Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Ferhat Abbas, Sétif
Faculté des Sciences de l'Ingénieur
Département d'Electronique

Thèse

Présentée par

Mr. **Soukkou Ammar**

Pour l'Obtention du Diplôme de

Doctorat en Sciences

En **Electronique**

Thème

Modélisation & Commande des systèmes Industriels Complexes par les Techniques Intelligentes

Soutenue le 07 / 10 / 2008 devant le jury composé de :

Président	**:**	**Krim Fateh**	**Prof. Université Ferhat Abbès, Sétif**
Rapporteurs	**:**	**Khellaf Abdelhafid**	**Prof. Université Ferhat Abbès, Sétif**
		Leulmi Salah	**Prof. Université 20 Août 1955, Skikda**
Examinateurs	**:**	**Belarbi Khaled**	**Prof. Université M. Mentouri, Constantine**
		Charef Abdelfatah	**Prof. Université M. Mentouri, Constantine**
		Mokrani Karim	**MC. Université A. Mira, Bejaia**
		Mostefai Mohammed	**Prof. Université Ferhat Abbès, Sétif**

Remerciements

Le travail présenté dans cette thèse est le fruit de quelques années de travail dans des conditions très difficiles qui demandent beaucoup de sacrifices, de courage et de régularité. Finaliser une thèse dans un petit coin de la maison, je pense, est un très grand défi au 21ème siècle. Mais, le faite de penser à ALLAH, notre Dieu et Dieu de toutes les existences, le chemin devient plus court & simple. L'épanouissement est un cadeau donné aux personnes ayant choisir la patience & le défi comme principe de continuité…

J'adresse mes vifs remerciements à Messieurs les Professeurs *Leulmi Salah* & *Khellaf Abdelhafid* d'avoir accepté de rapporter sur mon mémoire et pour l'intérêt qu'ils ont voulu porter à ce travail. Leur lecture approfondie du mémoire, leurs remarques et critiques judicieuses m'ont été très précieuses.

Mon professeur *Leulmi* « *Ammi Salah* », avec qui j'ai eu le plaisir, de travailler, pour la confiance qu'il m'a faite et pour l'encouragement. Je le remercie particulièrement pour son côté humain, qui représente une école exceptionnelle, sa méthodologie de rédaction typique ainsi que sa démocratie. Merci infiniment *Ammi Salah* …

Mon Professeur *Khellaf AbdelHafid*, pour la confiance qu'il m'a faite en me proposant le sujet de la thèse. Je le remercie infiniment pour la grande indépendance qu'il m'a accordée durant le développement de ce modeste travail. Le faite d'être encadré par le Professeur *Khellaf* est un honneur & aussi un signe de réussite. Merci infiniment *pour l'esprit scientifique*…

C'est un grand honneur & plaisir que Monsieur Monsieur *Krim Fateh*, Professeur à l'université de Sétif préside le jury de ce modeste travail. C'est l'occasion de lui remercier pour les connaissances qui nous a donné durant ma formation : Graduation & Post graduation au sein du département d'électroniquede l'université de Sétif.

Je veux remercier d'une manière toute particulière Monsieur *Belarbi Khaled*, Professeur à l'université de Constantine pour l'intérêt qu'il a porté à ce travail en acceptant de l'examiner. Les travaux de *K. Belarbi* et al. font intervenir comme des références de base pour ce modeste travail & c'est l'occasion de lui dire merci infiniment.

C'est un grand honneur d'être examiné par Monsieur *Charef Abdelfateh*, Professeur à l'université de Constantine Je le remercie infiniment.

Mes remerciements s'adressent aussi à Monsieur *Mostefai Mohammed*, Professeur & Vice Recteur Chargé de la Post graduation à l'université de Sétif, d'avoir accepter d'examiner & d'honorer le jury de cette thèse.

C'est un grand honneur que Monsieur *Mokrani Karim*, Maître de Conférence à l'Université de Bejaia soit membre de jury de cette thèse.

Je tiens particulièrement à remercier tous les membres du Service de l'Analyse et d'Automatique des Systèmes LAAS de l'université Libre de Bruxelles qui m'ont aider à finaliser ce modeste travail tout au long de mon séjour parmi eux. Sans oublier les conseils précises & précieuses du Professeur *Raymond Hanus*.

Enfin, mes remerciements vont a tous ceux qui m'ont soutenu ou qui, d'une manière ou d'une autre, ont contribué à l'élaboration de ce modeste travail.

Dédicaces

Je dédie cette thèse
à mes chers parents,
à toute ma famille,
à ELKOTKOUT Sid Ahmed Rachid,

إهداء خاص كما كتبه الأستاذ العلامة عبد الرحمـ ـ ـ ـن الجيلالي إلى كل الأساتذة الأجلاء الدين اغترفنا من بحورهم، إلى ربة البيت و العفاف الجليلة الكريمة، قرينتي و شريكة حياتي، ورفيقتي في السير دا العمل المتواضع، المنعمة المبرورة، مثال الوفاء و الإخلاص، أستاذة الصبر و التحدي و صانعة ا د و النجاح.... أم الكتكوت سيد أحمد.

à toutes et à tous mes ami(e)s…

ملخص

يهدف هذا العمل أساسا إلى استعمال الخصائص المميزة لتقنيات الذكاء الاصطناعي، أو ما يعرف بالأنظمة الهجينة الذكية. الهدف الأساسي من هذا البحث هو إنشاء طرق ذكية جديدة ، بسيطة وفعالة للتحكم الآلي للأنظمة الديناميكية اللاخطية. المزج بين تقنيات المنطق المبهم ، الشبكات العصبونية والخوارزميات الجينية تعتبر الوسيلة الرئيسية في هذه العملية.

إن العمل الذي بين أيدينا يشرح المشاكل التي نصادفها في مرحلة تشكيل الأنظمة المبهمة أي الغامضة، والتي تختصر أساسا في :

- مشكل تقييس و تحديد أبعاد قاعدة البيانات (الجمل المبهمة) للأنظمة المبهمة أي بنيتها.
- مشكل تمثيل معلمة أو بنية محددات قاعدة المعلومات.
- مشكل الاستقرار وصلابة أو متانة بنية المراقبة.

إن محتوى هذه المذكرة يتمحور أساسا حول النمذجة والتحكم المبهم بصيغة TS1 للأنظمة اللاخطية المعقدة.

الجزء الأول من هذه المذكرة يهدف إلى تشكيل نظام تحكم آلي مستقر. لتحقيق هذه الغاية، تم ضبط إستراتيجية تهجين التقنيات الذكية : المبهمة والعصبونية. ذلك، باستعمال الشبكة RBF المبهمة مرفقة بخوارزمية هجينة. تحقيق الخصائص البنيوية لنظام التحكم RBF المبهم المقترح يبنى وفق الخطوتين التاليتين :

- تشكيل كلي لبنية ومعلمات شبكة التحكم المقترحة، اعتمادا على خوارزمية جينية متعددة الأهداف.
- استعمال طريقة الانحدار البسيط للتسوية الدقيقة للمعلمات.

تبسيط نظام التحكم، فعاليته واستقراره تعتبر كمحددات رئيسية تتدخل في تحديد مواصفات دالة الجهد.

الجزء الثاني يتمحور حول تشكيل منهجية تأخذ بعين الاعتبار عدة خصائص هيكلية ووظيفية في آن واحد. إن مفهوم النموذج المتعدد الأهداف، قد، استعمل للوصول إلى نظام تحكم مبهم مع ضمان، في نفس الوقت، استقرار حلقة التحكم المغلق. وتقوم هذه الفكرة على العمل على تدني عدد القواعد المبهمة مع التعريف بمناطق النشاط لكل متغيرات الحالة للنظام قيد الدراسة. ثمّ، حول هذه المناطق يتم إنشاء نموذج متعدد الغايات عن طريق تحويله إلى نموذج خطي. هذه المرحلة تمثل النمذجة المبهمة للنظام غير الخطي. تعتبر الخوارزميات الجينية الوسيلة المستعملة في كل مراحل تشكيل نظام التحكم المقترح. تعتمد المنهجية المقترحة على ما يلي :

- محاكاة سلوك النظام قيد الدراسة.
- تحديد مجموعة القواعد التمهيدية المبسطة.
- النمذجة المبهمة للنظام قيد الدراسة بأسلوب النموذج المتعدد.
- إيجاد شروط استقرار النظام باستخدام طريق ليابونوف.

