global et se trouvent donc associées aux commandes partielles eu lieu et place des sorties des différents modèles pour le calcul de la sortie multimodèle.

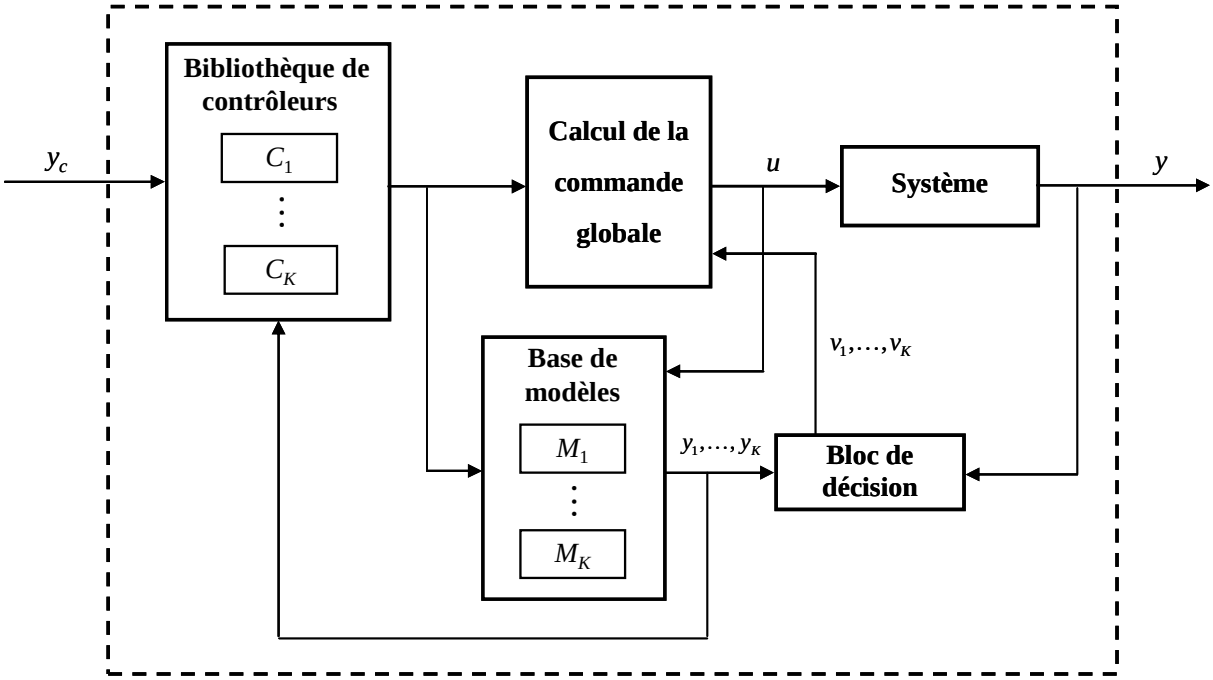


Figure III.1. Principe de la commande multimodèle

Dans le paragraphe suivant, nous allons présenter quelques types de commande dont on pourrait se servir pour la conception des contrôleurs partiels (correspondants aux différents modèles de la base).

III.3. Exemples de systèmes de commande

Pour démarrer la conception d'une commande multimodèle, on associe une loi de commande ou un contrôleur à chaque modèle de la base. Ces contrôleurs peuvent avoir la même structure ou des structures différenciées. Le choix de la ou des lois de commande à faire correspondre aux modèles a une grande influence sur les résultats, d'où l'importance cruciale que revêt cette étape.

Il existe plusieurs types de commande qui diffèrent par leurs principes et leur mise en œuvre. Pour la conception d'une commande, plusieurs paramètres doivent être pris en compte tels que : la précision, la stabilité, le temps de réponse, la fiabilité et la robustesse [Borne et al, 93]. La notion de robustesse, notamment, est très importante. En effet, une commande robuste est une commande « tolérante » c'est-à-dire qui assure un bon fonctionnement du processus même en présence d'erreurs de modélisation ou de perturbations, les critères de robustesse étant définis par le concepteur. Les commandes classiques peuvent être groupées en deux grandes classes : les commandes optimales et les commandes non optimales.

Une commande optimale est une commande qui permet de vérifier des conditions initiales et finales données, de satisfaire diverses contraintes imposées et surtout d'optimiser un critère choisi (par exemple : temps, énergie, consommation) [Borne et al, 90]. Parmi les commandes optimales, on peut citer la commande Linéaire Quadratique LQ, la commande Linéaire Quadratique Gaussienne LQG ou bien aussi la commande bang-bang.

Pour la deuxième classe de commande, il existe plusieurs techniques qui peuvent se présenter en boucle ouverte ou en boucle fermée. Dans la plupart des cas, on s'intéresse plutôt à la régulation en boucle fermée. En effet, un système est dit en boucle fermée lorsque la sortie du procédé est prise en compte pour calculer l'entrée. Généralement le contrôleur effectue une action fonction de l'erreur entre la mesure et la consigne désirée. Le schéma classique d'un système linéaire pourvu d'un régulateur linéaire en boucle fermée est donné par la figure III.2.

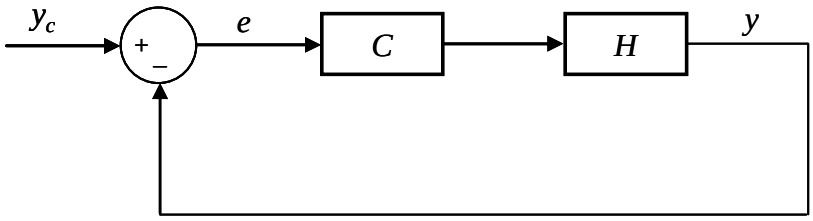


Figure III.2. Principe de la régulation en boucle fermée

- C : fonction de transfert du correcteur ;
- H : fonction de transfert du processus ;
- y_c : consigne à suivre ;
- y : sortie du processus ;
- e : erreur entre la sortie du processus y et la consigne désirée y_c.

La boucle ouverte du système est composée du procédé et du correcteur. La fonction de transfert de ce système en boucle ouverte est donc:

$$H_{BO} = H \cdot C \tag{III.1}$$

La fonction de transfert en boucle fermée s'écrit alors :

$$H_{BF} = \frac{H \cdot C}{1 + H \cdot C} \tag{III.2}$$

La formule précédemment énoncée n'est valable que si le schéma-bloc est à retour unitaire. La fonction de transfert change alors en ajoutant un bloc dans la boucle de retour.

La disposition et le type des blocs du contrôleur dépendent de la loi de commande adoptée. Dans les paragraphes suivants, nous allons présenter brièvement quelques exemples de ces régulateurs.

III.3.1. Régulateur PID

Un régulateur PID (Proportionnel, Intégral, Dérivée) est un outil de contrôle agissant en boucle fermée. Le principe consiste à calculer une action proportionnelle, intégrale et dérivée en fonction de l'erreur consigne/mesure. Il s'agit d'un régulateur très utilisé pour la commande de processus industriels.

III.3.2. Commande à modèle interne

Le principe de la commande à modèle interne consiste à agir séparément, suivant deux dynamiques indépendantes, d'une part sur la poursuite de la consigne et d'autre part sur le rejet des perturbations. Pour la mise en œuvre de cette commande, il faut que le système soit stable.

III.3.3. Commande par retour d'état

Ce type de commande consiste à imposer un comportement dynamique souhaité au moyen de retours d'état adéquats. Pour sa mise en œuvre, le modèle du processus doit être exprimé sous la forme d'une équation d'état.

Le principe de la conception d'une commande par retour d'état consiste généralement à déterminer deux matrices (ou gains matriciels de bouclage) de façon à satisfaire certaines spécifications qui peuvent reposer sur un placement de valeurs propres en boucle fermée ou plus généralement sur un placement de structure. Le placement des valeurs propres permet ainsi de fixer les pôles ou la dynamique du système bouclé [Borne et al, 93].

III.3.4. Commande de type RST

Cette stratégie de commande est connue pour ses capacités d'exploitation des possibilités de la commande numérique avec une approche permettant une mise en œuvre aisée sur calculateur numérique [Borne et al, 93].

Le régulateur RST, dont le nom reflète les trois polynômes qu'il fait intervenir, est basé sur un algorithme de commande largement diffusé et dont la synthèse s'effectue par placement de pôles. En effet, les trois polynômes R (Retour), S (Simplification) et T (Tracking) doivent être choisis et dimensionnés afin que la fonction de transfert du système en boucle fermée réponde à un modèle de référence, ou un modèle à poursuivre, préconisé par l'automaticien en imposant les pôles du système. La commande RST est donc un cas particulier de la commande par placement de pôles.

Dans la procédure de la commande multimodèle, il est possible de faire appel à l'une ou certaines des stratégies précédemment exposées pour la conception de la bibliothèque de contrôleurs. Dans notre étude, nous nous intéressons particulièrement à la régulation RST

qu'on adoptera pour l'ensemble des modèles de la base. Le principe de cette commande sera présenté en détails dans le paragraphe III.5.1.

III.4. Stratégies de la commande multimodèle

Les diverses études menées dans le domaine de l'approche multicommande ont abouti à l'apparition de plusieurs stratégies de commande multimodèle. Ces stratégies diffèrent en fonction de plusieurs paramètres tels que la nature et les caractéristiques du processus étudié, la structure du système multimodèle (à modèle global implicite ou explicite faisant intervenir des modèles locaux ou génériques) et les performances souhaitées pour la commande. De ce fait, il est difficile de déduire une classification unique et exhaustive de ces stratégies.

En considérant la façon selon laquelle la commande globale est obtenue, les stratégies de commande multimodèle peuvent être regroupées en deux grandes classes : stratégies par commutation ou stratégies par fusion.

III.4.1. Stratégie de commande multimodèle par commutation

Le principe de la commande multimodèle par commutation consiste à concevoir un contrôleur pour chaque modèle M_i de la base, puis à déduire la commande globale par commutation entre les différentes commandes obtenues u_i ($i = 1, \ldots, K$). Ainsi, à chaque instant, le processus se trouve commandé par un seul contrôleur relatif à l'un des modèles de la base. En d'autres termes, à chaque instant, un seul modèle M_j ($j \in \{1, \ldots, K\}$) est capable de représenter le processus dans le point de fonctionnement considéré, le système est alors équivalent au modèle M_j uniquement et sa sortie y est égale à la sortie y_j du modèle en question [Kardous Khaldi, 04].

$$\forall k,\ \exists\ j \in \{1, \ldots, K\} \text{ tel que } u(k) = u_j(k) \text{ et } y(k) = y_j(k). \qquad \text{(III.3)}$$

Cette stratégie de commande, illustrée par la figure III.3, présente l'avantage de la facilité d'implémentation et de manipulation dans la mesure où elle fait intervenir un mécanisme de décision assez simplifié. Elle s'adapte bien au cas de la modélisation idéale où la séquence de commutation peut être connue a priori. Toutefois, si la base contient un grand nombre de modèles, il est souvent difficile de déterminer une règle de commutation. D'autre

part, dans le cas d'une modélisation générique, à un instant donné, il est rare qu'un seul modèle puisse représenter le système global, et donc sa commande ne peut pas suffire pour contrôler le processus.

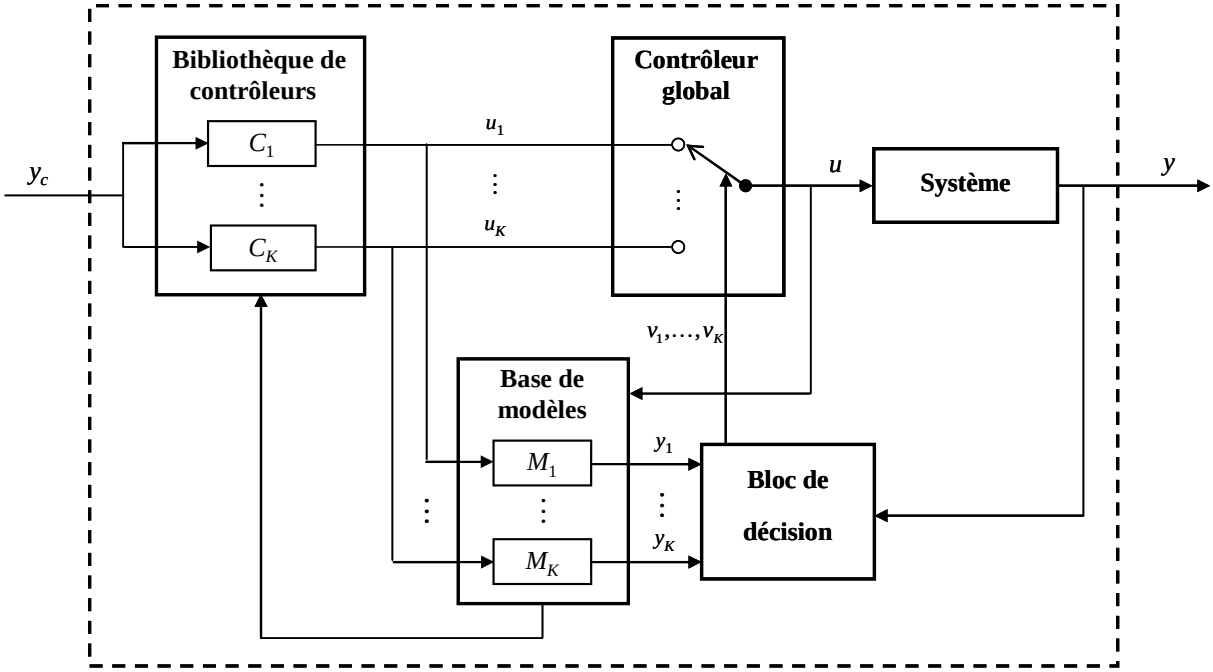


Figure III.3. Principe de la stratégie de commande par commutation

La stratégie de commande par commutation peut être considérée comme un cas particulier de la stratégie de commande par fusion qui sera abordée dans le paragraphe suivant.

III.4.2. Stratégie de commande multimodèle par fusion

Dans le premier chapitre (paragraphe I.3.2), on a introduit le principe de la fusion dans la représentation multimodèle qui se base sur le fait que chaque modèle de la base contribue partiellement à chaque instant à la description du processus dans le point de fonctionnement considéré. Et par conséquent, le modèle global du processus, qu'il soit implicite ou explicite, doit être déduit de tous les modèles de la base avec des proportions variables d'un modèle à un autre et d'un instant à un autre.

De la même façon, on définit la stratégie de commande par fusion. En effet, le contrôleur global du système doit être conçu de telle sorte qu'il tienne en compte, à chaque

instant, de toutes les commandes ou de tous les paramètres de commande des modèles de la base. Une opération de fusion entre ces commandes partielles ou leurs paramètres, pondérés par les validités des modèles respectifs, pourrait bien convenir à cette situation.

III.4.2.1. Fusion des commandes partielles

Cette stratégie de fusion consiste à appliquer au système une commande globale calculée directement à travers une fusion des commandes partielles, relatives aux différents modèles de la base, pondérées par leurs coefficients de validité respectifs, comme l'illustre l'expression suivante :

$$u(k) = \sum_{i=1}^{K} v_i(k)\, u_i(k)\,. \tag{III.4}$$

La figure III.4 représente la structure du système de commande dans le cas de la fusion des commandes partielles.