أخيرا، قد تم ضبط الخوارزميات الجينية على مستوى هذه المرحلة كوسيلة لتحديد عوامل ضمان الاستقرار عن الطريقة المعروفة ب LMI .

نتائج المحاكاة تبرز كفاءة و فعالية الطرق المقترحة في التحكم الآلي للأنظمة الديناميكية اللاخطية. تثمين النتائج يكون عن طريق المقارنة مع النتائج المحصل عليها بعض الباحثين عن طريق تقنيات أخرى .

كلمات مفتاحيه : الأنظمة المبهمة، التحكم العصبوني المبهم، الخوارزميات الجينية، التهيئة متعددة الأهداف، الخوارزميات الهجينة، النموذج المتعدد، PDC، استقرار ليابونوف، الأنظمة الديناميكية اللاخطية .

Abstract

The principal motivation of this work is the implementation of the capacities offered by the TIA, as improved hybrid systems, in order to develop approaches for the synthesis of the simple, effective and robust controller. The formulation of the design problems of the controller was made with a combination of the fuzzy systems, neural networks and genetic algorithms techniques.

Our work milked and explains, explicitly, the problems encountered in the design phase of the fuzzy systems. It acts, primarily, to solve :

- Problems of dimensioning of the rule of the fuzzy systems (its structure).
- Problems of parameters configuration of the fuzzy knowledge base.
- Problems of stability/robustness of the control loop.

The work, presented in this thesis, is articulated, primarily, around the principal axes of the fuzzy modelling and the fuzzy controllers of TS1-type for the strongly, nonlinear complexe systems.

Initially, we aim to synthesize a law of stable and robust fuzzy controller. To satisfy this objective, a control strategy will be developed and proposed. We anticipate, also, the use of the technique of fuzzy neural networks hybridization. The fuzzy RBF topology extended to the approximate TS1 - type is, also, suggested and used. Moreover, one plans to develop and insert an effective hybrid algorithm for an optimization structural and parametric of fuzzy RBF controller suggested. Two stages of training are, also, considered and developped :

- A global optimization, based on the multi objective genetic algorithm, from the structure & parameters design of the network controller will be developed and applied.
- The use of the gradient descent technique is anticipated for a fine-tuning of the parameters of the consequences part of the rules.

The simplicity of the control law, the effectiveness, the stability and the robustness are going to be regarded as a significant factors intervening in the performance criterion to be optimized.

In the second phase, our research consists to develop a methodology of synthesis allowing to consider, explicitly and simultaneously, several structural and functional specifications. The multi model concept is exploited for the synthesis of a fuzzy control law while ensuring closed loop stability. The idea consists in determining a reduced set of rules with the definition of the operation zones (membership function) for each state variable of the system. Around these zones, the multi model will be established by the linearization of the model of the system. This phase represents the fuzzy modelling of the nonlinear system. The entire behaviour results from the "fusion", by the formalism of TS, of the set of the local behaviours. The genetic algorithms represent the optimization tool intervening in all the stages of synthesis of the control law. The algorithm developped in this contribution consists of :

- Simulation of the behaviour of the studied process.
- Initialization of the reduced rule base.
- Fuzzy modelling of the process by the multi model approach.
- Establishment of the stability conditions of the complete model by the Lyapunov technique.

The genetic approach is adapted in this last stage. The LMI method is replaced by a genetic exploration of the space of research of the parameters to justify the stability conditions.

For each control structure, results of simulation will be presented to show and justify the desired performances. The validation of our results will be confronted with those obtained by other referred techniques.

Key words :

Fuzzy systems, Fuzzy neural controller, Genetic algorithms, Multiobjective optimization, Hybrid learning, Multimodel, PDC, Stability & robustness, Nonlinear systems.

Résumé

La motivation principale de ce travail est la mise en oeuvre des capacités offertes par les TIA, en tant que systèmes hybrides améliorés, en vue de développer des approches pour la synthèse des lois de commande simples, efficaces et robustes. La formulation des problèmes de commande a été faite avec une combinaison des SIF, des RNA et des AG. Notre travail traite et explique, explicitement, les problèmes rencontrés dans la phase de conception des SIF. Il s'agit, essentiellement, de :

- Problèmes de dimensionnement de la base des règles floues du SIF (sa structure).
- Problèmes de configuration des paramètres de la base de connaissances.
- Problèmes de stabilité/robustesse de la boucle de contrôle.

Les travaux, présentés dans cette thèse, s'articulent, essentiellement, autour des principaux axes de la modélisation et de la commande floue du type TS1 pour des systèmes, fortement, non linéaires complexes.

En premier lieu, on vise à synthétiser une loi de commande floue stable et robuste. Pour satisfaire cet objectif, une stratégie de commande va être développée et proposée. Nous anticipons, aussi, l'emploi de la technique de l'hybridation neuronale floue. La structure RBF - floue étendue au raisonnement approximatif du type TS1 est, elle - aussi, proposée et utilisée. De plus, on envisage à développer un algorithme hybride efficace pour une optimisation structurelle et paramétrique du contrôleur RBF - flou proposé. Deux étapes d'apprentissage sont, aussi, évoquées :

- Une optimisation globale, basée sur un algorithme de type AG multiobjectif, de l'ensemble structure & paramètres du réseau contrôleur va être développée et appliquée.
- L'utilisation de la technique de la descente du gradient est anticipée pour un réglage plus fin des paramètres des conséquences des règles.

La simplicité de la loi de commande, l'efficacité, la stabilité et la robustesse vont être considérées comme facteurs intervenant dans le critère des performances à optimiser.

En seconde phase, notre recherche consiste à développer une méthodologie de synthèse permettant la prise en compte, explicite et simultanée, de plusieurs spécifications structurelles et fonctionnelles. Le concept du multimodèle est exploité pour la synthèse d'une loi de commande floue tout en assurant la stabilité en BF. L'idée consiste à déterminer un ensemble réduit de règles avec la définition des zones de fonctionnement (fonction d'appartenance) pour chaque variable d'état du système. Autour de ces zones sera établi le multimodèle par la linéarisation du modèle de système. Cette phase représente la modélisation floue du système non linéaire. Le comportement global résulte de la « fusion » par le formalisme de TS de l'ensemble des comportements locaux. Les AG représentent l'outil d'optimisation intervenant dans toutes les étapes de synthèse de la loi de commande. L'algorithme développé dans cette contribution consiste en :

- Simulation du comportement du procédé à étudier.
- Initialisation de la base des règles réduite.
- Modélisation floue du procédé par l'approche multimodèle.
- Etablissement des conditions de stabilité du modèle complet par l'approche énergétique de Lyapunov.

L'approche génétique est adaptée dans cette dernière étape. La méthode LMI est remplacée par une exploration génétique de l'espace de recherche des paramètres justifiant les conditions de la stabilité.

Pour chaque structure de commande, des résultats de simulation vont être présentés pour montrer et justifier les performances désirées. La validation de nos résultats sera confrontée à ceux obtenus par d'autres techniques citées en références.

Mots – clés :

Systèmes flous, Commande neuronale floue, Algorithmes génétiques, Optimisation multiobjectif, Algorithmes hybrides, Multimodèle, PDC, Stabilité & robustesse, Systèmes non linéaires.

Table des matières

Chapitre 3
Synthèse d'une loi de commande floue à configuration minimale

Chapitre 4
Concept de multimodèle : De l'analyse à la synthèse des SIF

Liste des tableaux

Liste des figures

Liste des acronymes

Pour le contenu de la thèse un certain nombre d'acronymes, d'usage courant, a été, volontairement, employé le long de ce travail. Ces abréviations sont, explicitement, définies ci-dessous. Afin de faciliter la tâche du lecteur, le repérage de ces acronymes est établi par ordre alphabétique.

AE	Algorithmes Evolutionnaires
AEN	Action Evaluation Network
ANFIS	Adaptive Neural Fuzzy Inference System
AG	Algorithme(s) Génétique(s)
AGM	Algorithme Génétique Multiobjectif
ASIC	Application-Specific Integrated Circuit
ASN	Action Selection Network
BF	Boucle Fermée
BIBO	Bound Input-Bound Output
BMI	Bilinear Matrix Inequality (Inégalité Matricielle Bilinéaire)
BP-ALM	BP with Adaptive Learning Rate and Momentum coefficient
CANFIS	Coactive Adaptive Neural Fuzzy Inference System
CDF	Compensation et Division pour modèles Flous
CGA	Chaos-Genetic Algorithm
CSGA	Chaos-Search Genetic Algorithm
CSTR	Exothermic Continuous Stirred Tank Reactor
DFNN	Dynamic Fuzzy Neural Network
DOF	Degree Of Freedom
DSP	Digital Signal Processing
FALCON	Fuzzy Adaptive Learning Control Network
FCM	Fuzzy-C-Means
FIR	Finite Impulse Response
FPGA	Field-Programmable Gate Array
GARIC	Generalized Approximate Reasoning based Intelligent Control
G-FGS	Generalized Fransconi-Gori-Soda neurons
IAE	Integral of Absolute Error
ITAE	Integral of the Time weighted Absolute Error
IMC	Internal Model Control
IIR	Infinite Impulse Reference
LF	Logique Floue
LMI	Linear Matrix Inequality (Inégalité Matricielle Linéaire)
LSE	Least Square Estimation
MGA	Multiobjective Genetic Algorithms
MIMO	Multiple Input Multiple Output (Entrée Multiple Sortie Multiple)
MPG	Modus Ponens généralisé
NB	Negative Big

NEFPROX	NEuro-Fuzzy Function Approximator
NEFCLASS	NEuro-Fuzzy Classification
NFR	Neuro-Floues Récurrent(e)(s)
NL-P	Nonlinear Proportional controller
NL-PI	Nonlinear Proportional-Integral controller
NL-PD	Nonlinear Proportional-Derivative controller
NL-PID	Nonlinear Proportional-Integral- Derivative controller
NM	Negative Medium
OLS	Orthogonal Least-Squares
PB	Positive Big
PDC	Parallel Distributed Compensation (loi de commande basée sur le retour d'état)
PID	Proportional-Derivative-Integral
PMC	Perceptron MultiCouches
PM	Positive Medium
PSO	Particle Swarm Optimisation (méthode des essaims particulaires)
RBF	Radial Basis Function
RFCMAC	Recurrent Fuzzy Cerebellar Model Articulation Controller
RFNN	Recurrent Fuzzy Neural Network
RNA	Réseaux de Neurones Artificiels
RNF	Réseaux Neuro-Flous
RSONFIN	Recurrent Self Organizing Neural Fuzzy Inference Network
SAB	Self-Adaptive Back-Propagation
SAM	Stochastic Action Modifier
SANFIS	Self-Adaptive Neuro-Fuzzy Inference Systems
SEFC	Symbiotic Evolution based Fuzzy Controller
SIF	Systèmes d'Inférence Floue
SIMO	Single Input Multiple Output (Entrée Simple Sortie Multiple)
SOFLC	Self-Organizing Fuzzy Logic Controller
SVD	Singular Value Decomposition
TIA	Techniques de l'Intelligence Artificielle
TNFIN	Tsukamoto-type Neural Fuzzy Inference Network
TRFN	Takagi-Sugeno Type Recurrent Fuzzy Network
TSRFNN	Takagi-Sugeno Recurrent Fuzzy Neural Networks
TS0	Takagi-Sugeno d'ordre zéro
TS1	Takagi-Sugeno d'ordre un
ZE	Zero Environ

Introduction générale

A – Etude préliminaire

La conception et la mise en oeuvre des systèmes de modélisation et de contrôle avancées des procédés industriels complexes sont des tâches indéniables et nécessaires face aux exigences des nouveaux développements technologiques. La complexité de ces procédés exige la construction des algorithmes, étroitement, adaptés aux situations critiques et même très critiques.

Deux modes de contrôle automatique des procédés sont, perpétuellement, convoités :

- Les premiers, dits conventionnels, sont basés sur les mathématiques de l'ensemble contrôleur/procédé.
- Les seconds sont au centre d'une discipline scientifique, relativement, nouveaux et en tout cas controversée, qu'on appelle les techniques de l'intelligence artificielle « TIA ». Les TIA caractérisent l'ensemble des algorithmes et techniques, connues aussi sous l'appellation anglophone de « *soft computing »,* introduites par L.A. Zadeh en 1994, comme *un moyen de construire des systèmes intelligents répondant:*
  - *Aux obligations d'efficacité, de robustesse, de facilité d'implémentation.*
  - *A l'optimisation de coûts temporels, énergétiques, financiers, …etc.*

Ces nouvelles techniques doivent prendre, aussi, en compte la présence de la composante humaine et surtout de son expérience dans les systèmes à étudier et à analyser [1]. Les principales techniques sont : La logique floue « LF », les réseaux de neurones « RNA » et les algorithmes génétiques « AG » [2] - [4]. Les 2 premiers éléments des TIA « RNA & LF » tentent à modéliser le fonctionnement du cerveau humain. Les RNA tentent à modéliser l'architecture neuronale du cerveau. Les systèmes flous, quant à eux, modélisent son mode de fonctionnement (raisonnement) : Apprentissage et déduction. Par contre, les AG servent à modéliser le processus de l'évolution génétique des espèces vivants.

Ces dernières années, le développement de nouveaux concepts ou méthodologies et algorithmes, dans le domaine de l'intelligence artificielle, a fourni des outils alternatifs pour aborder le problème de modélisation et de contrôle des systèmes non linéaires (sciences de l'ingénieur). Le but visé par les chercheurs est de construire des systèmes artificiels qui retiennent les mécanismes importants des systèmes naturels.

L'accroissement de la robustesse, le contrôle non linéaire et le contrôle adaptatif sont des principaux objectifs pris en charge par de nouvelles stratégies de contrôle. Elles mettent en œuvre des méthodes avancées de contrôle. Les algorithmes de contrôle intelligent figurent parmi les nouvelles tendances de contrôle des systèmes complexes.

L'application des TIA ainsi que leurs hybridations, pour la modélisation et la commande des systèmes fortement non linéaires, peuvent fournir des solutions bien adaptées à la complexité des procédés industriels.

La logique floue excelle dans la représentation de connaissances imprécises ou incomplètes [2]. Elle constitue une interface commode pour la modélisation du langage naturel, en particulier des concepts linguistiques utilisés par les experts d'un procédé. Pour la gestion des procédés complexes, l'expert se heurte à la difficulté de formalisation des règles. Les données expérimentales (numériques) constituent une source de connaissance

complémentaire. L'apprentissage à partir de données permet de reproduire les relations qui peuvent exister entre les entrées et les sorties d'un tel procédé. Takagi & Sugeno ont été les premiers à proposer de tels systèmes conçus à partir des données numériques (identification). Ils représentent un second type de systèmes d'inférence floue « SIF » [5].

Dubois & Prade ont constaté qu'avec le temps et avec les techniques développées pour la conception des SIF à partir des données numériques, cette technologie s'éloigne du flou pour devenir, simplement, un outil d'approximation de fonctions. L'interprétabilité peut être vue comme une contrainte qui permet de passer du système de type boite noir vers un système flou. Si, elle est complètement relâchée, le SIF se comporte comme un approximateur et vise, simplement, à améliorer les performances numériques [6].

La capacité de l'approximation universelle des SIF représente la plateforme ou la *justification théorique* de presque toutes les recherches théoriques et de leurs applications dans le domaine de l'identification et de contrôle flou [7] - [10].

La logique floue aide à gérer des systèmes complexes de façon simple et facilement explicitable par l'expertise humaine. La difficulté de sa mise en oeuvre réside dans la mise au point des paramètres et des fonctions d'appartenance. Ce problème peut être résolu par l'utilisation des méthodes d'extraction automatique des connaissances : Les méthodes analytiques, les TIA, ...etc. (§ chapitre 2).

La synthèse d'un SIF présente un aspect primitif et artisanal indéniable. Le choix des fonctions d'appartenance, de leur nombre, de la défuzzification, voir même de l'inférence floue, est très arbitraire. Finalement, il est important de mentionner la difficulté de garantir la cohérence & l'interprétabilité des règles floues, en particulier, pour des systèmes multivariables où le nombre des règles devient très élevé. Le réglage par essais successifs de ces nombreux paramètres est, souvent, assez long et fastidieux [11].

Diverses techniques d'optimisation et d'apprentissage basées sur différentes approches ont été développées ces dernières années. Les méthodes hybrides intègrent toutes sortes d'outils, dont les plus utilisés sont les RNA et les algorithmes évolutionnaires « AE ». Les AE sont susceptibles de trouver un optimum global et permettent d'optimiser la structure et les paramètres d'un SIF. Les RNA ont apportés aux SIF leurs algorithmes d'apprentissage et leur précision dans l'ajustement numérique. Ce type d'outils peut être appréciable lorsque :

- La sémantique est secondaire par rapport aux performances numériques.
- La gestion des problèmes, pour lesquelles aucune connaissance experte, n'est en aucun cas disponible.