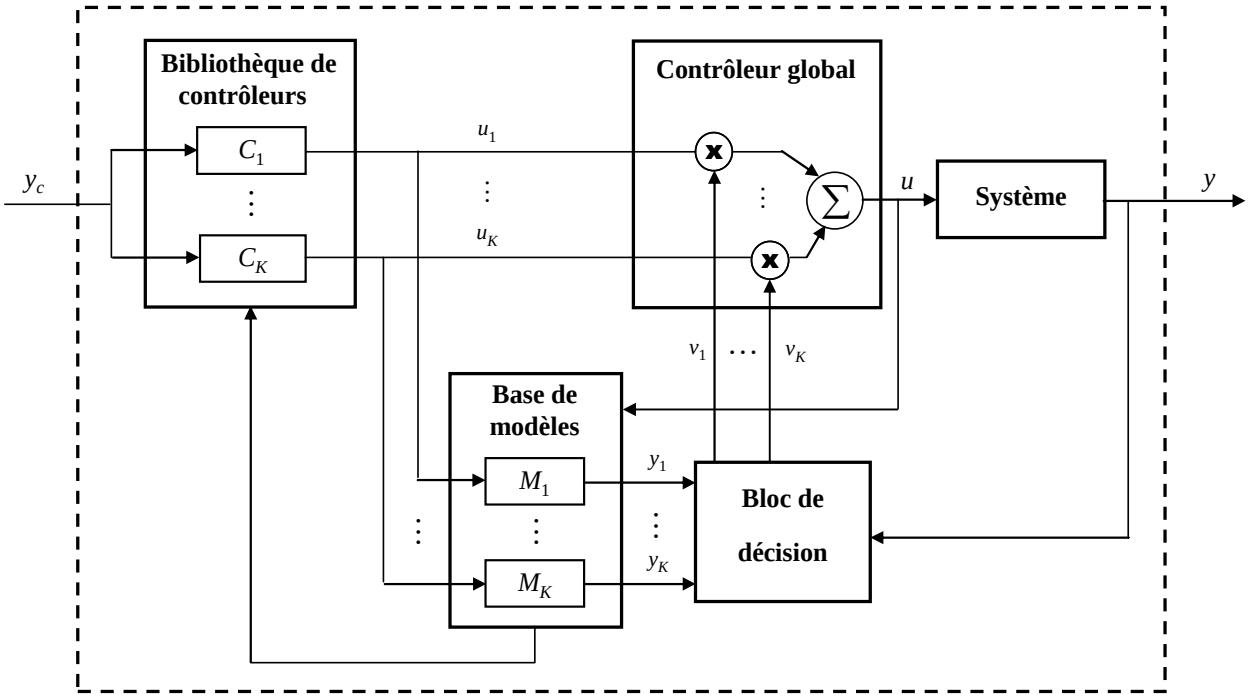


Figure III.4. Principe de la stratégie de commande par fusion des commandes partielles

Ce type de fusion offre la possibilité de considérer diverses lois de commande pour les modèles de la base. Il est même envisageable de concevoir différentes lois de commande pour chacun des modèles [Delmotte, 97].

III.4.2.2. Fusion des paramètres des commandes

S'agissant toujours d'une stratégie par fusion, une autre possibilité consiste à procéder par fusion non plus au niveau des commandes elles mêmes mais plutôt au niveau de leurs paramètres. Ceci laisse entendre bien évidemment que les commandes partielles associées aux différents modèles de la base doivent suivre la même loi sinon la fusion ne sera plus possible.

La figure III.5 illustre le principe de cette stratégie où les paramètres p de la commande globale (suivant la même loi que les commandes partielles) sont calculés, à chaque instant, à travers une fusion entre les paramètres p_i $(i = 1, \ldots, K)$ des contrôleurs C_i associés aux différents modèles M_i de la base, pondérés par les degrés de validité respectifs v_i :

$$p(k) = \sum_{i=1}^{K} v_i(k)\ p_i(k)\,. \tag{III.5}$$

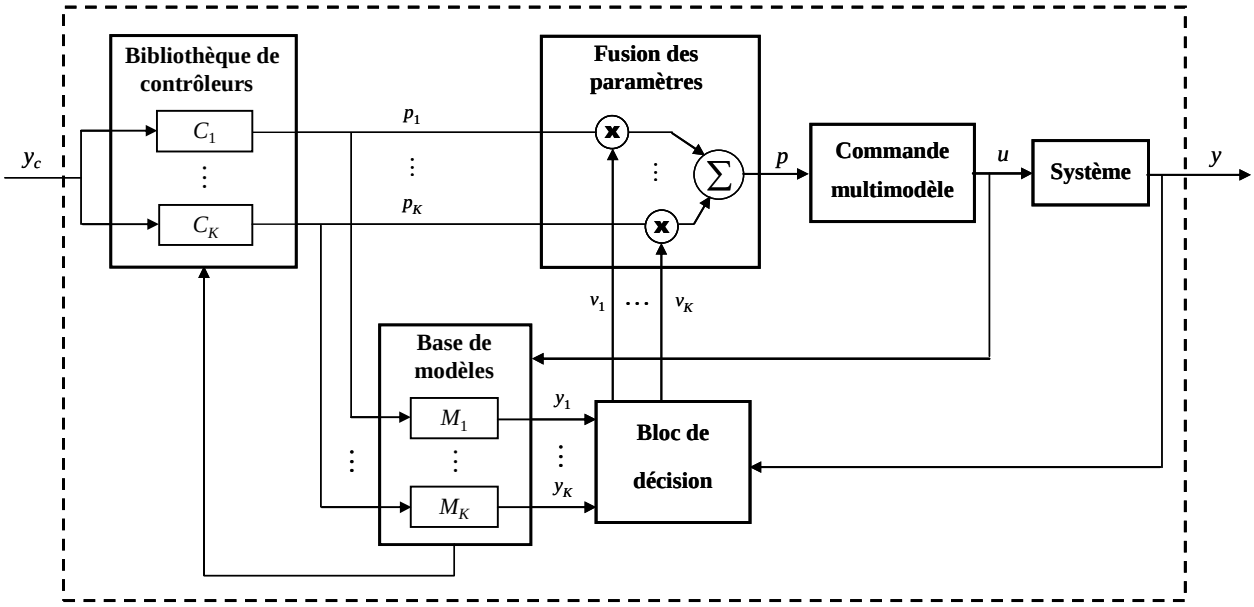


Figure III.5. Principe de la stratégie de commande multimodèle par fusion des paramètres des commandes partielles

La structure ainsi présentée rappelle celle utilisée dans les cas des commandes par séquencement de gain qui a souvent été considéré comme l'approche systématique la plus commune pour la commande des systèmes fortement non linéaires [Packard, 94] [Rugh et al, 00].

III.5. Description de l'approche envisagée pour la commande multimodèle de processus complexes

Basé sur la structure multimodèle, nous rappelons que le principe de la commande multimodèle est de concevoir un contrôleur pour chaque modèle de la base et la commande globale est ensuite déduite à partir des commandes partielles ou élémentaires. Dans ce contexte, les études élaborées diffèrent selon le ou les lois de commande adoptées pour les modèles de la base ainsi que la stratégie de la conception du contrôleur global.

En effet, certains auteurs adoptent l'approche neuronale comme loi de commande [Baruch et al, 08] ; d'autres se penchent plutôt vers les lois de commande classiques comme par exemple les régulateurs PID [Böling et al, 07], la commande prédictive [Özkan et al, 03] ou la commande adaptative [Fu et al, 07] [Cezayirli et al, 08].

Pour la stratégie de déduction de la commande globale du système à partir de la bibliothèque des contrôleurs, on a présenté deux grandes classes de méthodes. En effet, on peut envisager, dans certains cas, une commutation entre les commandes partielles [Fu et al, 07] ou plus généralement une fusion (entre les commandes ou entre leurs paramètres) en utilisant les indices de validité des modèles [Sadati et al, 04] [Lorimier et al, 03].

Dans le deuxième chapitre, nous avons proposé une approche pour la représentation multimodèle pour différentes classes de systèmes complexes. Basée sur des algorithmes de classification, cette approche permet la construction de la base de modèles en exploitant uniquement des données de type entrée/sortie collectées sur le système et décrivant sa dynamique dans l'ensemble de son espace de fonctionnement. Nous rappelons brièvement le principe qui consiste à déterminer tout d'abord le nombre de modèles, à répartir ensuite les données en classes de fonctionnement en utilisant un algorithme de classification adéquat et enfin à identifier les modèles en se servant des données relatives à chaque classe.

Ainsi, la mise en œuvre de l'approche proposée pour la représentation de systèmes complexes résulte en une certaine structure multimodèle. Cette dernière préserve bien les avantages des approches multimodèles qui sont essentiellement la simplicité et la flexibilité dans le sens où elle propose de décrire un système complexe et généralement fortement non linéaire par un nombre raisonnable de modèles linéaires assez simples.

En se basant sur une telle structure, la conception d'une commande pour les systèmes complexes ne présente plus de difficulté méthodologique conséquente comme elle pouvait l'être avec un modèle non linéaire assez complexe obtenu par l'application d'une approche classique de modélisation. Il est tout de tout de même important de noter que la conception d'une commande multimodèle, bien qu'elle apparaît simple, est une tâche assez délicate dans la mesure où elle exige un choix rigoureux des lois de commande et de la stratégie de déduction du contrôleur global afin d'aboutir à des résultats qui répondent bien au niveau de performances souhaité.

En partant de la base de modèles résultat de la mise en œuvre de l'approche décrite au deuxième chapitre, la démarche proposée pour la conception d'une commande multimodèle consiste tout d'abord à concevoir un contrôleur qu'on associe à chaque modèle. La loi de commande de type RST a été retenue du fait de sa mise en œuvre aisée et de l'obtention de performances correctes même en présence de processus avec des paramètres variant au cours du fonctionnement ou de processus d'ordre élevé.

III.5.1. Calcul des commandes partielles par la loi de régulation R-S-T

Bien que relativement facile à régler et très utilisé dans l'asservissement de nombre de processus industriels, le régulateur PID ne permet pas toujours de réaliser de bonnes performances dans le cas de systèmes :

- à retard pur important ;
- dont les caractéristiques dynamiques varient au cours du fonctionnement ;
- d'ordre supérieur à 2 ;
- sans dépassement de consigne autorisé.

Une alternative consiste alors à considérer le régulateur RST plus performant et plus robuste et qui présente un algorithme plus sophistiqué exploitant à fond les ressources numériques dont on dispose et dont la synthèse est purement algébrique.

La commande de type RST est basée sur un algorithme de commande largement diffusé, et dont la structure à trois branches R, S et T permet :

- de gérer indépendamment les dynamiques de poursuite (asservissement) et la dynamique de réjection de perturbation (régulation),
- de spécifier indépendamment les temps de montée et le dépassement sur la consigne,
- de tenir compte du retard pur du processus,
- d'obtenir une régulation robuste vis à vis des variations du processus d'une part et des changements de point de consigne d'autre part,
- et en final, d'obtenir des mesures plus proches de la consigne imposée et, en conséquence, des gains potentiels en précision, qualité, énergie et matières premières.

Ainsi, à chaque modèle de la base, on associe un régulateur RST dont la structure est donnée par la figure III.6 où :

- q^{-1} désigne l'opérateur du retard ;
- y_c est la consigne à suivre ;
- y_i est la sortie du modèle M_i obtenue en appliquant la commande partielle u_i.

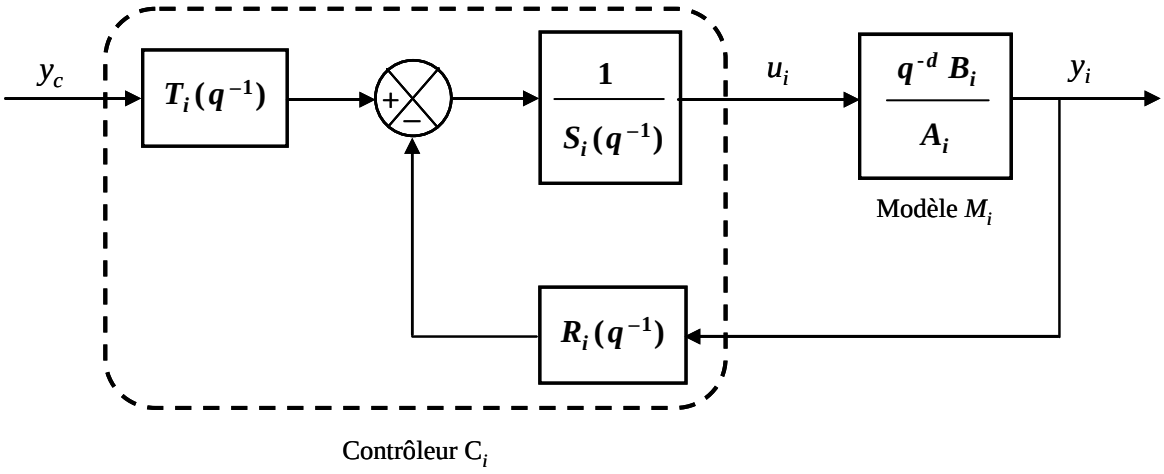


Figure III.6. Structure du régulateur R-S-T associé au modèle M_i

Ainsi, pour chaque modèle M_i ($i = 1, \ldots, K$) de la base, l'algorithme de régulation RST [Borne et al, 93] permet de calculer, à chaque période d'échantillonnage, la valeur de la

commande partielle u_i à partir de la lecture synchronisée de la consigne y_c (objectif à atteindre) et de la mesure y_i (vision du résultat), comme l'illustre l'équation suivante :

$$u_i(k) = \frac{1}{S_i(q^{-1})}\left[T_i(q^{-1})\, y_c(k) - R_i(q^{-1})\, y_i(k)\right] \quad i = 1,\ldots,K \ ; \tag{III.6}$$

avec :

$$R_i(q^{-1}) = r_{i0} + r_{i1}\, q^{-1} + \ldots + r_{in_R}\, q^{-n_R}; \tag{III.7}$$

$$S_i(q^{-1}) = s_{i0} + s_{i1}\, q^{-1} + \ldots + s_{in_S}\, q^{-n_S} \ ; \tag{III.8}$$

$$T_i(q^{-1}) = t_{i0} + t_{i1}\, q^{-1} + \ldots + t_{in_T}\, q^{-n_T} . \tag{III.9}$$

La fonction de transfert du modèle M_i est donnée par :

$$F_i(q^{-1}) = \frac{q^{-d} B_i(q^{-1})}{A_i(q^{-1})} \ ; \tag{III.10}$$

avec :

$$B_i(q^{-1}) = b_{i0} + b_{i1}\, q^{-1} + \ldots + b_{in_{B_i}}\, q^{-n_{B_i}}; \tag{III.11}$$

$$A_i(q^{-1}) = a_{i0} + a_{i1}\, q^{-1} + \ldots + a_{in_{A_i}}\, q^{-n_{A_i}} \ ; \ a_{i0} \neq 0 . \tag{III.12}$$

La fonction de transfert en boucle fermée du modèle M_i en réponse à la commande u_i et en considérant la consigne y_c s'écrit alors :

$$F_i^{BF}(q^{-1}) = \frac{q^{-d}\, T_i(q^{-1})\, B_i(q^{-1})}{P_i(q^{-1})} \ ; \tag{III.13}$$

tel que :

$$P_i(q^{-1}) = A_i(q^{-1})\, S_i(q^{-1}) + q^{-d} B_i(q^{-1})\, R_i(q^{-1}) . \tag{III.14}$$

Afin de calculer les paramètres (*R*, *S* et *T*) de chacun des régulateurs associés aux différents modèles de la base, on commence tout d'abord par définir la dynamique à imposer à travers le choix des pôles en boucle fermée, qui sont dans notre cas, les mêmes pour tous les

modèles. Ceci nous permet d'écrire un dénominateur commun $P(q^{-1})$ pour toutes les fonctions de transfert en boucle fermée correspondant aux différents modèles :

$$P(q^{-1}) = 1 + p_1 q^{-1} + p_2 q^{-2} + \dots \tag{III.15}$$

Ainsi, pour chaque modèle M_i ($i = 1, \dots, K$), le problème de calcul des paramètres de la commande partielle se ramène à la résolution du système suivant :

$$\begin{cases} P_i(q^{-1}) = 1 + p_1 q^{-1} + p_2 q^{-2} + \dots \\ P_i(q^{-1}) = A_i(q^{-1}) S_i(q^{-1}) + q^{-d} B_i(q^{-1}) R_i(q^{-1}) \end{cases} \tag{III.16}$$

Sachant que les polynômes $R_i(q^{-1})$ et $S_i(q^{-1})$ doivent être choisis de degrés compatibles avec le degré recherché pour $P(q^{-1})$, la résolution du système (III.16) connue sous le nom d' « identité de Bezout » ou encore d' « équation Diophantienne » s'effectue en identifiant les termes de même degré en *q*.

III.5.2. Calcul de la commande globale du système

Une fois les paramètres des régulateurs partiels identifiés, nous allons maintenant présenter la méthode à mettre en œuvre pour la déduction de la commande globale à appliquer au système complexe considéré afin de garantir le comportement désiré. Entre les deux stratégies par commutation et par fusion, nous nous sommes intéressés à la deuxième puisque celle-ci s'avère plus convenable à notre modélisation dans la mesure où il s'agit de modèles génériques sur lesquels on ne dispose d'aucune information a priori que ce soit sur leur nombre ou sur leurs validités dans les différents domaines de fonctionnement. Dans ce cas, il est très difficile qu'un seul modèle puisse représenter le système global dans n'importe quel point de fonctionnement, et par conséquent le contrôleur associé à ce modèle ne peut pas suffire pour commander le système.