L'objectif des différentes hybridations possibles, entre autre, LF-AG, LF-RNA et LF-RNA-AG est de développer des systèmes hybrides qui réunissent les capacités d'apprentissage des RNA, d'optimisation des AG et la lisibilité et la souplesse des éléments manipulés par les SIF.

B – Formulation du problème

Cette étude consiste à la mise en évidence des propriétés et des avantages des TIA pour la synthèse des lois de commandes floues des systèmes dynamiques. Ces lois assurent la stabilité et la robustesse des structures de commande par rapport aux perturbations de différentes formes qui peuvent être dues, soit, aux perturbations environnementales de fonctionnement soit aux problèmes d'exploitation internes. La synthèse d'une loi de commande se fait, généralement, en 3 étapes :

- Elaboration d'un modèle de commande.
- Détermination de la structure et des paramètres d'un contrôleur en fonction des performances désirées.
- Mise en œuvre de l'algorithme de contrôle.

Lors de la synthèse d'une loi de commande, les performances recherchées s'articulent souvent autour de 3 types de spécifications, à savoir :

- La simplicité d'implantation et de mise en œuvre.
- La précision.
- La stabilité et la robustesse face aux perturbations.

Une partie de cette thèse vise à synthétiser une loi de commande floue stable et robuste. L'objet principal porte sur une structure RBF-floue avec un nombre réduit de couches (des règles). Un AGM est dédié à l'optimisation de l'ensemble structure & paramètres de la loi de commande. L'apprentissage plus fin des paramètres de cette loi permet d'améliorer les performances désirées.

Une deuxième partie concerne la modélisation et le contrôle flou des systèmes, fortement, non linéaires. Pour procéder à une modélisation, nous avions besoin de 3 principaux ingrédients, à savoir [12] :

- Un champ d'application.
- Une structure du modèle.
- Un critère de sélection.

Selon R. Hanus, un bon modèle doit :

- Etre aussi simple que possible.
- Avoir un domaine de validité aussi large que possible.
- Le tout doit avoir un critère de sélection ayant, préférablement, un sens.

Il est évident que les 2 premières qualités, que nous désirons conférer à un modèle, sont souvent contradictoires ou en compromis. Plus un modèle est simple, plus petit est son domaine de validité. Plus un domaine de validité est grand et plus complexe est le modèle [12].

Devant la difficulté de modélisation et de commande des systèmes non linéaires fortement complexes, on est souvent amené, à partir des considérations physiques, à considérer certaines classes de systèmes manipulables par des outils mathématiques existants. L'approche locale consiste à approximer le modèle au voisinage d'un point de fonctionnement [13]. Le modèle linéaire résultant n'est qu'une description locale du comportement du système. Le caractère local est l'inconvénient majeur de cette représentation.

L'approche globale ou tout simplement le concept de multimodèle consiste à représenter le modèle autour de différents points de fonctionnement. Le multimodèle peut être obtenu soit par transformation directe d'un modèle affine en l'état ou par linéarisation autour de différents points de fonctionnement, soit par identification à partir des données d'entrées/sorties [5] & [14] - [15].

L'utilisation du concept multimodèle, c'est-à-dire, des modèles linéaires interconnectés par des fonctions non linéaires permet, alors, de se ramener dans un contexte où il est possible d'utiliser des outils de la théorie des systèmes linéaires. L'outil de synthèse des lois de commande privilégié dans le cadre des modèles TS est l'approche directe de Lyapunov. La fonction candidate la plus, couramment, utilisée est de forme quadratique. La loi de commande, couramment, utilisée est basée sur le concept de commande par retour d'état. Cette idée introduite par Wang et al. est appelée dans le cas d'un SIF de type TS : Compensation parallèle distribuée «PDC» [16]. L'obtention du contrôleur flou PDC consiste à déterminer les matrices des gains de retour d'état satisfaisant les conditions de stabilité. Ce problème se réduit à un problème de faisabilité et peut être résolu à l'aide des outils issus de l'optimisation convexe et plus particulièrement des LMI.

L'étude de la stabilité constitue une phase importante dans l'analyse des comportements dynamiques d'un système en BF. La représentation multimodèle est bien adaptée au mode d'analyse de la stabilité au sens énergétique de Lyapunov.

La résolution d'un problème de commande en automatique comprend, le plus souvent, 2 étapes : L'analyse & la synthèse. La synthèse est alors suivie par une phase d'analyse pour vérifier si toutes les spécifications mentionnées dans le cahier de charges sont satisfaites. La stratégie d'analyse et de synthèse des lois de commande dans le contexte multimodèle est conçue autour de 3 axes :

- Le modèle flou issu du processus à commander par linéarisation autour des points de fonctionnement choisis. Le modèle flou modélise d'une manière approximative le processus à étudier.
- La loi de commande floue par retour d'état.
- Des conditions de stabilité du modèle en BF.

Le développement d'une stratégie de commande doit assurer non seulement la stabilité, mais aussi la robustesse en présence de phénomènes perturbateurs.

La mise en œuvre des diverses méthodes et techniques d'étude de la stabilité des systèmes dynamiques est d'autant plus difficile que leur dimension est élevée et que leur structure est complexe. Elle nécessite parfois le développement d'approches de résolution de l'équation de Lyapunov et de LMI [17] - [18].

C – Objectifs de ce travail

Les travaux présentés dans cette thèse ont pour objectif de montrer les capacités des SIF :

- A commander les systèmes complexes.
- De modéliser les dynamiques de ces systèmes.
- De concevoir des structures de commande non linéaires stables et robustes.

Notre travail a été motivé par les problèmes rencontrés dans la phase de conception des SIF. Il s'agit, essentiellement, de résoudre les :

- Problèmes de dimensionnement de la base des règles floues du SIF, c'est-à-dire, sa structure.
- Problèmes de configuration des paramètres de la base de connaissances.
- Problèmes de stabilité/robustesse de l'ensemble contrôleur/procédé à commander.

D – Contributions

*En premier lieu***,** le problème de modélisation consiste à choisir une structure appropriée du SIF puis à concevoir des lois d'ajustement des paramètres satisfaisant un critère de performances prédéterminé par le concepteur.

La première contribution de cette thèse concerne l'établissement d'une loi de commande floue efficace, stable & robuste. Le choix du type de raisonnement, les contraintes conceptuelles, la méthode d'optimisation, le mode de représentation des informations et le critère des performances désirées sont des facteurs significatifs intervenant dans la phase de conception. *Une RBF-floue à structure minimale a été développée.* L'apprentissage hybride intégrant une couche d'optimisation génétique suivie par une descente du gradient permet d'avoir un compromis entre structure optimale & paramètres optimaux. Dans ce cas, la fonction à optimiser tient compte des performances du système à évaluer, de la complexité de la structure de contrôle et de l'effort énergétique à appliquer.

L'inclusion des termes structurels (nombre des règles) dans le critère des performances à minimiser en plus des termes fonctionnels (erreurs de poursuite, efforts

énergétiques) constituent *l'aspect multiobjectif* de la méthode d'optimisation. L'introduction des contraintes de conception dans le processus d'optimisation est présentée dans cette partie de la thèse. Des contraintes sur les limites du vecteur des paramètres à identifier, des limites sur les grandeurs de commande et des variables d'état du procédé à commander et des équations contraignantes assurent ou plus tôt constituent un bon compromis entre les objectifs de conception mentionnés par le cahier des charges : L'efficacité & la stabilité robuste. La fonction objective doit être, explicitement ou implicitement, dépendante de l'ensemble des paramètres de conception. Le but principal de l'optimisation consiste à :

- Trouver une représentation structurelle/paramétrique adéquate pour le réseau RBF-flou.
- Etablir un jeu de contraintes permettant de préserver un compromis entre les objectifs de conception tout au long du processus d'optimisation.

Nous avons décomposé la construction de la base de connaissances floues en 2 étapes : Une étape d'optimisation simultanée de la structure des règles et des paramètres correspondants. La deuxième étape est un apprentissage plus fin des paramètres, initialement, optimisés dans la première phase.