Ayant opté pour la même loi de commande pour tous les modèles de la base, nous avons le choix entre deux méthodes de fusion. La première est une simple fusion entre les commandes partielles u_i pondérés par les coefficients de validité v_i ($i = 1, \dots, K$). Ainsi, étant donné une consigne à suivre, la commande globale *u* appliquée au système est calculée à chaque instant suivant l'expression (III.4).

La deuxième méthode, quant à elle, est un peu plus compliquée dans le sens où elle exige une étape intermédiaire qui consiste à calculer les paramètres de la commande globale, qui doit être du même type que les commandes partielles, avant de l'appliquer au système. Chaque paramètre du contrôleur global est obtenu en effectuant une fusion entre les paramètres des contrôleurs issus du multimodèle pondérés par les validités des modèles associés. S'agissant d'une loi de commande de type RST, le contrôleur global est alors défini par les polynômes $R_g(q^{-1})$, $S_g(q^{-1})$ et $T_g(q^{-1})$ qui sont calculés, à chaque instant, suivant les formules ci-dessous :

$$R_g(q^{-1}) = \sum_{i=1}^{K} \nu_i \, R_i(q^{-1}) \ ; \tag{III.17}$$

$$S_g(q^{-1}) = \sum_{i=1}^{K} \nu_i \, S_i(q^{-1}) \ ; \tag{III.18}$$

$$T_g(q^{-1}) = \sum_{i=1}^{K} \nu_i \, T_i(q^{-1}) \ . \tag{III.19}$$

En considérant une certaine consigne y_c à suivre, la commande globale est alors calculée comme suit :

$$u(k) = \frac{1}{S_g(q^{-1})} \left[T_g(q^{-1}) \, y_c(k) - R_g(q^{-1}) \, y(k) \right]. \tag{III.20}$$

Comme pour la représentation multimodèle, les validités sont calculées à travers un mécanisme de décision qui permet d'estimer, à chaque instant, la contribution de chaque modèle à la génération de la commande globale en mettant en œuvre l'une des deux méthodes (validités simples ou renforcées), présentées dans le deuxième chapitre (paragraphe II.4.1). Basé sur l'hypothèse émise dans la section II.4.2 du chapitre II, le choix de la méthode adéquate de calcul des indices de validité associés aux commandes partielles ou à leurs paramètres est effectué lors de l'élaboration de la structure multimodèle suivant que les domaines de fonctionnement du processus étudié, résultats de la classification des données système, se chevauchent ou non.

Pour résumer, la procédure de calcul de la commande multimodèle que nous proposons consiste à synthétiser un contrôleur de type RST pour chacun des modèles de la base en imposant une dynamique adéquate. Ensuite, étant donné une consigne à suivre, une stratégie

de fusion est mise en place pour le calcul de la commande globale du système. A cet effet, deux méthodes peuvent être utilisées, soit une simple fusion entre les commandes partielles, soit une fusion entre les paramètres des différents régulateurs RST issus du multimodèle en faisant appel à un mécanisme de décision en ligne permettant d'estimer le poids de contribution de chaque contrôleur.

L'approche proposée pour la commande multimodèle de a été implémentée et testée pour quelques systèmes dynamiques complexes, parmi lesquels deux seront présentés dans les sections suivantes. Les résultats obtenus vont permettre de faire quelques interprétations concernant la stratégie de la déduction du contrôleur global.

III.6. Mise en œuvre sur des exemples de systèmes complexes

Afin mettre en évidence l'intérêt et la performance de l'approche proposée pour la commande multimodèle, nous allons étudier, dans ce qui suit, sa faisabilité sur deux exemples de simulation.

III.6.1. Exemple 1 : Commande d'un système discret du second ordre avec des paramètres dépendant du temps

On considère le premier exemple de simulation présenté dans le deuxième chapitre dont on rappelle les équations décrivant la dynamique :

$$y(k) = -a_1(k)\, y(k-1) - a_2(k)\, y(k-2) + b_1(k)\, u(k-1) + b_2(k)\, u(k-2)$$

avec :

$$a_1(k) = 0.04 \sin(0.035k) - 0.8 \ ;$$

$$a_2(k) = 0.005 \sin(0.03k) + 0.1 \ ;$$

$$b_1(k) = 0.02 \sin(0.03k) + 0.5 \ ;$$

$$b_2(k) = 0.01 \sin(0.035k) + 0.2 .$$

L'implémentation de l'approche de modélisation décrite dans le deuxième chapitre a permis de construire une base de trois modèles linéaires du second ordre dont les fonctions de

transfert sont décrites par les expressions (II.35) à (II.37). Une fois cette base testée et validée, nous allons procéder à la conception d'une commande multimodèle permettant d'assurer le comportement désiré.

- **Conception des contrôleurs partiels**

La loi de commande de type RST étant adoptée, la première étape de l'implémentation de l'approche envisagée consiste à associer un contrôleur à chaque modèle de la base. Pour ce faire, on commence tout d'abord par imposer une dynamique adéquate à l'ensemble de ces modèles pour déduire les différents paramètres des commandes qui y sont associées : R_i, S_i et T_i ($i = 1,2,3$). Le tableau 1 donne les valeurs de ces paramètres pour chacun des contrôleurs partiels.

Tableau III.1. Exemple 1 : Paramètres des commandes partielles

Modèle	**Paramètres du contrôleur**
M_1	$R_1(q^{-1}) = 2.225 - 1.609\,q^{-1} + 0.024\,q^{-2}$ $S_1(q^{-1}) = 1 - 0.968\,q^{-1} - 0.032\,q^{-2}$ $T_1(q^{-1}) = 2.56 - 2.56\,q^{-1} + 0.64\,q^{-2}$
M_2	$R_2(q^{-1}) = 3.112 - 2.021\,q^{-1} - 0.138\,q^{-2}$ $S_2(q^{-1}) = 1 - 1.152\,q^{-1} + 0.152\,q^{-2}$ $T_2(q^{-1}) = 3.811 - 3.811\,q^{-1} + 0.952\,q^{-2}$
M_3	$R_3(q^{-1}) = 1.588 - 1.096\,q^{-1} - 0.016\,q^{-2}$ $S_3(q^{-1}) = 1 - 0.892\,q^{-1} - 0.108\,q^{-2}$ $T_3(q^{-1}) = 2.034 - 2.034\,q^{-1} + 0.508\,q^{-2}$

- **Calcul de la commande globale**

Pour le calcul de la commande globale, on procèdera tout d'abord à une fusion des commandes partielles et ensuite à une fusion au niveau des paramètres de commande en mettant en œuvre un mécanisme de décision permettant d'estimer la contribution de chacun des contrôleurs partiels à chaque instant. Afin de vérifier que l'hypothèse émise dans le deuxième chapitre (section II.4.2) et portant sur le choix de la méthode appropriée de calcul

des validités, est aussi bien valable pour la représentation multimodèle que pour la commande multimodèle, nous allons procéder, pour chaque méthode de fusion, à une comparaison entre les résultats donnés par chacune des deux méthodes en faisant intervenir des validités simples ou des validités renforcées. D'autre part, nous émettrons quelques interprétations sur l'efficacité de chacune des méthodes de fusion.

Dans un premier temps, nous allons considérer comme consigne la séquence suivante :

$$y_c(k) = 1 + \sin(0.05k). \tag{III.21}$$

- **Fusion des commandes partielles**

En considérant la consigne donnée par l'expression (III.21), la loi de régulation RST permet de calculer les différentes commandes partielles u_i $(i = 1,2,3)$ suivant la formule (III.6). La figure III.7 donne les évolutions de ces trois commandes.

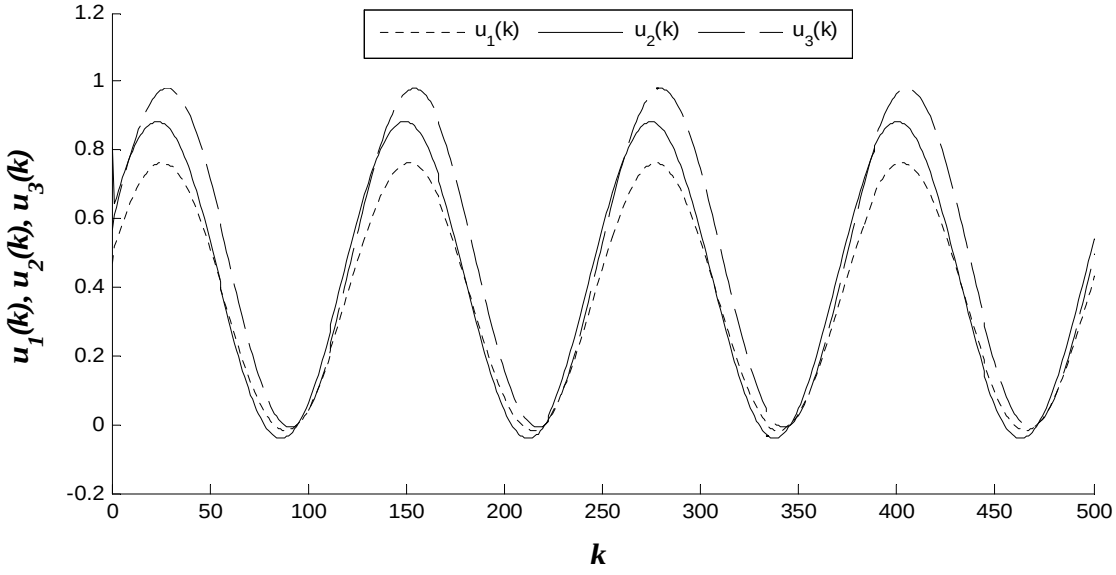


Figure III.7. Exemple 1 : Evolutions des commandes partielles (consigne 1)

Sur les figures III.8 et III.9, on présente les évolutions de la consigne y_c et la sortie du système pour chacune des méthodes de calcul des validités où :

- y_{fc}^{simp} est la réponse du système à la commande u_{fc}^{simp} obtenue par fusion des commandes partielles pondérées par les validités simples :

$$u_{fc}^{simp}(k) = \sum_{i=1}^{3} v_i^{simp}(k)\, u_i(k) \ ; \tag{III.22}$$

- y_{fc}^{renf} est la réponse du système à la commande u_{fc}^{renf} calculée par fusion des commandes partielles pondérées par les validités renforcées :

$$u_{fc}^{renf}(k) = \sum_{i=1}^{3} v_i^{renf}(k)\, u_i(k)\,. \qquad \text{(III.23)}$$

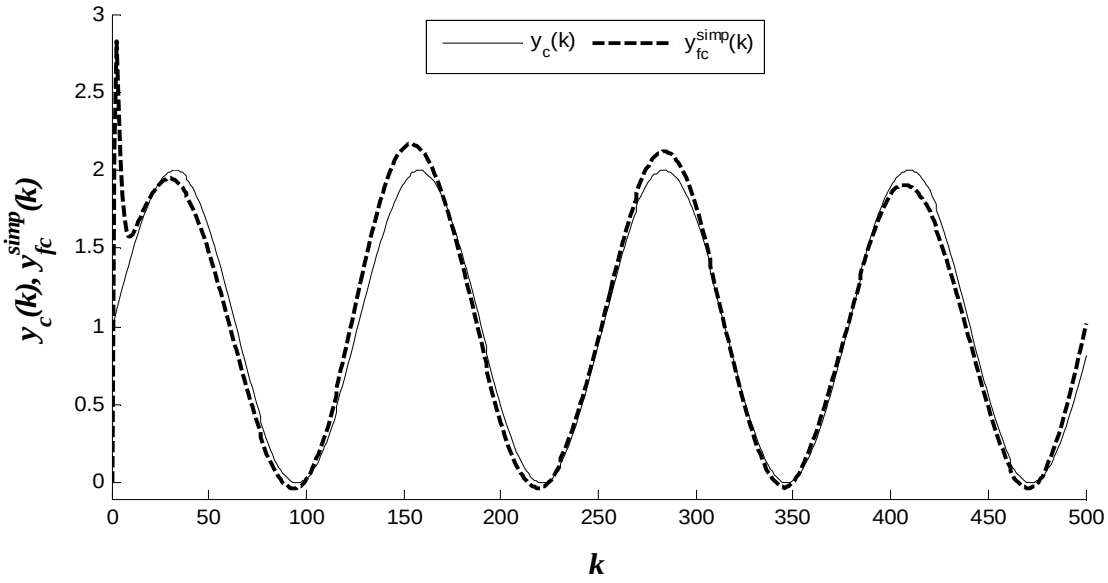


Figure III.8. Exemple 1 : Evolutions de la consigne 1 et de la réponse du système (fusion des commandes / validités simples)

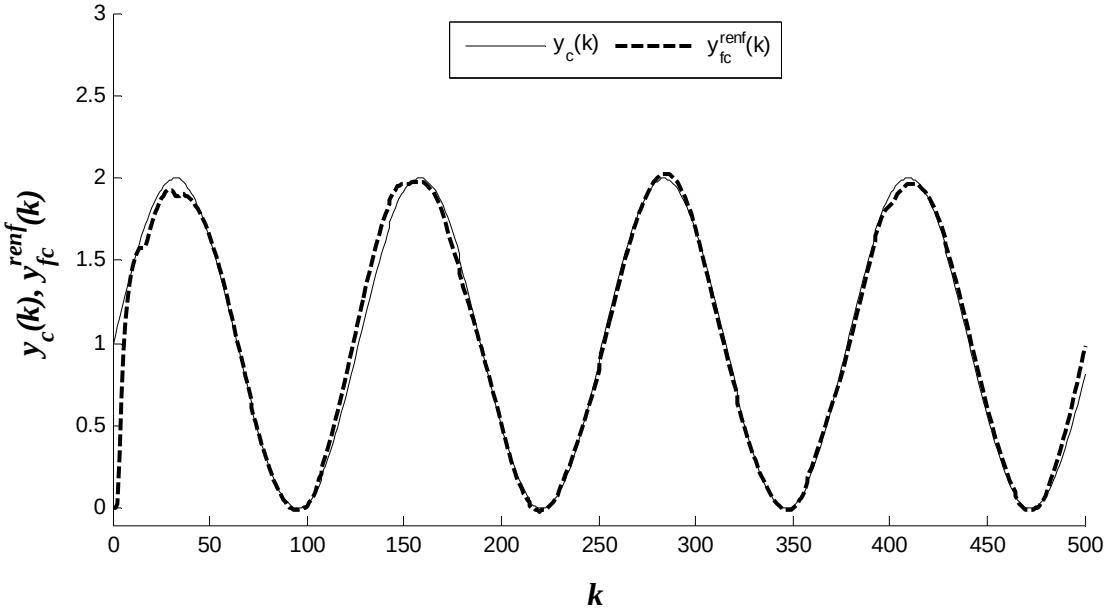


Figure III.9. Exemple 1 : Evolutions de la consigne 1 et de la réponse du système (fusion des commandes / validités renforcées)

En comparant les résultats donnés par les deux figures précédentes, on constate que l'utilisation des validités renforcées est plus convenable pour le calcul de la commande globale du système considéré.

- **Fusion des paramètres des contrôleurs partiels**

Dans ce cas, le contrôleur global est défini par les polynômes R_g, S_g et T_g calculés, à chaque instant, par fusion des polynômes issus des contrôleurs du multimodèle R_i, S_i et T_i ($i = 1,2,3$) pondérés par les coefficients de validité.

S'agissant toujours de la même consigne (expression (III.21)), la réponse du système à la commande obtenue par fusion des paramètres des contrôleurs, est générée en considérant les deux méthodes de calcul des validités. Les résultats sont donnés par les figures III.10 et III.11 avec :

- y_{fp}^{simp} est la réponse du système à la commande u_{fp}^{simp} dont les paramètres sont calculés à travers une fusion entre les paramètres des contrôleurs partiels pondérés par les validités simples :

$$u_{fp}^{simp}(k) = \frac{1}{S_g^{simp}(k)}\left[T_g^{simp}(k)\, y_c(k) - R_g^{simp}(k)\, y(k)\right] ; \qquad \text{(III.24a)}$$

$$R_g^{simp}(k) = \sum_{i=1}^{3} v_i^{simp}(k)\, R_i \ ; \qquad \text{(III.24b)}$$

$$S_g^{simp}(k) = \sum_{i=1}^{3} v_i^{simp}(k)\, S_i \ ; \qquad \text{(III.24c)}$$

$$T_g^{simp}(k) = \sum_{i=1}^{3} v_i^{simp}(k)\, T_i \ ; \qquad \text{(III.24d)}$$

- y_{fp}^{renf} est la réponse du système à la commande u_{fp}^{renf} dont les paramètres sont obtenus par une fusion entre les paramètres des contrôleurs partiels pondérés par les validités renforcées :

$$u_{fp}^{renf}(k) = \frac{1}{S_g^{renf}(k)}\left[T_g^{renf}(k)\, y_c(k) - R_g^{renf}(k)\, y(k)\right] ; \qquad \text{(III.25a)}$$

$$R_g^{renf}(k) = \sum_{i=1}^{3} v_i^{renf}(k)\, R_i \ ; \qquad \text{(III.25b)}$$

$$S_g^{renf}(k) = \sum_{i=1}^{3} v_i^{renf}(k) S_i \ ; \tag{III.25c}$$

$$T_g^{renf}(k) = \sum_{i=1}^{3} v_i^{renf}(k) T_i \ . \tag{III.25d}$$

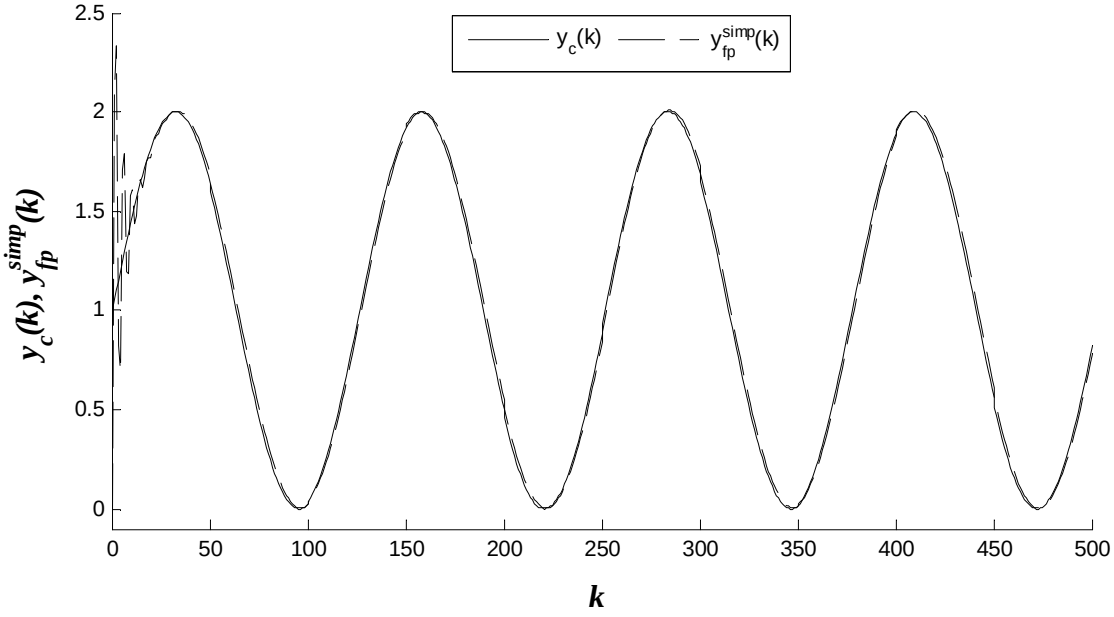


Figure III.10. Exemple 1 : Evolutions de la consigne 1 et de la réponse du système (fusion des paramètres des commandes / validités simples)

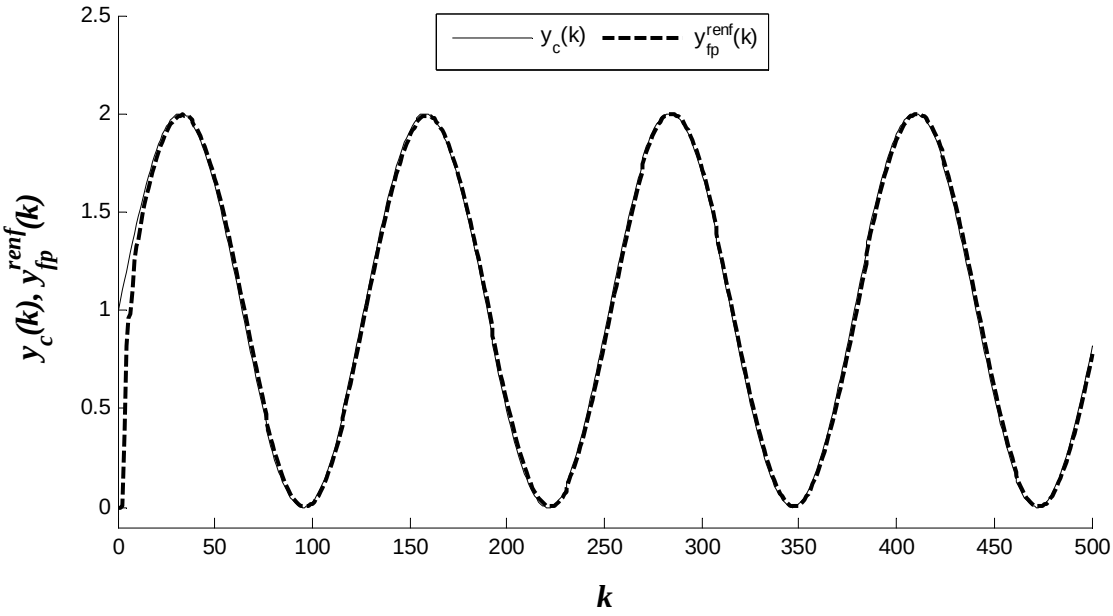


Figure III.11. Exemple 1 : Evolutions de la consigne 1 et de la réponse du système (fusion des paramètres des commandes / validités renforcées)

D'après les résultats donnés par les deux figues précédentes, on constate que la méthode des validités renforcées permet une meilleure fusion des paramètres des contrôleurs partiels. En effet, la réponse du système y_{fp}^{renf} suit presque parfaitement la consigne (figure III.11) alors qu'en utilisant les validités simples, la sortie du système y_{fp}^{simp} présente quelques oscillations avant de finir par suivre la consigne (figure III.10).

- **Interprétations sur le choix de la méthode de calcul des validités**

Qu'il s'agisse d'une fusion entre les commandes partielles ou une fusion entre leurs paramètres, on a constaté que les validités renforcées sont mieux adaptées, pour le calcul de la commande globale du système considéré, ce qui coïncide avec la déduction faite lors de la représentation multimodèle (section II.5.1 su chapitre II). Ces constatations permettent donc de conclure que l'hypothèse émise dans le chapitre II (section II.4.2) et portant sur le choix des validités à utiliser reste valable pour la commande multimodèle.

- **Comparaison entre les deux méthodes de fusion**

Ayant opté pour les validités renforcées comme étant les mieux adaptées pour l'estimation de la contribution, à chaque instant, des différents contrôleurs partiels, la comparaison des résultats obtenus suite à l'application de chacune des méthodes de fusion précédemment évoquées, est donnée par la figure III.12 sur laquelle on représente les erreurs absolues entre la consigne et la sortie du système :

$$e_{fc}(k) = \left| y_c(k) - y_{fc}^{renf}(k) \right| \ ; \qquad \text{(III.26a)}$$

$$e_{fp}(k) = \left| y_c(k) - y_{fp}^{renf}(k) \right| \ ; \qquad \text{(III.26b)}$$

sachant que y_c est la consigne donnée par l'expression (III.21).

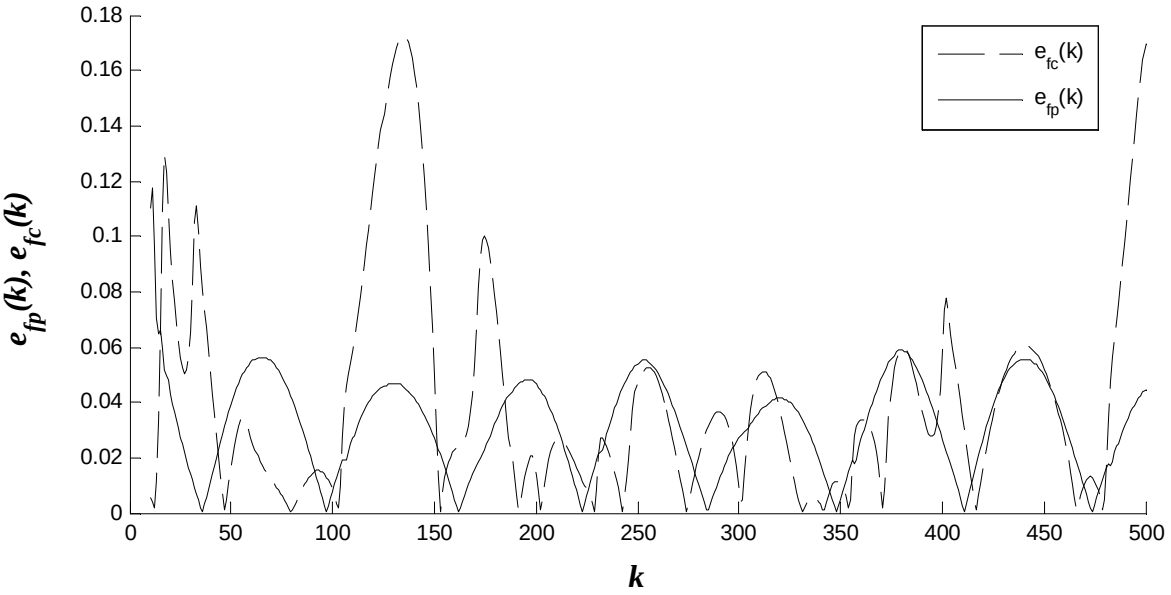


Figure III.12. Exemple 1 : Erreurs absolues entre la consigne 1 et les réponses du système obtenues par les deux méthodes de fusion

Il est clair que la commande multimodèle dont les paramètres sont calculés à travers une fusion entre les paramètres des commandes partielles aboutit à des résultats plus performants que ceux obtenus par une simple fusion de ces commandes. En effet, il apparaît sur la figure III.12 que la courbe e_{fc} présente les erreurs maximales contrairement à e_{fp} qui présente des pics réguliers qui correspondent plutôt au retard de la poursuite.

Afin de confirmer ces déductions, nous allons considérer une deuxième séquence d'entrée :

$$y_c(k) = 3 + 0.5\sin(0.2k) + 0.7\sin(0.04k)\,. \qquad \text{(III.27)}$$

Les commandes partielles u_i ($i = 1,2,3$) sont alors calculées. En se servant des validités renforcées, la commande multimodèle est calculée par les deux méthodes de fusion considérées. Les figures III13 et III.14 montrent les évolutions de la consigne et de la réponse du système obtenue par l'application de chacune de ces méthodes.

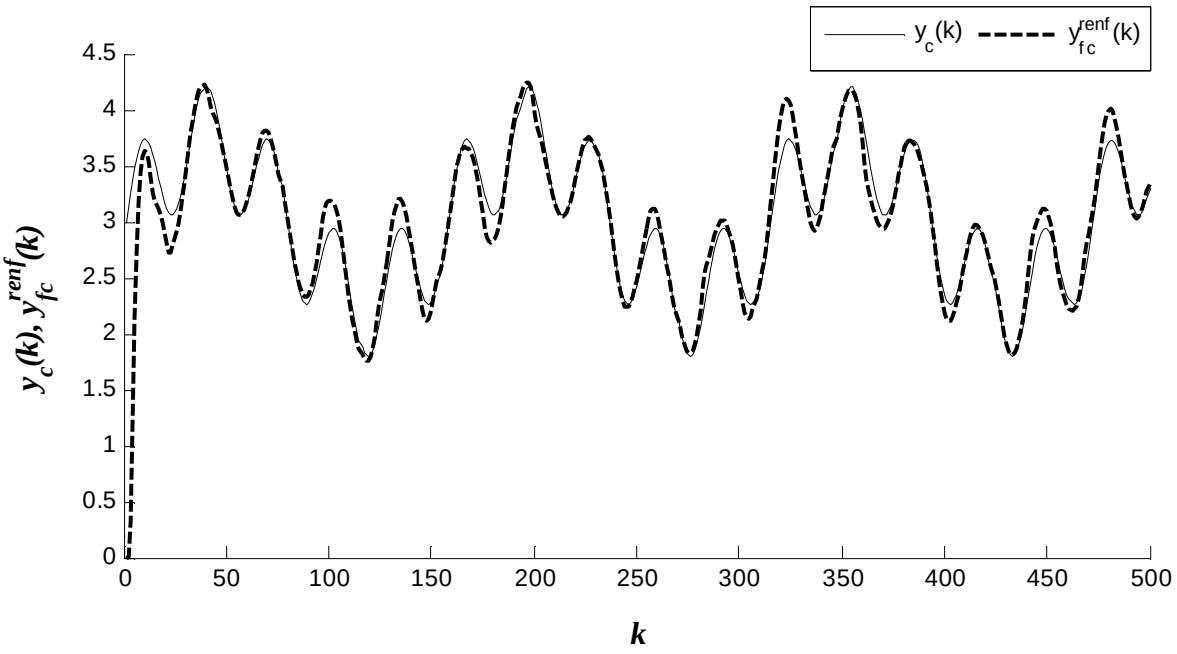


Figure III.13. Exemple 1 : Evolutions de la consigne 2 et de la réponse du système (fusion des commandes / validités renforcées)

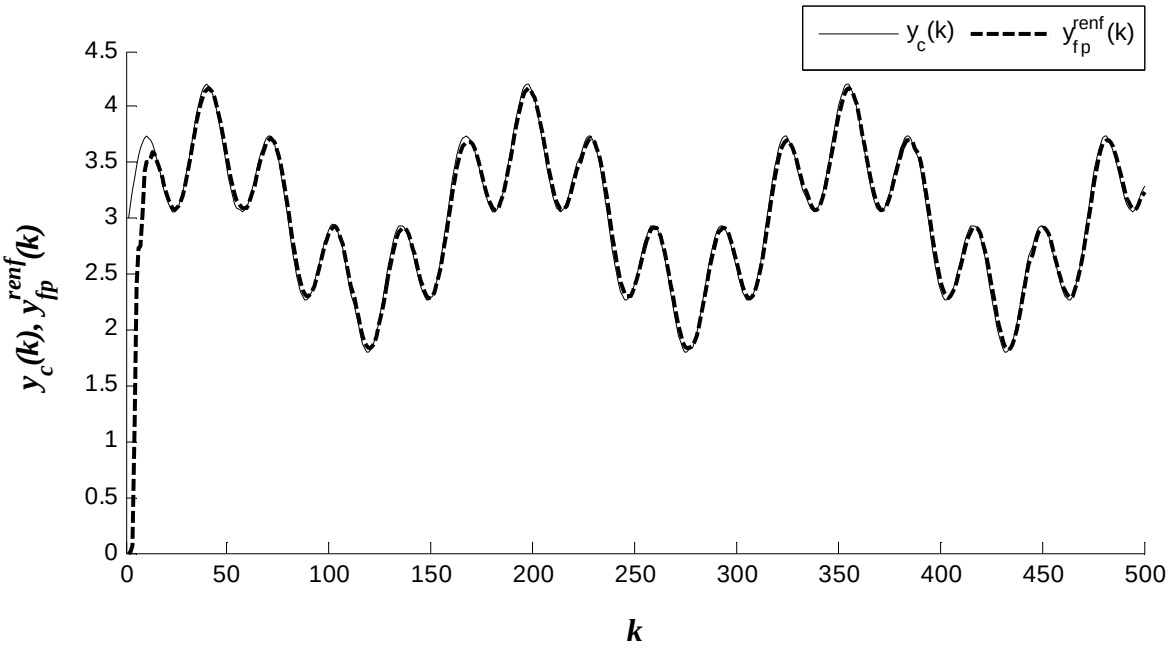


Figure III.14. Exemple 1 : Evolutions de la consigne 2 et de la réponse du système (fusion des paramètres des commandes / validités renforcées)

Pour une meilleure interprétation des résultats donnés par les deux figures précédentes, on calcule les erreurs absolues entre la consigne et les réponses du système y_{fc}^{renf} et y_{fp}^{renf}, qu'on représente sur la figure III.15.