L'algorithme d'optimisation, caractérisé par un AGM, représente les caractéristiques suivantes :

- *Taille réduite des chromosomes* : La stratégie du codage mixte et le mode de partitionnement de l'espace d'entrée permettent de rendre minimale la représentation chromosomiale.
- *Concept de changement d'échelles* : L'analyse des objectifs de modélisation et la notion de *compétitions imposées* de ces objectifs en sont une façon de changer d'une manière dynamique l'échelle de la fonction d'évaluation de l'algorithme d'optimisation. Ce mécanisme assure une diversité des solutions dans la population de l'AGM.
- *Types des mutations appliquées* : non linéaires & uniformément forcée.

Il est à souligner que selon le type des variables d'entrées et de la loi de commande synthétisée, on peut avoir des similarités avec les contrôleurs conventionnels du type PID. Les non linéarités des gains des actions de proportionnalité, de dérivation et d'intégrale intervenant dans la construction de la loi de commande font la différence entre le PID conventionnel et le contrôleur ainsi développé.

Dans le but de tester et d'évaluer les performances de l'algorithme proposé pour la synthèse d'une loi de commande à objectifs multiples (simple, précis, stable et robuste), nous avons effectués des simulations sur quelques systèmes non linéaires. A chaque type de système non linéaire est appliquée une forme spécifique de la loi de commande développée. Cette différence est due, principalement, à la nature et à la complexité du procédé à commander. Le contrôle de la température d'un bain d'eau, contrôle du pendule inversé et la commande d'un robot manipulateur à 2 degrés de liberté sont des exemples de test.

La deuxième contribution de cette étude consiste à développer une méthodologie de synthèse permettant la prise en compte explicite de plusieurs spécifications structurelles et fonctionnelles, simultanément. Le concept du multimodèle sera exploité pour la synthèse d'une loi de commande floue tout en assurant la stabilité en BF. L'idée consiste à déterminer un ensemble réduit de règles avec la définition des zones de fonctionnement (fonction d'appartenance) pour chaque variable d'état du système. Autour de ces zones sera établie le multimodèle par la linéarisation du modèle du système. Cette phase représente la modélisation floue du système non linéaire. Le comportement global résulte de la « fusion » par le formalisme de TS de l'ensemble des comportements locaux. Les AG

représentent l'outil d'optimisation intervenant dans toutes les étapes de synthèse de la loi de commande. L'algorithme développé dans cette contribution consiste en :

- *Simulation du comportement du procédé à étudier*. Cette opération nous permet de déterminer, approximativement, la plage de variation des variables d'état décrivant le comportement dynamique du système.
- *Initialisation de la base de règles*. Cette étape ne concerne que la partie commande par retour d'état du type PDC. Dans cette phase, l'algorithme d'optimisation développé dans la première contribution avec quelques modifications permet de trouver :
  - Un nombre minimal de règles floues assurant un degré de l'efficacité de la loi de commande par retour d'état.
  - Définition des fonctions d'appartenance (zones de fonctionnement) intervenant dans la détermination des points de fonctionnement.
- *Modélisation floue du procédé*. La procédure consiste en la construction des modèles locaux du système par linéarisation de modèle du système autour des points de fonctionnement prédéterminés. Ces points sont déterminés à partir de la partie prémisse qui est commune entre le contrôleur et le modèle. Les centres des fonctions d'appartenance (zones de fonctionnement) correspondants aux définitions des variables d'état représentent les points de fonctionnement autour duquel la linéarisation sera établie.
- *Etablissement des conditions de stabilité du modèle complet*. L'approche génétique est adaptée dans cette étape. La méthode LMI est remplacée par une exploration génétique de l'espace de recherche des paramètres justifiant les conditions de stabilité.

La méthode proposée permet d'établir la synthèse d'une loi de commande et de donner des conditions suffisantes de stabilité du modèle en BF. Le nombre minimal de règles (resp. le nombre des modèles locaux capables de représenter, convenablement, le système non linéaire) est l'avantage de l'algorithme développé. Pour illustrer les performances de l'algorithme proposé pour la synthèse d'une loi de commande floue tout en assurant la stabilité en BF, nous considérons la commande en poursuite d'un bioprocédé de traitement des eaux usées dans l'industrie du papier.

E – Organisation du travail

Ce mémoire, décomposé en 4 chapitres, est organisé de la façon suivante :

Un premier chapitre, consacré intégralement aux notions et aux outils utilisés le long de ce travail, met l'accent sur les principales notions théoriques et pratiques des techniques intelligentes : La LF, les RNA et les AG. L'objectif est de fournir un cadre pour les recherches menées dans cette thèse. Ce chapitre, de nature théorique, porte tout d'abord sur l'étude approfondie des caractéristiques des TIA dans le but d'analyser leur capacité d'approximation et d'optimisation des fonctions. Après avoir introduit des concepts de base concernant la structure générale et les différents types des modèles flous, nous avons étudié plus, particulièrement, les caractéristiques structurelles et paramétriques des ces modèles. Par la suite, nous présentons les caractéristiques fondamentales des RNA. La topologie du réseau, les fonctions d'activation des neurones, le seuil de ces fonctions et l'algorithme d'apprentissage utilisé sont des facteurs significatifs dans l'emploi des modèles connexionnistes. Nous avons aussi exposé, en détail, les différentes étapes nécessaires à l'exécution d'un AG : Configuration paramétriques, initialisation de la population, évaluation, sélection pour la reproduction, test d'arrêt. Les améliorations de l'AG de base, pour l'aider à se converger vers une solution optimale, ont, elles – aussi, été

évoquées. Pour chaque technique, nous avons exposé un état de l'art des propositions pour l'amélioration des performances.

Par contre, le deuxième chapitre est dédié aux modes d'hybridation SIF-AG, SIF-RNA et SIF-RNA-AG. Un état de l'art bien détaillé des méthodes de modélisation des SIF est effectué. Nous avons commencé par les différentes techniques d'optimisation des SIF par les AG. Ensuite, l'accent est mis sur les différentes structures de réseaux neuronaux - flous «RNF», développées au cours de ces dernières années, alliés avec leurs algorithmes d'apprentissage.

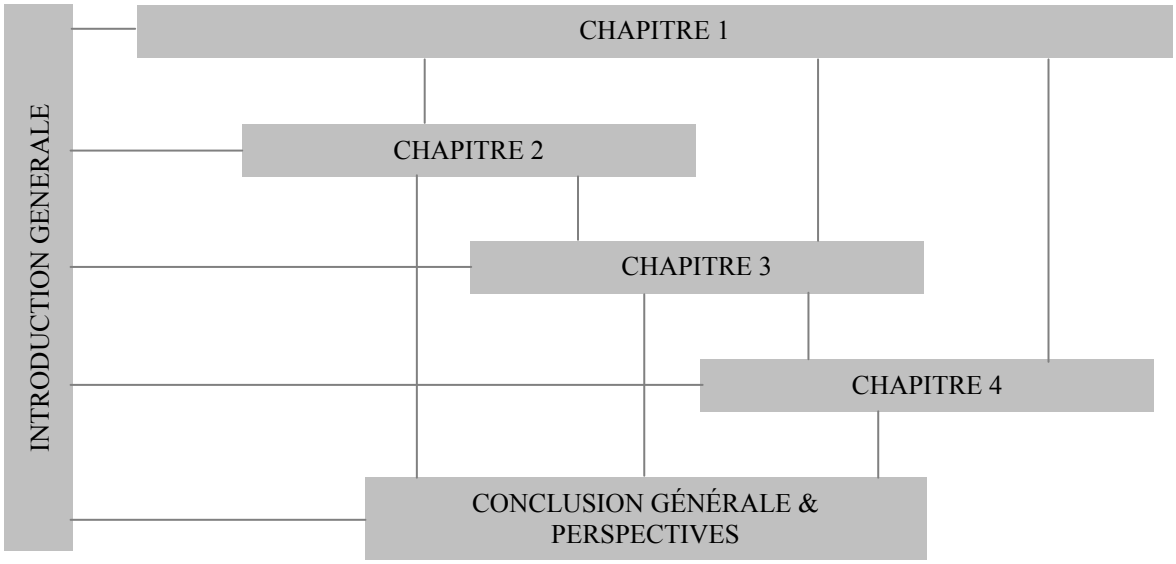
Ensuite, une partie du fruit de cette étude est couronnée par le troisième chapitre. Ce dernier est consacré à la description d'une méthodologie de synthèse d'une loi de commande simple, efficace, stable et robuste pour le contrôle des systèmes, fortement, non linéaires et complexes. La stratégie de commande proposée dans ce chapitre repose sur le concept d'hybridation neuronale - floue. Un réseau RBF-flou étendu au raisonnement approximatif du type Takagi-Sugeno d'ordre 1 est utilisé. Un algorithme hybride est employé pour une optimisation structurelle et paramétrique du contrôleur proposé. La simplicité de la loi de commande, l'efficacité, la stabilité et la robustesse sont des facteurs intervenant dans le critère des performances à optimiser. Les détails de la mise en œuvre et des exemples d'application (la température d'un bain, le pendule inversé et le robot manipulateur à 2 degrés de liberté) sont utilisés pour justifier la validité de l'approche proposée.