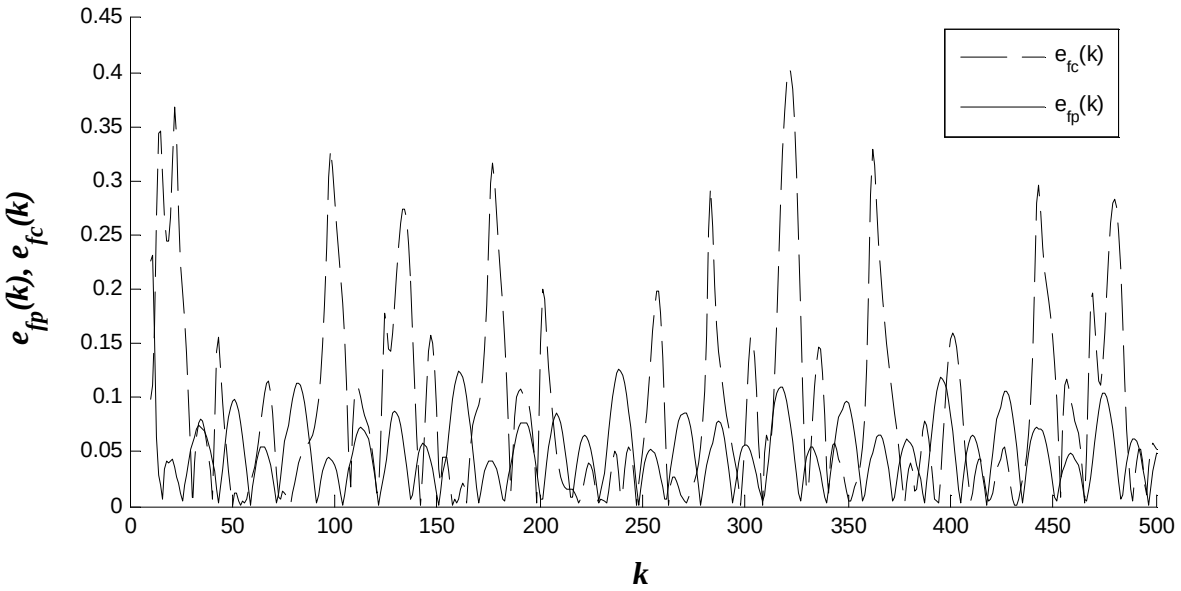


Figure III.15. Exemple 1 : Erreurs absolues entre la consigne 2 et les réponses du système obtenues par les deux méthodes de fusion

Les résultats présentés permettent de conclure que la commande multimodèle la mieux adaptée au système étudié est celle dont les paramètres sont calculés, à chaque instant, à travers une fusion des paramètres des contrôleurs partiels pondérés par les validités renforcées.

III.6.2. Exemple 2 : Commande d'un système discret non linéaire

Afin de situer notre étude par rapport aux travaux élaborés dans le domaine et mettre en évidence les apports de nos approches, le deuxième exemple de simulation est un système discret non linéaire qui a été également traité dans les travaux de [Cho et al, 07] et dont l'évolution est décrite par le système suivant :

$$\begin{cases} x_1(k+1) = x_2(k) \\ x_2(k+1) = -\dfrac{3}{16}\left[\dfrac{x_1(k)}{1+\left(x_2(k)\right)^2}\right] + x_2(k) + u(k) \\ y(k) = x_1(k) \end{cases} ; \qquad \text{(III.28)}$$

où *u* et *y* représentent respectivement l'entrée et la sortie du système.

Cet exemple n'a pas été traité dans le deuxième chapitre. C'est pour cela, avant de procéder à la commande, que nous allons donner un aperçu sur les résultats de l'implémentation de l'approche proposée dans ce mémoire pour la représentation multimodèle.

- **Représentation multimodèle du système**

Contrairement à l'étude de [Cho et al, 07] dans laquelle on propose de décrire le fonctionnement du système avec 64 modèles, nous avons réussi à obtenir des résultats comparables avec une base contenant seulement 4 modèles linéaires tout en préservant les avantages des approches multimodèles basées essentiellement sur la simplicité.

Après la collecte de données système et la détermination du nombre de modèles par l'apprentissage compétitif pénalisant le rival, la sélection de l'algorithme des K-moyennes floues comme étant le plus approprié pour la classification des mesures a permis de générer quatre classes de fonctionnement à partir desquelles quatre modèles linéaires ont été identifiés.

En se basant sur l'hypothèse émise dans le chapitre II, nous avons retenu la méthode des validités simples pour l'estimation, à chaque instant, de la contribution de chacun des quatre modèles à la génération de la sortie multimodèle après avoir constaté que les résultats de la classification présentent des classes avec un chevauchement assez important.

Afin de souligner l'efficacité et la précision de la modélisation ainsi proposée, on considère, par exemple, la séquence d'entrée suivante :

$$u_1(k) = 0.5\sin(0.2k + 1.5)\,. \qquad \text{(III.29)}$$

Sur la figure III.16, on représente la sortie réelle du système *y* et la sortie multimodèle y_{mm} calculée par une fusion entre les sorties partielles des modèles pondérées par les validités simples.

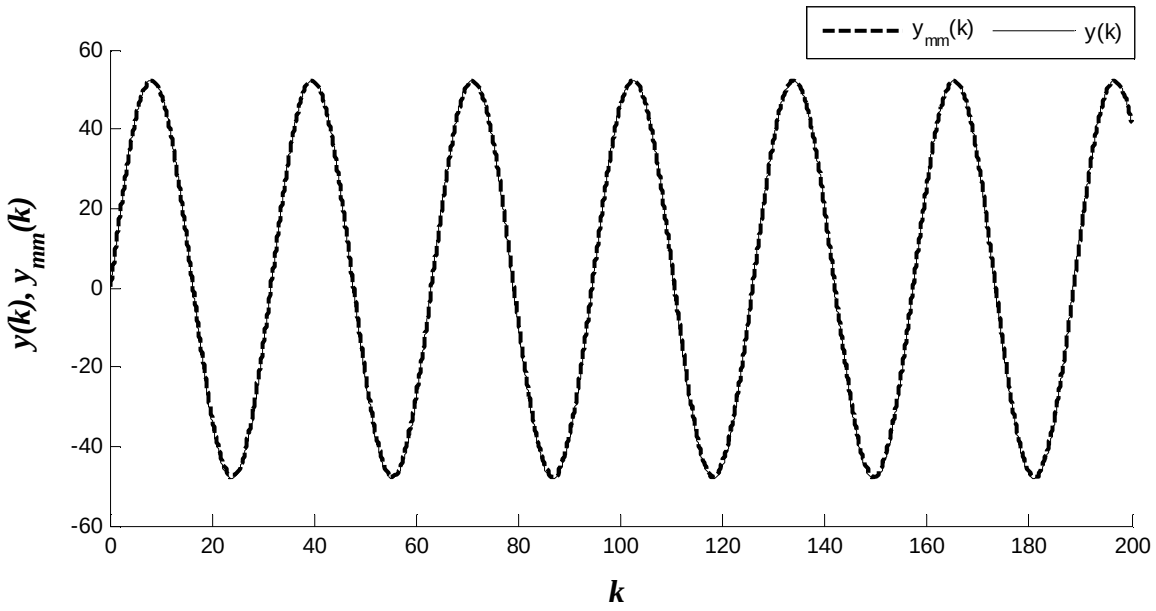


Figure III.16. Exemple 2 : Sorties réelle et multimodèle (séquence d'entrée u_1)

Afin de démonter la capacité des modèles identifiés de reproduire le fonctionnement du système dans différents domaines et pour différents types d'entrée, on se propose d'appliquer la séquence d'entrée u_2 donnée par la figure suivante.

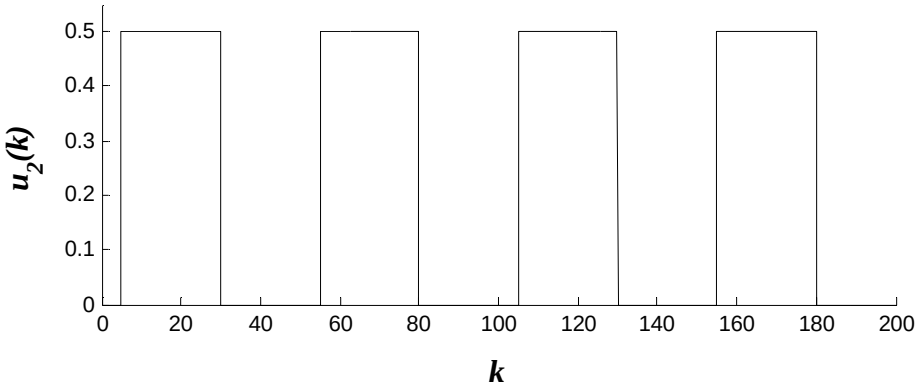


Figure III.17. Exemple 2 : Séquence d'entrée u_2

La figure III.18 donne les évolutions de la réponse du système et celle de la structure multimodèle à la séquence d'entrée u_2.

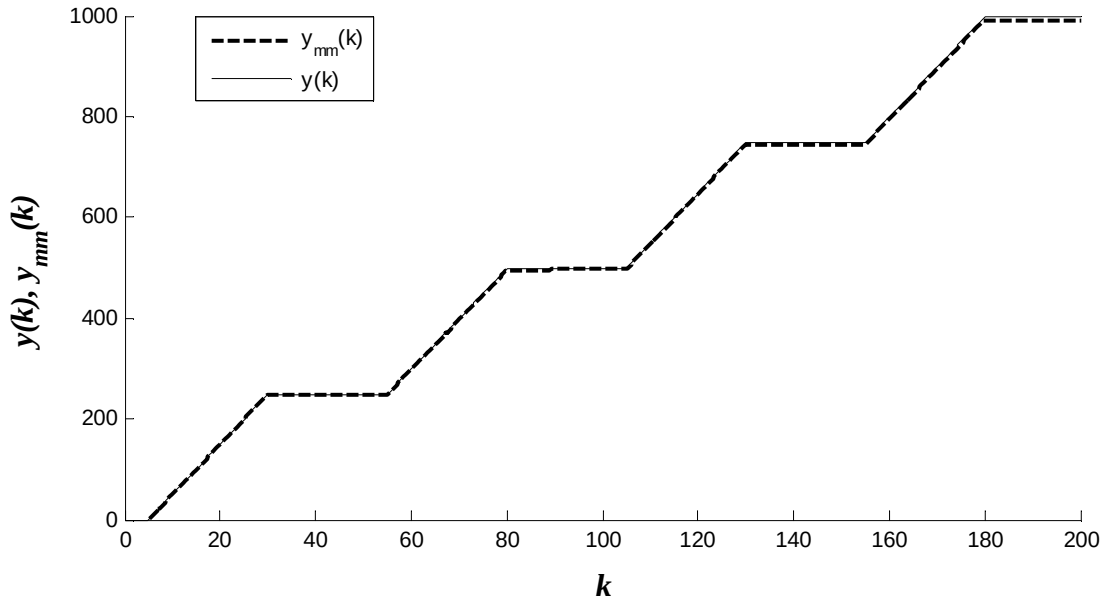


Figure III.18. Exemple 2 : Sorties réelle et multimodèle (séquence d'entrée u_2)

Les résultats obtenus témoignent de la précision et de la performance de la stratégie de modélisation proposée pour le système considéré. En effet, pour les deux séquences d'entrée u_1 et u_2, la sortie multimodèle et la sortie réelle sont presque confondues.

- **Commande multimodèle du système**

Une fois la structure multimodèle élaborée et validée, la première étape de notre approche multicommande consiste à associer un contrôleur RST à chacun des modèles de la base en imposant une dynamique adéquate.

Ayant déjà conclu que l'hypothèse de la sélection de la méthode de calcul des degrés de pertinence reste valable pour la commande multimodèle, nous avons procédé à la conception d'un contrôleur global par les deux stratégies de fusion considérées (fusion entre les commandes partielles et fusion entre les paramètres des contrôleurs) en faisant appel à la méthode des validités simples pour l'estimation de la contribution de chaque contrôleur partiel.

Dans un premier temps, on considère la consigne décrite par la séquence suivante :

$$y_{c1}(k) = 0.5 + 2\sin(0.05k)\,. \qquad \text{(III.30)}$$

Pour la première méthode de fusion, les commandes partielles u_i ($i = 1,\ldots,4$) sont alors calculées suivant la formule (III.6) et la commande globale du système est déduite comme suit :

$$u_{fc}(k) = \sum_{i=1}^{4} v_i^{simp} u_i(k) . \qquad \text{(III.31)}$$

Quant à la deuxième méthode de fusion, les paramètres du contrôleur global R_g, S_g et T_g doivent être calculés à chaque instant et loi de régulation RST permet de déduire la commande u_{fp} (expression (III.20)) à appliquer au système :

$$R_g(k) = \sum_{i=1}^{4} v_i^{simp}(k) R_i \ ; \qquad \text{(III.32a)}$$

$$S_g(k) = \sum_{i=1}^{4} v_i^{simp}(k) S_i \ ; \qquad \text{(III.32b)}$$

$$T_g(k) = \sum_{i=1}^{4} v_i^{simp}(k) T_i . \qquad \text{(III.32c)}$$

Si dans le premier exemple, la stratégie de fusion des commandes partielles aboutissait à des résultats qui n'étaient seulement pas très performants, dans le présent exemple, cette stratégie paraît complètement inappropriée avec une réponse du système très éloignée de la consigne. Ce résultat peut être confirmé par la figure III.19 qui donne les erreurs absolues entre la consigne y_{c1} et les réponses du système y_{fc} et y_{fp}, obtenues en appliquant respectivement les commandes u_{fc} et u_{fp} :

$$e_{fc}(k) = \left| y_{c1}(k) - y_{fc}(k) \right| \ ; \qquad \text{(III.33a)}$$

$$e_{fp}(k) = \left| y_{c1}(k) - y_{fp}(k) \right| . \qquad \text{(III.33b)}$$

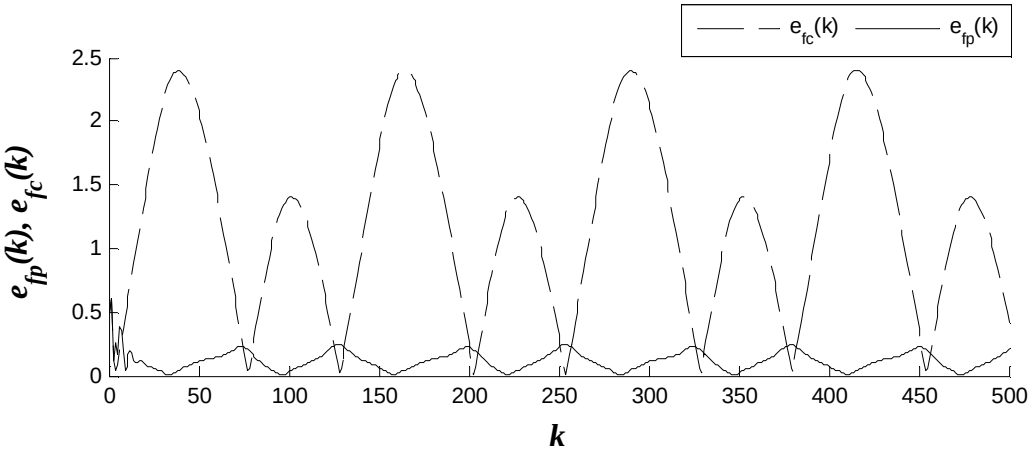


Figure III.19. Exemple 2 : Erreurs absolues entre la consigne 1 et les réponses du système obtenues par les deux méthodes de fusion

Ainsi, contrairement à la première méthode de fusion, on constate qu'une fusion des paramètres des contrôleurs partiels donne des résultats très satisfaisants (figure III.20) avec une réponse du système qui suit parfaitement la consigne.

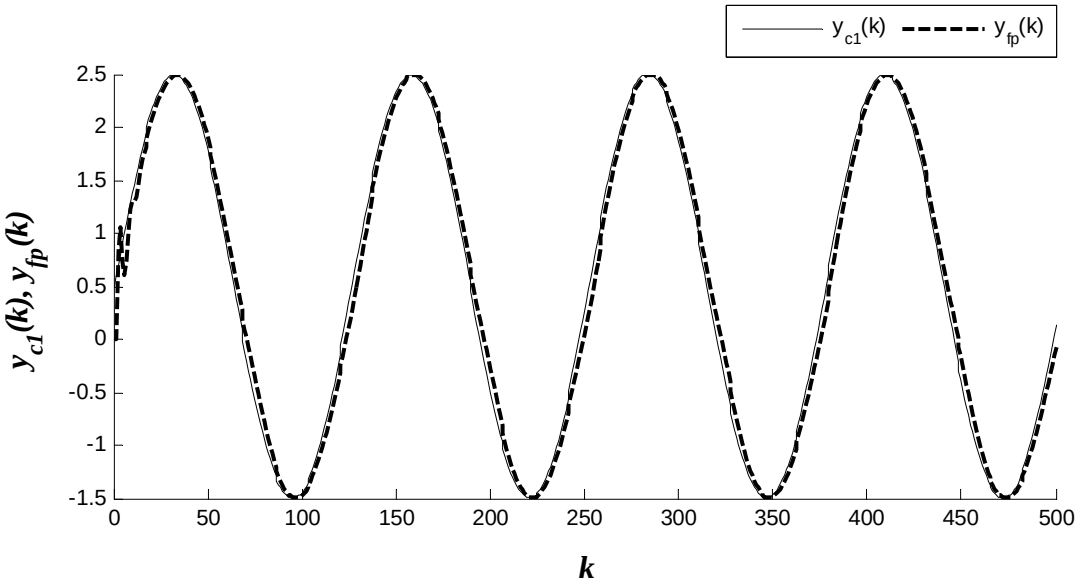


Figure III.20. Exemple 2 : Evolutions de la consigne 1 et de la réponse du système (Méthode de fusion des paramètres des commandes)

Les mêmes constatations sur l'efficacité des deux stratégies de fusion ont été faites en considérant d'autres consignes. La figure III.21 illustre les évolutions de la consigne y_{c2},

définie par la séquence (III.34), et la réponse du système y_{fp} à la commande calculée par la méthode de fusion des paramètres des contrôleurs partiels.

$$y_{c2} = 2 + 0.5\sin(0.2k) + 0.7\sin(0.04k)\,. \qquad \text{(III.34)}$$

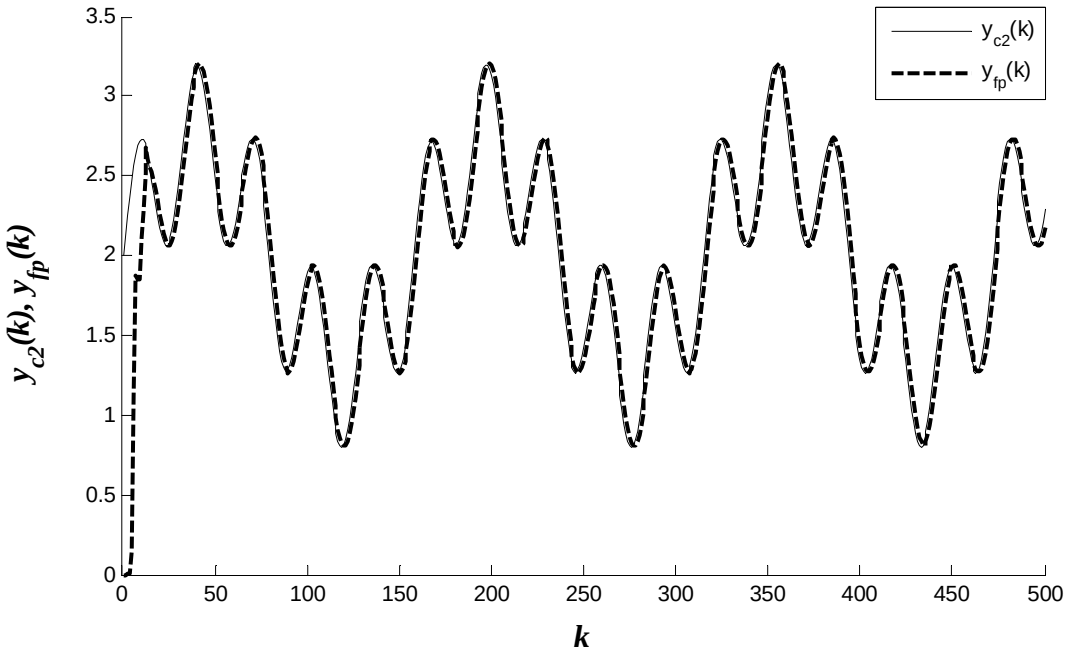


Figure III.21. Exemple 2 : Evolutions de la consigne 2 et de la réponse du système (Méthode de fusion des paramètres des commandes)

III.6.3. Interprétations des résultats

A travers les résultats obtenus suite à l'implémentation de l'approche proposée pour la commande multimodèle pour différents systèmes complexes, nous avons tout d'abord constaté que l'hypothèse de la sélection de la méthode adéquate de calcul des validités présentée dans le deuxième chapitre est applicable aussi bien pour la représentation multimodèle que pour la commande multimodèle. En effet, le choix est basé sur l'observation des résultats de la classification des données système. Dans le cas où les classes sont bien séparées et présentent peu de variations sur les frontières, la méthode des validités renforcées, permet, grâce précisément à l'étape de renforcement, d'accentuer la contribution du contrôleur correspondant au modèle dominant. Dans le cas contraire c'est-à-dire lorsqu'il s'agit de classes présentant un recouvrement assez important, la méthode des validités simples est privilégiée dans la mesure où elle permet de tenir compte de la participation de plusieurs contrôleurs partiels pour la détermination de la commande globale à appliquer au système.

D'autre part, le fait d'opter pour la même loi de commande pour tous les contrôleurs partiels de la base nous a permis de comparer les deux stratégies de fusion considérées pour le calcul de la commande multimodèle. Pour la première stratégie, la commande globale est calculée par une simple fusion entre les commandes partielles pondérées par les indices de validité. La deuxième stratégie, quant à elle, consiste à concevoir un contrôleur global dont les paramètres sont obtenus à travers une fusion entre les paramètres des contrôleurs partiels pondérés par les validités. A travers les exemples de simulation, nous avons pu conclure sur la performance et la précision de l'approche proposée et sur l'efficacité de la stratégie de fusion des paramètres des contrôleurs en la comparant à la méthode de fusion des commandes partielles qui paraît moins performante et parfois même inappropriée (exemple 2 de simulation).

III.7. Conclusion

La base de modèles étant déterminée par l'approche présentée dans le chapitre II qui est basée sur l'application de différentes méthodes de classification, la conception d'une commande multimodèle consiste à associer un contrôleur à chacun de ces modèles et ensuite à définir la stratégie du calcul de la commande globale en mettant en œuvre un mécanisme de décision permettant d'estimer la contribution de chaque contrôleur.

Dans ce chapitre, nous avons tout d'abord présenté le principe de la commande multimodèle ainsi que les différentes stratégies de la détermination du contrôleur global. Ensuite, nous avons envisagé une commande multimodèle à base de contrôleurs partiels de type RST avec deux méthodes de fusion : fusion au niveau des commandes partielles et fusion au niveau des paramètres (R, S et T) des commandes en utilisant un mécanisme de décision permettant d'estimer la contribution de chaque contrôleur partiel à la déduction de la commande.

L'implémentation de la commande multimodèle proposée sur deux exemples de systèmes complexes a permis de conclure sur la pertinence des résultats et de confirmer l'efficacité de la stratégie de fusion entre les paramètres des commandes par rapport à celle effectuée au niveau des commandes elles mêmes. Les résultats donnés dan ce chapitre ont permis aussi de vérifier que l'hypothèse émise sur le choix de la méthode adéquate de calcul des validités (évoquée dans le chapitre II) reste valable pour la commande multimodèle. D'autre part, étant donné qu'on se base sur la structure multimodèle pour la conception de la

commande globale, ces résultats témoignent d'une façon indirecte de l'efficacité et des performances de l'utilisation des méthodes de classification pour l'aide à la conception de la représentation multimodèle de processus complexes.

Conclusion générale

Les travaux présentés dans ce mémoire ont porté sur la modélisation et la commande multimodèles des systèmes complexes, fortement non linéaires et/ou incertains. La conception de la structure multimodèle est essentiellement basée sur différentes techniques de classification neuro-floues et deux méthodes d'identification structurelle et paramétrique. L'approche proposée permet ainsi de décrire un système complexe et éventuellement de type « boîte noire » à travers un certain nombre de modèles linéaires génériques en se servant seulement des mesures expérimentales de type entrée/sortie collectées sur le système. Le problème de la détermination du nombre de modèles a été résolu par l'utilisation d'un réseau de neurones et un apprentissage compétitif pénalisant le rival. Une fois ce paramètre calculé, les données système sont réparties en utilisant un algorithme de classification adapté (carte de Kohonen, K-moyennes ou K-moyennes floues) afin de générer les différents domaines de fonctionnement. Les résultats de la classification sont ensuite exploités pour l'identification des modèles de la base. Pour la génération de la sortie multimodèle, la stratégie de fusion des sorties partielles des modèles pondérées par leurs coefficients de validité respectifs, a été adoptée. A cet effet, deux méthodes de calcul des validités, basées sur l'approche des résidus, ont été présentées.

L'approche suggérée pour la représentation multimodèle se distingue essentiellement par son aspect général et pratique dans la mesure où elle est simple à mettre en œuvre, ne nécessite aucune connaissance a priori, peut être appliquée pour différents types de systèmes complexes et propose d'adapter le traitement en choisissant les méthodes adéquates de classification et de calcul des validités, suivant certains aspects de l'espace de fonctionnement du processus considéré. Avant de passer à la conception d'une commande globale du processus, la structure multimodèle proposée est testée afin de vérifier que les modèles identifiés permettent une reproduction fidèle du comportement du processus dans les différents domaines de fonctionnement. Les résultats obtenus, suite à l'implémentation de l'approche proposée pour la représentation multimodèle sur quelques exemples de systèmes dynamiques, se révèlent intéressants et témoignent de l'efficacité et de la précision de la stratégie de modélisation adoptée.

En se basant sur la structure multimodèle élaborée, la procédure de conception d'une commande multimodèle a été décrite. Elle consiste à associer un contrôleur à chaque modèle de la base et ensuite à déduire la commande globale en mettant en œuvre une stratégie par fusion et un mécanisme de décision permettant d'estimer en ligne la contribution de chaque contrôleur partiel. A travers l'étude de quelques exemples de simulation, il a été démontré que quand la même loi de commande est adoptée pour chaque modèle, une fusion entre les paramètres des contrôleurs partiels donne des résultats meilleurs que ceux obtenus par une simple fusion entre les commandes partielles. Les résultats obtenus ont permis aussi de conclure que le choix de la méthode adéquate de calcul des validités, basé sur les interprétations faites lors de la construction de la base des modèles, reste valable pour le calcul de la commande globale.

D'une façon générale, l'implémentation des différentes méthodes présentées dans le cadre de ce mémoire sur différents systèmes complexes et la comparaison avec des résultats donnés dans d'autres travaux de recherche, ont permis de conclure sur l'efficacité et la précision de l'approche proposée pour la représentation et la commande multimodèles.

Par ailleurs, partant du fait que l'approche envisagée est essentiellement basée sur l'ensemble de données système de type entrée/sortie dans le sens où il s'agit des seules informations permettant de décrire la dynamique du processus en absence de toute connaissance a priori, il serait intéressant d'envisager une étude de robustesse de cette approche vis-à-vis des perturbations externes pouvant agir lors de la collecte des mesures et de proposer une procédure de filtrage par exemple dans le cas d'un niveau important de bruit. D'autre part, il semble être utile d'étudier et d'adapter d'autres approches de calcul des validités telles que les approches floue et probabiliste pour une éventuelle amélioration des degrés de fiabilité des modèles.

Bibliographie

[Al-Akhras et al, 96] : M. A. Al-Akhras , G. M. Aly, R. J. Green, « Neural network learning approach of intelligent multimodel controller », IEE Proceedings on Control Theory and Applications, vol. 143, pp. 395-400, 1996.

[Angelov et al, 04] : P. P. Angelov, D. P. Filev, « An approach to online identification of Takagi-Sugeno fuzzy models », IEEE Transactions on Systems Man and Cybernetics, Part B, vol. 34, pp. 484-498, 2004.

[Baruch et al, 04] : I. S. Baruch, L. R. Beltran, J-L. Olivares, R. Garrido, « A fuzzy-neural multi-model for mechanical systems identification and control », Mexican International Conference on Artificial Intelligence, vol. 2972, pp. 774-783, 2004.

[Baruch et al, 08] : I. S. Baruch, R. B. Lopez, J-L. Olivares, J-M. Flores, « A fuzzy-neural multi-model for nonlinear systems identification and control », Fuzzy Sets and Systems, vol. 159, pp. 2650-2667, 2008.

[Bastin et al, 90] : G. Bastin, D. Dochain, « On-line estimation and adaptive control of bioreactors », Elsevier, Amsterdam,1990.

[Ben Abdennour et al, 01] : R. Ben Abdennour, P. Borne, M. Ksouri, F. M'sahli, « Identification et commande numérique des procédés industriels », Editions Technip, 2001.

[Ben Aoun Talmoudi, 05] : S. Ben Aoun Talmoudi, « L'approche multimodèle et les réseaux de neurones artificiels pour l'identification des systèmes complexes », Thèse de Doctorat, Ecole Nationale d'Ingénieurs de Tunis, 2005.

[Benhadj Braiek, 90] : E. Benhadj Braiek, « Application des fontions de Walsh et des fonctions modulatrices à la modélisation des systèmes continus non linéaires », Thèse de Doctorat, Université des Sciences et Techniques de Lille, 1990.

[Benhadj Braiek, 95] : E. Benhadj Braiek, « Développement de méthodes analytiques pour l'étude et la commande de processus non linéaires », Thèse de Docteur ès sciences génie électrique, Ecole Nationale d'Ingénieurs de Tunis, 1995.

[Benrejeb, 80] : M. Benrejeb, « Sur l'analyse et la synthèse de processus complexes hiérarchisés. Applications aux systèmes singulièrement perturbés », Thèse de Docteur ès-sciences physiques, Lille, 1980.

[Bezdek, 81] : J.C. Bezdek, « Pattern recognition with fuzzy objective function algorithms », Plenum Press, 1981.

[Blanco, 01] : Y. Blanco, « Stabilisation des modèles Takagi-Sugeno et leur usage pour la commande des systèmes non linéaires », Thèse de Doctorat, Université des Sciences et Technologies de Lille, 2001.

[Böling et al, 07] : J. M. Böling, D. E. Seborg, J. P. Hespanha, « Multi-model adaptive control of a simulated pH neutralization process », Control Engineering Practice, vol. 15, pp. 663-672, 2007.

[Borne et al, 07] : P. Borne, M. Benrejeb, J. Haggège, « Les réseaux de neurones », Editions Technip, 2007.

[Borne et al, 08] : P. Borne, M. Benrejeb, « On the representation and the stability study of large scale systems », International Journal of Computers, Communication and Control, vol. 3, pp. 55-66, 2008.

[Borne et al, 90] : P. Borne, G. Dauphin-Tanguy, J-P. Richard, F. Rotella, I. Zambettakis, « Commande et optimisation des processus », Technip, 1990.

[Borne et al, 92a] : P. Borne, G. Dauphin-Tanguy, J-P. Richard, F. Rotella, I. Zambettakis, « Modélisation et identification des processus, Tome 1 », Technip, 1992.

[Borne et al, 92b] : P. Borne, G. Dauphin-Tanguy, J-P. Richard, F. Rotella, I. Zambettakis, « Modélisation et identification des processus, Tome 2 », Technip, 1992.

[Borne et al, 93] : P. Borne, G. Dauphin-Tanguy, J-P. Richard, F. Rotella, I. Zambettakis, « Analyse et régulation des processus industriels, Tome 2 », Technip, 1993.

[Borne, 76] : P. Borne, « Contribution à l'étude des systèmes discrets non linéaires de grande dimension. Application aux systèmes interconnectés », Thèse de Docteur ès-sciences physiques, Lille, 1976.

[Boyd et al, 94] : S. Boyd, L. El Ghaoui, E. Feron, V. Balakrishnan, « Linear Matrix Inequalities in systems and control theory », Studies in Applied Mathematics, SIAM, 1994.

[Brockett, 76] : R. W. Brockett, « Volterra series and geometric control theory », Automatica, vol. 12, pp. 167-176, 1976.

[Buckley, 93] : J. J. Buckley, « Sugeno type controllers are universal controllers », Fuzzy Sets and Systems, vol. 53, n° 3, pp. 299-303, 1993.