Enfin, l'essence de la thèse se concrétise dans le quatrième chapitre. Il est dédié à l'étude de l'approche multimodèle qui permet de représenter un système dynamique non linéaire comme une combinaison d'un ensemble de modèles linéaires conçus dans des zones de fonctionnement. Les modèles de TS sont bien adaptés à ce type de représentation. La synthèse d'un SIF par l'analyse de la stabilité de systèmes dynamiques en BF, utilisant le concept de multimodèle a été adaptée et considérée comme outil de conception. Une stratégie de synthèse a été développée dans ce chapitre. Une exploration génétique de l'espace de recherche structurelle et paramétrique a été exploitée pour la synthèse d'une loi de commande floue assurant la précision, l'efficacité, la simplicité et la stabilité. Pour illustrer la validité de l'algorithme proposé, un exemple de systèmes non linéaires du type MIMO est examiné. Les résultats de simulations justifient la capacité de l'algorithme développé de réaliser les objectifs visés par le concepteur.

Il est à souligner que tous les algorithmes proposés dans cette thèse, aussi modeste qu'elle soit, *assurent la stabilité et la robustesse des structures de commande par rapport aux perturbations et aux erreurs d'approximations.*

Finalement, et pour clôturer ce mémoire, nous présentons quelques conclusions tout en synthétisant les différentes contributions et en discutant des perspectives et propositions envisagées pour poursuivre cette recherche.

Les liens entre les différents chapitres constituant cette thèse sera représentée par la figure arborescente suivante :



F – Travaux scientifiques réalisés

- Publication internationale avec comité de lecture international

[1]. **A. Soukkou, A. Khellaf, S. Leulmi**, "Multiobjective Optimization of a Fuzzy PID controller"*, Archives of Control Sciences* (ACS), Vol. 16(LII), No. 4, 2006, pp. 445-461.

PL ISSN: 0004-072X

http://157.158.12.1/ACS/index.php

[2]. **A. Soukkou, A. Khellaf, S. Leulmi**, "Systematic Design Procedure of TS-type Fuzzy Controllers"*, International Journal of Computational Intelligence and Applications (IJCIA),* Vol. 6, No. 4, December 2006, pp. 531-549. (**Word Scientific Publisher**)

ISSN: **1469-0268**.

http://www.worldscinet.com/ijcia/ijcia.shtml

Abstracting/ indexing

- CompuScience.
- INSPEC (IET).

[3]. **A. Soukkou, S. Leulmi, A. Khellaf**, "Systematic Design and Hybrid Learning of Robust Fuzzy Neural Network Network Controller with Reduced Rule Base"*, International Journal of Hybrid and Intelligent Systems (IJHIS)*, Vol. 4, No. 2, June 2007, pp. 63-88. (**IOS Press Publisher**)

ISSN: **1448-5869**.

http://ijhis.hybridsystem.com/

Abstracting/ indexing

- CompuScience.
- Computer & Communications Security.
- Abstracts Database.
- SCOPUS.
- ACM Computing Reviews.
- Zentralblatt Math.
- ACM Guide to Computing Literature.

- DLBP.
- COMPEDEX PLUS.
- COMPUTER ABSTRACTS.
- EBSCO's Database.

[4]. **A. Soukkou, A. Khellaf, S. Leulmi**, "Multiobjective optimization of Robust Takagi-Sugeno Fuzzy Neural Controller with Hybrid Learning Algorithm", *International Journal of Modelling, Identification and Control (IJMIC)*, Vol. 2, No. 4, 2007, pp. 332-346. (**Inderscience Publisher – Suisse**).

ISSN (Online): 1746-6180 - ISSN (Print): 1746-6172

http://www.inderscience.com/ijmic/

Abstracting/ indexing

- Compendex.
- Computer and Information Systems.
- Electronics & Communications Abstracts.
- Google Scholar.
- Inspec.
- Pascal.
- Scopus.

[5]. **A. Soukkou, S. Leulmi, A. Khellaf,** "How to Optimize the TS-Fuzzy Knowledge Base to Achieve a Desired Performances: Accuracy and Robustness", *International Journal of Optimal Control Applications and Methods (OCAM)*, Vol. 29, No. 1, 2008, pp. 19-40. (**Interscience - Wiley Publisher**).

Online ISSN : 1099-1514
Print ISSN : 0143-2087

http://www3.interscience.wiley.com/journal/2133/home

Abstracting/ indexing

- Cambridge Scientific Abstracts (CSA/CIG).
- COMPENDEX (Elsevier).
- CompuMath Citation Index® (Thomson ISI).
- CSA Technology Research Database (CSA/CIG).
- Current Contents®/Engineering, Computing & Technology (Thomson ISI).
- Current Index to Statistics (ASA/IMS).
- INSPEC (IET).
- Journal Citation Reports/Science Edition (Thomson ISI).
- Mathematical Reviews/MathSciNet/Current Mathematical Publications (AMS).
- Science Citation Index Expanded™ (Thomson ISI).
- SCOPUS (Elsevier).
- Statistical Theory & Method Abstracts (International Statistical Institute).
- Web of Science® (Thomson ISI).
- Zentralblatt MATH/Mathematics Abstracts (FIZ Karlsruhe).

[6]. **A. Soukkou, S. Leulmi, A. Khellaf, M. Grimes,** "Control of Dynamical Systems: An Intelligent Approach", *International Journal of Control, Automation, and Systems (IJCAS),* Vol. 6, No. 4, August 2008, pp. 583-595.

ISSN: 1598-6446

http://www.ijcas.org/

Abstracting/ indexing

- EI Compendex.
- INSPEC (IET).

[7]. **Soukkou, A. Khellaf, S. Leulmi,** "Reduced Rule Base + Hybrid Learning = Robust and Optimal Fuzzy Controller", *International Applied Soft Computing*, Elsevier, (Revised form October 2007). Revised Paper has been submitted, Elsevier Edition.

ISSN: 1568-4946

Abstracting/ indexing

- CompuMath Citation Index.
- Current Contents/Engineering, Computing & Technology.
- SCI Expanded.
- Scopus.

[8]. **Soukkou, S. Leulmi, A. Khellaf**, **K. Boudeghdegh**, "Optimal Control of a CSTR Process", *Brazilian Journal of Chemical Engineering*, Vol. 25, No. 04, October-December 2008, pp. 809-822.

ISSN: 0104-6632

Abstracting/ indexing

- Chemical Abstracts.
- Engineering Index.
- Information Access Co.
- Chemical Abstracts Service.
- Engineering Index.
- Scientific Eletronic Library Online.
- Science Citation Index Expanded (SciSearch ®).
- ISI Alerting Services^{sm}.
- Current Contents® /Engineering, Computing, and Technology.
- International Nuclear Information System.
- Ulrich's Periodicals Directory™.
- All-Russian Institute of Scientific and Technical Information.

- Publication nationale avec comité de lecture international

[9]. **A. Soukkou, A. Khellaf, S. Leulmi**, "Supervision Neuro-Floue a Apprentissage Génétique d'un PID Robuste"*, Revue Sciences et Technologies*, Université Mentouri, Constantine, No. 23, Juin 2005, pp. 95-106.

ISSN: **1111-5041**.

- Communications internationales et nationales avec comité de lecture

[10]. **A. Soukkou, A. Khellaf and S. Leulmi**, "Control of Overhead Crane by Fuzzy-PID with Genetic Optimization", *Artificial Intelligence Applications and Innovations: IFIP 18th World Computer Congress TC12 First International Conference on Artificial Intelligence Applications and Innovations,* (AIAI-2004), 22–27 August 2004 Toulouse, France, pp. 67-80.