[Celeux et al, 91] : G. Celeux, G. Govaert, « A classification EM algorithm for clustering and two stochastic versions », Rapport de recherche RR-1364, INRIA (Institut National de Recherche en Informatique et en Automatique), 1991.

[Cezayirli et al, 08] : A. Cezayirli, M. K. Ciliz, « Indirect adaptive control of non-linear systems using multiple identification models and switching », International Journal of Control, vol. 81, pp. 1434-1450, 2008.

[Chadli, 02] : M. Chadli, « Stabilité et commande de systèmes décrits par des multimodèles : approche LMI », Thèse de Doctorat, Institut National Polytechnique de Lorraine, 2002.

[Chapellat et al, 89] : H. Chapellat, S. P. Bhattacharyya, « An alternative proof of Kharitonov's theorem », IEEE Transactions on Automatic Control, vol. 34, n° 4, pp. 448-450, 1989.

[Chen et al, 00] : M. S. Chen, S. T. Tsao, « Exponential stabilization of a class of unstable bilinear systems », IEEE Transactions on Automatic Control, vol. 45, pp. 989-992, 2000.

[Chen et al, 92] : S. Chen, S. Billings, P. Grant, « Recursive hybrid algorithm for nonlinear system identification using radial basis function networks », International Journal of Control, vol. 55, pp. 1051-1070, 1992.

[Chen, 06] : C-W. Chen, « Stability conditions of fuzzy systems and its application to structural and mechanical systems », Advances in Engineering Software, vol. 37, pp. 624-629, 2006.

[Cheng et al, 06] : J. Cheng, Y-N. Guo, J-S. Qian, « A novel multiple neural networks modeling method based on FCM », Proceedings of the third International Symposium on Neural Networks, vol. 3973, pp. 783-789, 2006.

[Cheung, 03] : Y. M. Cheung, « K*-means: A new generalized K-means clustering algorithm », Pattern Recognition Letters, vol. 24, pp. 2883-2893, 2003.

[Cho et al, 06] : J. Cho, D. Erdogmus, M. A. Motter, « Modeling and inverse controller design for an unmanned aerial vehicle based on the self-organizing map », IEEE Transactions on Neural Networks, vol. 17, pp. 445-460, 2006.

[Cho et al, 07] : J. Cho, J. C. Principe, D. Erdogmus, M. A. Motter, « Quasi-sliding mode control strategy based on multiple-linear models », Neurocomputing, vol. 70, pp. 960-974, 2007.

[Contois, 59] : D. Contois, « Kinetics of bacterial growth relationship between population density and specific growth rate of continuous cultures », Journal of Genetic Macrobiol, vol. 21, pp. 40-50, 1959.

[Courties, 99] : C. Courties, « Sur la commande robuste et L.P.V. des systèmes à paramètres lentement variables », Thèse de Doctorat, Institut National des Sciences Appliquées de Toulouse, 1999.

[Cristianini et al, 00] : N. Cristianini, J. Shawe-Taylor, « An introduction to Support Vector Machines and other kernel-based learning methods », Cambridge University Press, 2000.

[Cutululis et al, 06] : N. A. Cutululis, E. Ceanga, A. D. Hansen, P. Sorensen, « Robust multi-model control of an autonomous wind power system », Wind Energy, vol. 9, pp. 399-419, 2006.

[Delmotte, 97] : F. Delmotte, « Analyse multimodèle », Thèse de Doctorat, Université des Sciences et Techniques de Lille, 1997.

[Dembélé et al, 03] : D. Dembélé, P. Kastner, « Fuzzy C-means method for clustering microarray data », Bioinformatics, vol. 19, pp. 973-980, 2003.

[Derese et al, 80] : I. Derese, E. Noldus, « Design of linear feedback laws for bilinear systems », International Journal of Control, vol. 31, pp. 219-237, 1980.

[Domlan 06] : E. Domlan, « Diagnostic des systèmes à changement de régime de fonctionnement », Thèse de Doctorat, Institut National Polytechnique de Lorraine, 2006.

[Du, 10] : K. L. Du, « Clustering: A neural network approach », Neural Networks, vol. 23, pp. 89-107, 2010.

[Dubois, 95] : L. Dubois, « Utilisation de la logique floue dans la commande des systèmes complexes », Thèse de Doctorat, Université des Sciences et Techniques de Lille, 1995.

[Dunn, 1973] : J. C. Dunn, « A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters », Cybernetics, vol. 3, pp. 32-57, 1973.

[Elfelly et al, 08] : N. Elfelly, J-Y. Dieulot, P. Borne, « A neural approach of multimodel representation of complex processes », International Journal of Computers, Communication and Control, vol. 3, pp. 149-160, 2008.

[Elfelly et al, 09] : N. Elfelly, J-Y. Dieulot, M. Benrejeb, P. Borne, « Approche neuronale de la représentation multimodèle de processus complexes », 3èmes Journées Doctorales / Journées Nationales MACS (JD-JN MACS 2009), Angers, 2009.

[Elfelly et al, 10a] : N. Elfelly, J-Y. Dieulot, M. Benrejeb, P. Borne, « A new approach for multimodel identification of complex systems based on both neural and fuzzy clustering algorithms », Engineering Applications of Artificial Intelligence, vol. 23, pp. 1064-1071, 2010.

[Elfelly et al, 10b] : N. Elfelly, J-Y. Dieulot, M. Benrejeb, P. Borne, « A new multimodel approach for complex processes modeling based on classification algorithms: experimental validation », Proceedings of the 12ème Symposium IFAC Large Scale Systems: Theory and Applications, 2010.

[Feliot, 97] : C. Feliot, « Modélisation des systèmes complexes : Intégration et formalisation de modèles », Thèse de Doctorat, Université des Sciences et Technologies de Lille, 1997.

[Feng, 06] : G. Feng, « A survey on analysis and design of model-based fuzzy control systems », IEEE Transactions on Fuzzy Systems, vol. 14, pp. 676-697, 2006.

[Fliess, 76] : M. Fliess, « Un outil algébrique : les series formelles non commutatives », Mathematical system theory, Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, vol. 131, pp. 122-148, 1976.

[Forgy, 65] : E. Forgy, « Cluster analysis of multivariable data: efficiency versus interpretability of classifications », Biometrics, vol. 21, pp. 768-769, 1965.

[Frank, 90] : P. M. Frank, « Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results », Automatica, vol. 26, pp. 459-474, 1990.

[Fu et al, 07] : Y. Fu, T. Chai, « Nonlinear multivariable adaptive control using multiple models and neural networks », Automatica, vol. 43, pp. 1101-1110, 2007.

[Gasso, 00] : K. Gasso, « Identification des systèmes dynamiques non linéaires : approche multimodèle », Thèse de Doctorat, Institut National Polytechnique de Lorraine, 2000.

[Gauthier et al, 92] : J. P. Gauthier, H. Hammouri, S. Othman, « A simple observer for nonlinear systems applications to bioreactors », IEEE Transactions on Automatic Control, vol. 37, pp. 875-880, 1992.

[Gegúndez et al, 08] : M. Gegúndez, J. Aroba, J. Bravo, « Identification of piecewise affine systems by means of fuzzy clustering and competitive learning », Engineering Applications of Artificial Intelligence, vol. 21, pp. 1321-1329, 2008.

[Haykin, 94] : S. Haykin, « Neural Networks: A Comprehensive Foundation », New York, Macmillan College Publishing, 1994.

[He et al, 03] : Y. He, M. Wu, « Absolute stability for multiple delay general Lur'e control systems with multiple nonlinearities », Journal of Computational and Applied Mathematics, vol. 159, pp. 241-248, 2003.

[Hore et al, 08] : P. Hore, L. Hall, D. Goldgof, W. Cheng, « Online fuzzy C means », Annual Meeting of the North American Fuzzy Information Processing Society, pp. 1-5, 2008.

[Hunt et al, 92] : K. J. Hunt, D. Sbarbaro, R. Zbikowski, P. J. Gawthrop, « Neural network for control systems (a survey) », Automatica, vol. 28, pp. 1083-1112, 1992.

[Isidori et al, 81] : A. Isidori, A. Krener, C. Gori-Giorgi, S. Monaco, « Non-linear decoupling via feedback: A differential geometric approach », IEEE Transactions on Automatic Control, vol. AC-26, pp. 331-345, 1981.

[Jain et al, 88] : A. K. Jain, R. C. Dubes, « Algorithms for clustering data », Prentice-Hall, 1988.

[Johansen et al, 93] : T. A. Johansen, B. A. Foss, « Constructing NARMAX models using ARMAX models », International Journal of Control, vol. 58, pp. 1125-1153, 1993.

[Johansen et al, 99] : T. A. Johansen, B. A. Foss, « Editorial: Multiple model approaches to modelling and control », International Journal of Control, vol. 72, pp. 575, 1999.

[Johansen, 94] : T. A. Johansen, « Fuzzy model based control: stability, robustness, and performance issues", IEEE Transactions on Fuzzy Systems, vol. 2, pp. 221-234, 1994.

[Kanzawa et al, 08] : Y. Kanzawa, Y. Endo, S. Miyamoto, « Fuzzy C-means algorithms for data with tolerance using kernel functions », IEICE Transactions on Fundamentals of Electronics, vol. 91, pp. 2520-2534, 2008.

[Kardous Khaldi, 04] : Z. Kardous Khaldi, « Sur la modélisation et la commande multimodèle des processus complexes et/ou incertains », Thèse de Doctorat, Ecole Centrale de Lille, 2004.

[Kharitonov, 78] : V. L. Kharitonov, « Asymptotic stability of an equilibrium position of a family of systems of linear differential equations », Differential Uravnen., vol.14, pp. 2086-2088, 1978.

[King et al, 98] : I. King, L. Xu, L. Chan, « Using rival penalized competitive clustering for feature indexing in Hong Kong's textile and fashion image database », Proceedings of the IEEE International Joint Conference on Neural Networks, vol. 1, pp. 237-240, 1998.

[Kohonen, 90] : T. Kohonen, « The self-organizing map », IEEE Proceedings, vol. 78, pp. 1464-1480, 1990.

[Ksouri-Lahmari et al, 04] : M. Ksouri-Lahmari, P. Borne, M. Benrejeb, « Multimodel: the construction of model bases », Studies in Informatics and Control, vol. 3, pp. 199-210, 2004.

[Ksouri-Lahmari, 99] : M. Ksouri-Lahamari, « Contributions à la commande multimodèle des processus complexes », Thèse de Doctorat, Université des Sciences et Techniques de Lille, 1999.

[Kukolj et al, 04] : D. Kukolj, E. Levi, « Identification of complex systems based on neural and Takagi-Sugeno fuzzy model », IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 34, pp. 272-282, 2004.

[Leith et al, 99] : D. J. Leith, W. E. Leithead, « Analytic framework for blended multiple model systems using linear local models », International Journal of Control, vol. 72, pp. 605-619, 1999.

[Lesiak et al, 78] : C. Lesiak, A. J. Krener, « The existence and uniqueness of Volterra series for non-linear systems », IEEE Transactions on Automatic Control, vol. 23, pp. 1030, 1978.

[Liu et al, 08] : B. Liu, Z. Jiang, K-L. Fang, « Multi-model generalized predictive control for temperature control system », Proceedings of the 5th International Conference on Fuzzy Systems and Knowledge Discovery, vol. 3, pp. 594-598, 2008.

[Lorimier et al, 03] : L. Lorimier, P. Borne, « Local observer-based multi-control and online validity estimation for multiple-model control of complex systems », IEEE International Conference on Systems, Man and Cybernetics, vol. 1, pp. 822-827, 2003.

[MacQueen, 67] : J. B. MacQueen, « Some methods for classification and analysis of multivariable observations », Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281-297, 1967.

[Mamdani et al, 75] : E. H. Mamdani, S. Assilian, « An experiment in linguistic synthesis with a fuzzy logic controller », International Journal of Man-Machine Studies, vol. 7, pp. 1-13, 1975.

[Manioudakis et al, 01] : G. D. Manioudakis, E. N. Demiris, S. D. Likothanassis, « A self-organised neural network based on the multi-model partitioning theory », Neurocomputing, vol. 37, pp. 1-29, 2001.

[Mezghani, 00] : S. Mezghani, « Approche multimodèle pour la détermination d'une commande discrète d'un système incertain », Thèse de Doctorat, Université des Sciences et Techniques de Lille, 2000.

[Mirkin, 96] : B. Mirkin, « Mathematical Classification and Clustering », Kluwer Academic Press, 1996.

[Mohler et al, 80] : R. R. Mohler, W. J. Kolodzie, « An overview of bilinear system theory and applications », IEEE Transactions on Systems, Man, and Cybernetics, vol. 10, pp. 683-688, 1980.

[Mohler, 91] : R. R. Mohler, « Nonlinear Systems – V.2: Application to Bilinear Control », Englewood Cliffs, Prentice-Hall, 1991.

[Murlidharan Nair et al, 03] : T. Murlidharan Nair, C. L. Zheng, J. Lynn Fink, R. O. Stuart, M. Gribskov, « Rival penalized competitive learning (RPCL): a topology-determining algorithm for analyzing gene expression data », Computational Biology and Chemistry, vol. 27, pp. 565-574, 2003.

[Murray-Smith et al, 97] : R. Murray-Smith, T. A. Johansen, « Multiple model approaches to modelling and control », Taylor and Francis, 1997.

[Narendra et al, 90] : K. S. Narendra, K. Parthasarathy, « Identification and control of dynamical systems using neural networks », IEEE Transactions on Neural Networks, vol. 1, pp. 4-27, 1990.

[Narendra et al, 95] : K. S. Narendra, J. Balakrishman, M. K. Ciliz, « Adaptation and learning using multiple models, switching, and tuning », IEEE Control Systems Magazine, vol. 15, pp. 37-51, 1995.

[Narendra et al, 97] : K. S. Narendra, J. Balakrishman, « Adaptive control using multiple models », IEEE Transactions on Automatic Control, vol. 42, pp. 171-187, 1997.

[Nascimento et al, 00] : S. Nascimento, B. Mirkin, F. Moura-Pires, « A fuzzy clustering model of data and fuzzy c-means », The Ninth IEEE International Conference on Fuzzy Systems, vol. 1, pp. 302-307, 2000.

[Omatu et al, 95] : S. Omatu, M. Khalil, R. Yusof, « Neuro-Control and its applications », Springer, London, 1995.

[Özkan et al, 03] : L. Özkan, M. V. Kothare, C. Georgakis, « Control of a solution copolymerization reactor using multi-model predictive control », Chemical Engineering Science, vol. 58, pp. 1207-1221, 2003.

[Packard, 94] : A. Packard, « Gain scheduling via linear fractional transformation », Systems and Control Letters, vol. 22, pp. 79-92, 1994.

[Pham et al, 95] : D. T. Pham, S. Yildirim, « Robot control using Jordan neural networks », in: Proceedings of the International Conference on Recent Advances in Mechatronics, pp. 888-893, 1995.

[Principe et al, 98] : J. C. Principe, L. Wang, M. A. Motter, « Local dynamic modelling with self-organizing maps and application to nonlinear system identification and control », IEEE Proceedings, vol. 86, pp. 2240-2258, 1998.

[Raissi et al, 01] : T. Raissi, A. El Kamel, W. Byrski, P. Borne, « Multimodel analysis and control of multivariable systems », IEEE International Conference on Systems Man and Cybernetics, Vol. 1, pp. 640-645, 2001.

[Ronen et al, 02] : M. Ronen, Y. Shabtai, H. Guterman, « Hybrid model building methodology using unsupervised fuzzy clustering and supervised neural networks », Biotechnology and Bioengineering, vol. 77, pp. 420-429, 2002.

[Rotella et al, 88] : F. Rotella, G. Dauphin-Tanguy, « Non-linear systems: identification and optimal control », International Journal of Control, vol. 48, pp. 525-544.

[Rugh et al, 00] : W. J. Rugh, J. S. Shamma, « Research on gain scheduling », Automatica, vol. 36, pp. 1401-1425, 2000.

[Sadati et al, 04] : N. Sadati, A. Talasaz, « Robust fuzzy multimodel control using variable structure system », IEEE Conference on Cybernetics and Intelligent Systems, vol. 1, pp. 497-502, 2004.

[Salcedo et al, 08] : J.V. Salcedo, M. Martínez, « Design of PDC fuzzy controllers under persistent disturbances and application in mechanical systems », Advances in Engineering Software, vol. 39, pp. 937-946, 2008.