Editors = Max Bramer & Devedzic, Vladan ;
Publisher Info = Boston, Mass.; London: Kluwer Academic Publishers, c2004;
Year = {2004};

ISBN: 1-4020-8150-2

[11]. **A. Soukkou, S. Leulmi, A. Khellaf**, "Global optimization of a fuzzy knowledge using genetic algorithms"*, First Baha Technical Meeting (BTM'2004),* Mai 3-6, 2004, Saudi Arabic, pp. 370-379.

[12]. **A. Soukkou, A. Khellaf, S. Leulmi**, "Commande floue robuste à apprentissage génétique d'un pendule inversé"*, 2eme séminaire sur les systèmes de détection : Architecture et Technologie, DAT'04*, 30-1, Juin 2004, CFDAT, Alger, pp. 228-233.

[13]. **A. Soukkou, A. Khellaf, S. Leulmi**, "Genetic training of a fuzzy PID"*, International Conference on Modelling and Simulation –ICMS'04-* Valladolid (Spain), 22 - 24 September 2004.

[14]. **A. Soukkou, A. Khellaf, S. Leulmi**, "Genetic optimization of a neural controller"*, AMSE International Conference on Modelling and Simulation,* Lyon- France, July 2004.

[15]. **A. Soukkou, A. Khellaf, S. Leulmi**, "Commande neuronale à apprentissage génétique d'un bras manipulateur à 02 DOF"*, Conférence sur le Génie Electrique CGE'04*, Ecole Militaire Polytechnique, Alger, 12-13 Avril 2005.

Chapitre 1

Outils de Conception : Structurels et Fonctionnels

Résumé :

L'objectif de ce chapitre est de fournir un cadre pour les travaux menés dans cette thèse, qui met l'accent sur les principales notions théoriques et pratiques des techniques intelligentes : la logique floue, les réseaux de neurones et les algorithmes génétiques.

« Le vrai danger, ce n'est pas quand les ordinateurs
penseront comme les hommes,
c'est quand les hommes
penseront comme
les ordinateurs. »

Sydney J. Harris

Sommaire

1. Outils de conception : Structurels & fonctionnels

1. Introduction

Le *soft computing* a été introduit par L. A. Zadeh en 1994 comme *un moyen de construire des systèmes intelligents répondant à des obligations d'efficacité, de robustesse, de facilité d'implémentation et d'optimisation de coûts temporels, énergétiques, financiers, …etc., tout en prenant en considération la composante humaine, généralement, présente dans les systèmes* [1].

Ses principales composantes dites aussi *techniques intelligentes* en sont la logique floue, inspirée du processus de raisonnement approximatif des êtres humains, les réseaux neuronaux artificiels et des méthodes d'optimisation telles que les algorithmes génétiques.

- La logique floue part, essentiellement, de la notion de variable linguistique. Ce type de variables sert à modéliser des connaissances imprécises ou incomplètes.
- Les réseaux de neurones artificiels se basent sur l'apprentissage, en simulant le raisonnement humain.
- Les AG sont des algorithmes stochastiques fondés sur les mécanismes de l'évolution génétique des espèces, plus précisément, du principe de la sélection naturelle. Ils sont destinés à la recherche optimale des solutions.

Depuis le début des années 1990, ces techniques intelligentes font leur entrée dans les sciences de l'ingénieur. Le but visé par les chercheurs est de concevoir des systèmes artificiels qui retiennent les mécanismes importants des systèmes naturels.

Le reste du chapitre est consacré à la description des éléments de base de la théorie des systèmes flous, des réseaux de neurones artificiels et les algorithmes génétiques. Ces 3 concepts constituent la plateforme pour les différents travaux exposés dans cette thèse.

1.1. Systèmes d'inférence Floue

1.1.1. Introduction

La théorie de la logique floue est apparue dans les années 40 avec les premières approches du concept d'incertitude. Son intérêt réside dans sa capacité de traitement des données imprécises, incertaines et vagues. Elle est issue de la capacité de décision et d'action de l'être humain de façon pertinente malgré l'incertitude et l'imprécision des connaissances disponibles. Son but est de réaliser des systèmes artificiels effectuant des tâches, habituellement, prises en charge par l'être humain. Les lois de commande classiques sont remplacées par des règles linguistiques de type :

SI (Condition (s)) **ALORS** (Action (s)).

La logique floue fournit un cadre formel pour construire des systèmes qui offrent à la fois une bonne performance numérique (précision) et une représentation linguistique (interprétabilité). D'un point de vue numérique, les SIF sont des systèmes non linéaires capables de traiter des informations imprécises et incomplètes. Linguistiquement, ils représentent les connaissances sous forme de règles floues. Ceci est une explication naturelle des processus décisionnels.

1.1.2. Ensembles flous

Les ensembles flous sont introduits par L. A. Zadeh comme une généralisation de la notion de l'ensemble ordinaire [2]. On a alors une transition graduelle et non stricte entre l'appartenance complète et la non appartenance. D'une manière formelle, un ensemble flou A est caractérisée par une fonction d'appartenance μ_A qui attribut un degré d'appartenance à tous les éléments dans l'univers de discours X :

$$\mu_A(x): X \to [0,1] \tag{1.1}$$

Il existe 4 caractéristiques essentielles caractérisant l'ensemble flou de l'ensemble booléen : Le type, le noyau, la hauteur et le support [19].

- **Les types :** Elles peuvent être triangulaire, gaussienne, trapézoïdale, sigmoïdale ou de type pic 'singleton', ...etc. La figure 1.1 représente les caractéristiques d'une fonction trapézoïdale.
- **Le noyau :** Le noyau est défini par :

$$\eta(A) = \{ x \in X \mid \mu_A(x) = 1 \} \tag{1.2}$$

C'est l'ensemble booléen de tous les éléments appartenant de façon absolue à l'ensemble flou A. Quand $\eta(A) = \upsilon \in]0,1]$, celui-ci est appelée *'valeur modale'* de A. Pour la fonction d'appartenance triangulaire, la valeur modale correspond à la valeur du sommet.

- **La hauteur** : C'est la valeur maximale de la fonction d'appartenance, généralement, égale à 1. Elle est définie par :

$$h(A) = \text{Max}_{x \in X} \{ \mu_A(x) \} \tag{1.3}$$

- **Le support :** C'est l'ensemble des éléments de X qui appartiennent au moins un peu à A. Il est défini par :

$$s(A) = \{ x \in X \mid \mu_A(x) > 0 \} \tag{1.4}$$

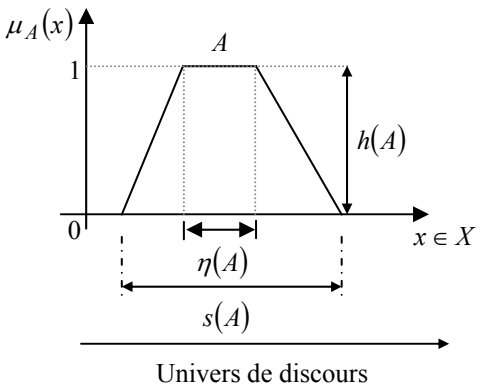


Fig. 1.1 : Caractéristiques d'un sous ensemble flou A.

1.1.2.1. Partition floue

On dit qu'un nombre de sous ensembles flous $N\ \{A_1, A_2, ..., A_N\}$ définis sur l'univers de discours (référentiel) X forment une partition floue si :

$$\forall x \in X, \quad \sum_{i=1}^{N} \mu_{A_i}(x) = 1 \tag{1.5}$$

L'univers de discours est dit aussi espace caractéristique ou espace de décision. L'un des intérêts essentiels de la notion d'ensemble flou réside dans la possibilité d'étendre les opérateurs ensemblistes usuels, tels que l'intersection, l'union et la complémentation. Si A et B sont 2 sous ensembles flous de Ω, on pose généralement :

$$\begin{aligned} \mu_{A\cap B}(w) &= \min(\mu_A(w), \mu_B(w)) \\ \mu_{A\cup B}(w) &= \max(\mu_A(w), \mu_B(w)) \\ \mu_{\overline{A}}(w) &= 1 - \mu_A(w) \end{aligned} \tag{1.6}$$

Pour tout $w \in \Omega$, $\overline{A}$ désignant le complémentaire de A dans Ω.