[Salgado et al, 05] : P. Salgado, J. Boaventura Cunha, « Greenhouse climate hierarchical fuzzy modelling », Control Engineering Practice, vol. 13, pp. 613-628, 2005.

[Sasaki et al, 03] : S. Sasaki, K. Uchida, « Quadratic cost output feedback control for bilinear systems », International Journal of Systems Science, vol. 34, pp. 345-355, 2003.

[Sastry et al, 94]: P. S. Sastry, G. Santharam, K. P. Unnikrishnan, « Memory networks for identification and control of dynamical systems », IEEE Transactions on Neural Networks, vol. 5, pp. 306-320, 1994.

[Schorten et al, 99] : R. Schorten, R. Marry-Smith, R. Bjorgan, H. Gollee, « On the interpretation of local models in blended multiple model structures », International Journal of Control, vol. 72, pp. 620-628, 1999.

[Selmic et al, 01] : R. R. Selmic, F. L. Lewis, « Multimodel neural networks identification and failure detection of non-linear systems », IEEE Conference on Decision and Control, vol. 4, pp. 3128-3133, 2001.

[Stipanović et al, 00] : D. M. Stipanović, D. D. Šiljak, « Stability of polytopic systems via convex M-matrices and parameter-dependant Liapunov functions », Nonlinear Analysis: Theory, Methods & Applications, vol. 40, pp. 589-609, 2000.

[Takagi et al, 85] : T. Takagi, M. Sugeno, « Fuzzy identification of systems and its applications to modelling and control », IEEE Trans. on Systems, Man and Cybernetics, vol. 15, pp. 116-132, 1985.

[Talmoudi et al, 08] : S. Talmoudi, K. Abderrahim, R. Ben Abdennour, M. Ksouri, « Multimodel approach using neural networks for complex systems modeling and identification », Nonlinear Dynamics and System Theory, vol. 8, pp. 299-316, 2008.

[Tambe et al, 96] : S.S. Tambe, B.D. Kulkarni, P.B. Deshpande, « Elements of artificial neural networks with selected application on chemical engineering, and chemical and biological sciences », Simulation & Advanced Controls Inc., 1996.

[Tanaka et al, 98] : K. Tanaka, T. Ikeda, H. Wang, « Fuzzy regulators and observers: Relaxed stability conditions and LMI-based designs », IEEE Transactions on Fuzzy Sets, vol. 6, pp. 1-16, 1998.

[Vernieuwe et al, 06] : H. Vernieuwe, B. D. Baets, N. Verhoest, « Identification of complex systems based on neural and Takagi-Sugeno fuzzy model », Fuzzy sets and systems, vol. 157, pp. 2876-2896, 2006.

[Vieira et al, 04] : J. J. Vieira, F. M. Dias, A. Mota, « Artificial neural networks and neuro-fuzzy systems for modelling and controlling real systems: A comparative study », Engineering Applications of Artificial Intelligence, vol. 17, pp. 265-273, 2004.

[Xu et al, 92] : L. Xu, A. Krzyzak, E. Oja, « Unsupervised and supervised classification by rival penalized competitive learning », Pattern Recognition, vol. 11, pp. 496-499, 1992.

[Xu et al, 93] : L. Xu, A. Krzyzak, E. Oja, « Rival penalized competitive learning for cluster analysis RBF and curve detection », IEEE Transactions on Neural Networks, vol. 4, pp. 636-649, 1993.

[Xue et al, 06] : Z. Xue, S. Li, « Multi-model modelling and predictive control based on local model networks », Control and Intelligent Systems, vol. 34, pp. 105-112, 2006.

[Yip et al, 94] : P. P. C. Yip, Y. H. Pao, « A recurrent neural net approach to one-step-ahead control problems », IEEE Transactions on Systems, Man and Cybernetics, vol. 24, pp. 678-683, 1994.

[Yu, 06] : W. Yu, « Multiple recurrent neural networks for stable adaptive control », Neurocomputing, vol. 70, pp. 430-444, 2006.

[Zadeh 73] : L. A. Zadeh, « Outline of a new approach to the analysis of complex systems and decision processes », IEEE Transactions on Systems, Man and Cybernetics, vol. 3, pp. 28-44, 1973.

Annexe A. Approche floue de type Takagi-Sugeno

Une approche floue consiste généralement à représenter les systèmes par des implications logiques en se basant sur des données verbales dérivées de l'expertise humaine. Elle se distingue essentiellement par l'absence de modèles analytiques. La révolution dans ce domaine fût l'apparition de l'approche floue de type Takagi-Sugeno (T-S) qui, en revanche, fait intervenir des modèles analytiques dans la représentation floue d'un processus [Takagi et al, 85].

- **Structure des modèles de type Takagi-Sugeno**

On considère le système dynamique non linéaire suivant :

$$\dot{x} = f(x,u) \ ; \tag{A.1}$$

où $x \in X \subset \mathbb{R}^n$ et $u \in U \subset \mathbb{R}^m$.

L'approche floue proposée par Takagi et Sugeno consiste à décrire ce système par un ensemble de règles floues de type SI-ALORS dont la partie conclusion est exprimée sous la forme d'un modèle analytique exact [Kardous Khaldi, 04]. Dans le cas continu, le modèle flou de type T-S a la forme suivante ($i = 1,\ldots,r$) :

$i^{\text{ème}}$ règle (R_i) :

$$SI \quad z_1(t) \text{ est } Z_{i_1} \text{ ET} \ldots \text{ET } z_p(t) \text{ est } Z_{i_p}$$

$$ALORS \quad (M_i) \begin{cases} \dot{x}(t) = A_i x(t) + B_i u(t) \\ y(t) = C_i x(t) \end{cases} \tag{A.2}$$

Dans le cas discret, il s'écrit comme suit ($i = 1,\ldots,r$) :

$i^{\text{ème}}$ règle (R_i) :

$$SI \quad z_1(k) \text{ est } Z_{i_1} \text{ ET} \ldots \text{ET } z_p(k) \text{ est } Z_{i_p}$$

$$ALORS \quad (M_i) \begin{cases} x(k+1) = A_i x(k) + B_i u(k) \\ y(k) = C_i x(k) \end{cases} \tag{A.3}$$

Z_{i_j} ($j = 1, \ldots, p$) étant les sous-ensembles flous ; r le nombre de règles ; $x(t)$ (ou $x(k)$) le vecteur état ; $u(t)$ (ou $u(k)$) le vecteur entrée ; $y(t)$ (ou $y(k)$) le vecteur sortie ; $z_1(t), \ldots, z_p(t)$ (ou $z_1(k), \ldots, z_p(k)$) les variables des prémisses ; $A_i \in \mathbb{R}^{n \times n}$; $B_i \in \mathbb{R}^{n \times m}$ et $C_i \in \mathbb{R}^{q \times n}$.

A chaque règle (R_i) est attribuée une fonction de pondération : $h_i : X \times U \to [0,1]$ dépendant des degrés d'appartenance des variables des prémisses z_j aux sous-ensembles flous Z_{i_j}. Ainsi, le modèle flou global est obtenu par une défuzzification barycentrique, ce qui donne dans le cas continu :

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^{r} h_i\left(z(t)\right)\left[A_i x(t) + B_i u(t)\right] \\ y(t) = \sum_{i=1}^{r} h_i\left(z(t)\right) C_i x(t) \end{cases} \tag{A.4}$$

et dans le cas discret :

$$\begin{cases} x(k+1) = \sum_{i=1}^{r} h_i\left(z(k)\right)\left[A_i x(k) + B_i u(k)\right] \\ y(k) = \sum_{i=1}^{r} h_i\left(z(k)\right) C_i x(k) \end{cases} \tag{A.5}$$

Dans les deux cas, les fonctions de pondération doivent vérifier la propriété de somme convexe :

$$\begin{cases} h_i(z) \geq 0 \; ; \; i = 1, \ldots, r \\ \sum_{i=1}^{r} h_i(z) = 1 \end{cases} \tag{A.6}$$

Annexe B. Méthodes de renforcement des validités

Dans notre étude, nous avons considéré une seule méthode de renforcement pour le calcul des validités des modèles (section II.4.1.3). Dans ce qui suit, nous allons présenter d'autres expressions de renforcement qui ont été évoquées dans la littérature [Delmotte, 97] [Kardous Khaldi, 04].

B.1. Première méthode de renforcement

L'expression de renforcement donnée par la formule (B.1) est essentiellement utilisée pour le calcul des validités basées sur l'approche géométrique mais qui peut servir également pour l'approche des résidus :

$$v_i'^{renf} = v_i^{simp} \prod_{\substack{j=1 \\ j \neq i}}^{K} \left(1 - e^{-\left(r_j^{norm} / \sigma \right)^2} \right) ; \qquad \text{(B.1)}$$

où r_j^{norm} désigne le résidu normalisé du modèle M_j et v_i^{simp} la validité simple du modèle Mi définie dans le deuxième chapitre (expression II.29) ; σ est un réel positif convenablement choisi dans]0,1] pour moduler l'influence de la décroissance de la validité d'un modèle sur les autres. Plus σ est grand plus les transitions entre les modèles sont brusques.

Dans ce cas, si un seul modèle M_j est valide, son résidu r_j^{norm} est alors égal à zéro annulant ainsi la quantité $\left(1 - e^{-\left(r_j^{norm} / \sigma \right)^2} \right)$; ce qui implique l'annulation des validités de tous les autres modèles. Par ailleurs, un problème se pose dans le cas où deux modèles ou plus sont valides (leurs résidus sont égaux à zéro) et par conséquent tous les modèles auront des validités nulles. Pour pallier à ce problème, une méthode de renforcement généralisé peut être considérée.

B.2. Méthodes de renforcement généralisé

Le principe des méthodes de renforcement généralisé consiste à utiliser l'expression (B.1) en lui associant un test permettant de détecter s'il y a plusieurs modèles valides au

même instant et évite que leurs validités ne soient nulles. Dans ce qui suit, nous présentons les trois types de renforcement généralisé.

- **Première méthode de renforcement généralisé**

Cette méthode de renforcement généralisé considère qu'un modèle valide est un modèle « parfait » c'est-à-dire que son résidu est égal à 0. L'expression de renforcement peut être alors définie comme suit :

$$v_i'^{renf} = \begin{cases} v_i^{simp} \prod_{\substack{j=1 \\ j \neq i}}^{K} \left(1 - e^{-\left(r_j^{norm} / \sigma \right)^2} \right) & si\ n_p \in \{0,1\}\ \forall\ r_i^{norm} \\ \dfrac{1}{n_p} & si\ n_p \geq 2\ et\ r_i^{norm} = 0 \\ 0 & si\ n_p \geq 2\ et\ r_i^{norm} \neq 0 \end{cases} ; \qquad \text{(B.2)}$$

où n_p désigne le nombre de modèles parfaits.

Si aucun modèle n'est « parfait », les validités sont calculées par la première équation du système (B.2). Si un seul modèle est « parfait », la validité de celui-ci est calculé alors toutes les autres validités sont nulles.
Si deux modèles ou plus sont « parfaits », ils auront alors des validités à valeurs égales (deuxième équation du système (B.2)) et les validités de tous les autres modèles seront nulles (troisième équation du système (B.2)).

- **Deuxième méthode de renforcement généralisé**

Dans l'expression de renforcement généralisé (B.2), les tests sont effectués en considérant comme référence la valeur nulle c'est-à-dire qu'un modèle valide est un modèle ayant un résidu nul. Or, de part la nature incertaine des processus étudiés et les caractéristiques de la modélisation générique, il est très rare qu'un modèle puisse représenter parfaitement le processus et donc avoir un résidu nul. Par conséquent, il s'avère plus raisonnable de vérifier si un modèle est valide en comparant son résidu par rapport à une valeur ε très faible mais non nulle. On ne parlera plus d'un modèle « parfait » mais plutôt d'un modèle « presque parfait ».

L'expression (B.2) devient alors :

$$v_i'^{renf} = \begin{cases} v_i^{simp} \prod_{\substack{j=1 \\ j \neq i}}^{K} \left(1 - e^{-\left(r_j^{norm} / \sigma \right)^2} \right) & si\ n_p \in \{0,1\}\ \forall\ r_i^{norm} \\ \dfrac{1}{n_p} & si\ n_p \geq 2\ et\ r_i^{norm} \leq \varepsilon \\ 0 & si\ n_p \geq 2\ et\ r_i^{norm} > \varepsilon \end{cases} ; \qquad (B.3)$$

- **Troisième méthode de renforcement généralisé**

Par analogie à l'expression précédente, un autre type de renforcement généralisé peut être envisagé en considérant l'expression (II.30) des validités renforcées définie dans le deuxième chapitre de ce mémoire :

$$v_i'^{renf} = \begin{cases} v_i^{simp} \prod_{\substack{j=1 \\ j \neq i}}^{K} \left(1 - e^{-\left(r_j^{norm} / \sigma \right)^2} \right) & si\ n_p \in \{0,1\}\ \forall\ v_i^{simp} \\ \dfrac{1}{n_p} & si\ n_p \geq 2\ et\ v_i^{simp} \geq v_{seuil} \\ 0 & si\ n_p \geq 2\ et\ r_i^{norm} < v_{seuil} \end{cases} ; \qquad (B.4)$$

On note que le test est directement effectué sur les validités simples avant de procéder au renforcement. On considère comme référence une valeur de validité seuil v_{seuil} à partir de laquelle un modèle peut être jugé comme satisfaisant. Comme pour ε, la valeur de v_{seuil} doit être convenablement choisie en tenant compte des caractéristiques du processus étudié et du niveau de précision exigé. Elle doit être généralement assez proche de 1. Un modèle est donc qualifié de « presque parfait » si sa validité est supérieure ou égale à v_{seuil}.

RESUME

Les travaux présentés dans ce mémoire portent sur la représentation et la commande multimodèles de processus complexes. L'approche envisagée, essentiellement basée sur des techniques de classification neuro-floues, vise à établir une base de modèles décrivant le système dans l'ensemble de son espace de fonctionnement en se servant seulement des mesures de type entrée/sortie. L'implémentation de cette approche nécessite trois étapes principales :

(1) détermination de la structure multimodèle, pour laquelle le nombre de modèles est tout d'abord sélectionné en utilisant un réseau de neurones à apprentissage compétitif pénalisant le rival. Les différentes classes de fonctionnement sont ensuite déterminées en se servant d'un algorithme de classification adéquat (carte de Kohonen, K-moyennes ou K-moyennes floues),

(2) identification paramétrique des modèles basée sur les résultats de la classification et une procédure de validation dont l'objectif est de confirmer l'efficacité de la structure multimodèle proposée en faisant intervenir un mécanisme de décision convenable permettant l'estimation de la contribution (ou validité) de chaque modèle,

(3) calcul des paramètres du contrôleur global du système à travers une fusion entre les paramètres des commandes partielles associées aux différents modèles de la base.

L'approche suggérée se distingue essentiellement par son aspect général et pratique dans la mesure où elle est simple à mettre en œuvre, ne nécessite aucune connaissance a priori et propose d'adapter le traitement en choisissant les méthodes adéquates de classification et de calcul des validités, suivant certains aspects de l'espace de fonctionnement du processus considéré.

Mots clés : Représentation multimodèle, Commande multimodèle, Processus complexes, Classification, Apprentissage compétitif pénalisant le rival, Algorithmes des K-moyennes, Carte de Kohonen, Validités.

ABSTRACT

This contribution deals with a new approach for complex processes modeling and control. It is essentially based on neuro-fuzzy classification methods and aims to derive a base of models describing the system in the whole operating domain by using only input/output measurements. The implementation of this approach requires three main steps:

(1) determination of the multimodel stucture, for which the number of models are firstly worked out by using a neural network with a rival penalized competitive learning. The different operating clusters are then selected referring to an adequate classification algorithm (Kohonen card, K-means or fuzzy K-means),

(2) parametric model identification using the classification results and a validation procedure to confirm the efficiency of the proposed multimodel structure through an appropriate decision mechanism which allows the estimation of the contribution (or validity) of each model.

(3) determination of the global system control parameters deduced through a fusion of models control parameters.

The suggested approach seems to be interessent since it's easy to apply, doesn't require any a priori knowledge and propose to adapt the processing by choosing the adequate methods of data classification and validity computation referring to some aspects of the operating domain of the considered process.

Keywords: Multimodel representation, Multimodel control, Complex processes, Clustering, Rival penalized competitive learning, K-means algorithms, Kohonen card, Validities.