L'extension (1.6), initialement proposé par Zadeh [2], n'est cependant pas unique. Plus généralement, posons :

$$\mu_{A\cap B}(w) = \mathrm{T}(\mu_A(w), \mu_B(w)) \tag{1.7}$$

$$\mu_{A\cup B}(w) = \perp(\mu_A(w), \mu_B(w)) \tag{1.8}$$

où $\mathrm{T} : [0,1]\times[0,1] \to [0,1]$ correspond à la norme triangulaire ou *t-norme* et $\perp : [0,1]\times[0,1] \to [0,1]$ est une conorme triangulaire ou *t-conorme*.

1.1.2.2. Intersection

A et B sont 2 sous ensembles flous définis dans le référentiel X avec des degrés d'appartenance $\mu_A(x)$ et $\mu_B(x)$, respectivement. L'intersection entre A et B est aussi un sous ensemble flou défini par :

$$A \cap B = \{ (x, \mu_{A\cap B}(x)) \mid x \in X \} \tag{1.9}$$

où le degré d'appartenance $\mu_{A\cap B}(x)$ est défini par :

$$\mu_{A\cap B}(x) = \mu_A(x) \tilde{\wedge} \mu_B(x) \tag{1.10}$$

Le symbole $\tilde{\wedge}$ est choisi pour représenter la conjonction ET-flou des fonctions d'appartenance. Les opérateurs d'intersection les plus utilisés sont donnés par le tableau 1.1(a).

1.1.2.3. Union

L'union entre A et B est donnée par :

$$A \cup B = \left\{ \left(x, \mu_{A\cup B}(x)\right) \middle| \; x \in X \right\} \tag{1.11}$$

où le degré d'appartenance $\mu_{A\cup B}(x)$ est défini par :

$$\mu_{A\cup B}(x) = \mu_A(x) \tilde{\vee} \mu_B(x) \tag{1.12}$$

Le symbole $\tilde{\vee}$ est choisi pour représenter la disjonction OU-flou des fonctions d'appartenance. Les opérateurs d'union les plus utilisés sont donnés par le tableau 1.1(b).

Tab. 1.1 : Différentes relations *t-norme* et *t-conorme*.

Type	*t-norme*
Min	$\mu^{m}_{A\cap B}(x) = \min(\mu_A(x), \mu_B(x))$
Produit algébrique	$\mu^{a}_{A\cap B}(x) = \mu_A(x).\mu_B(x)$
Différence bornée	$\mu^{b}_{A\cap B}(x) = \max(0, \mu_A(x) + \mu_B(x) - 1)$
Produit d'Einstein	$\mu^{E}_{A\cap B}(x) = \dfrac{\mu_A(x).\mu_B(x)}{2 - (\mu_A(x) + \mu_B(x) - \mu_A(x).\mu_B(x))}$
Produit d'Hamasher	$\mu^{H}_{A\cap B}(x) = \dfrac{\mu_A(x).\mu_B(x)}{\mu_A(x) + \mu_B(x) - \mu_A(x).\mu_B(x)}$
(a)	
Type	*t-conorme*
Max	$\mu^{m}_{A\cup B}(x) = \max(\mu_A(x), \mu_B(x))$
Somme algébrique	$\mu^{a}_{A\cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x).\mu_B(x)$
Somme bornée	$\mu^{b}_{A\cup B}(x) = \min(1, \mu_A(x) + \mu_B(x))$
Somme d'Einstein	$\mu^{E}_{A\cup B}(x) = \dfrac{\mu_A(x) + \mu_B(x)}{1 + \mu_A(x).\mu_B(x)}$
Somme d'Hamasher	$\mu^{H}_{A\cap B}(x) = \dfrac{\mu_A(x) + \mu_B(x) - 2\mu_A(x).\mu_B(x)}{1 - \mu_A(x).\mu_B(x)}$
(b)	

Dans [20], les auteurs utilisent l'intersection '*t-norme*' de Yager définie par la fonction :

$$\mu^{Y}_{A\cap B}(x) = 1 - \min\left(1, \left((1-\mu_A(x))^{w} + (1-\mu_B(x))^{w}\right)^{\frac{1}{w}}\right) \tag{1.13}$$

avec $w \in [0, +\infty[$. Une méthode d'optimisation est adoptée pour le choix d'une meilleure stratégie de *t-norme*. Il est remarquable que si :

$$
\begin{aligned}
&w \to \infty, \quad \mu^{Y}_{A\cap B}(x) = \mu^{m}_{A\cap B}(x) \\
&w \to 0, \quad \mu^{Y}_{A\cap B}(x) = \begin{cases} \min(\mu_A(x), \mu_B(x)) & \text{si } \max(\mu_A(x), \mu_B(x)) = 1 \\ 0 & \text{autrement} \end{cases} \\
&w = 1, \quad \mu^{Y}_{A\cap B}(x) = \mu^{b}_{A\cap B}(x)
\end{aligned} \tag{1.14}
$$

L'univers de discours permet de décrire le domaine de variation de ce qu'on appelle *variable floue*. L'appellation associée à chaque sous-ensemble flou constitue ce qu'on appelle *variable linguistique*. Une variable linguistique est représentée par un triplet : $\{V_{arx}, X, T_{V_{arx}}\}$. V_{arx} est la variable elle-même, X l'univers de discours et $T_{V_{arx}}$ est l'ensemble des caractérisations floues de la variable V_{arx}.

Exemple 1.1 :

Considérons la variable *taille* définie sur l'ensemble des réels $\Re^+$ et caractérisée par les ensembles flous (labels linguistiques) $\{P,\ M,\ G\}$ où ***P*** désigne **P**etite, ***M*** est l'abréviation de **M**oyenne et ***G*** caractérise la notion **G**rande, respectivement. La variable *taille* est, alors, représentée par le triplet suivant : $\{taille,\ \Re^+,\ \{P,\ M,\ G\}\}$.

Les codes ***P***, ***M*** et ***G*** représentent les variables linguistiques (les fonctions d'appartenance) qui décrivent l'état ou le comportement du phénomène (ou du procédé) à étudier. D'après la figure 1.2, une personne mesurant 136 cm est de taille **P**etite avec un degré d'appartenance 0.4 et de taille **M**oyenne avec un degré d'appartenance 0.6.

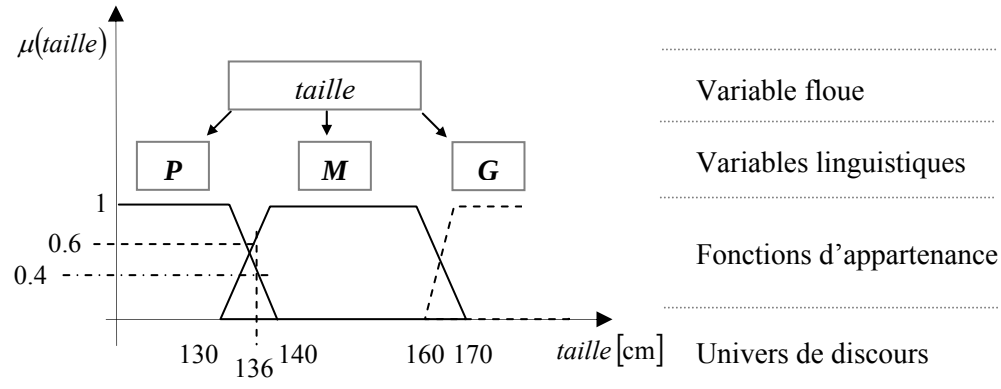
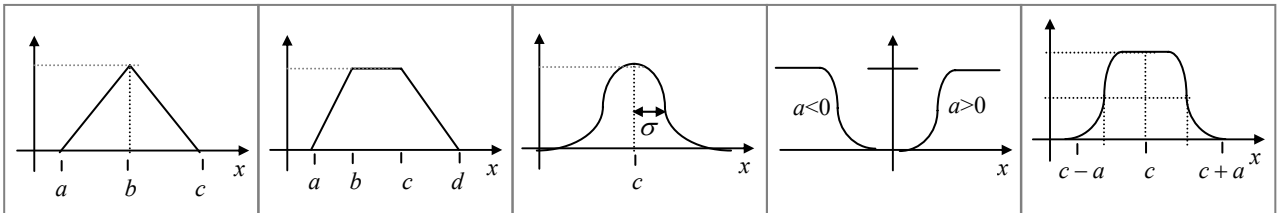


Fig. 1.2 : Forme des fonctions d'appartenance de la variable *taille*.

Les fonctions d'appartenance peuvent avoir différentes formes, mais il est judicieux de choisir des fonctions convexes de sorte qu'il existe au moins un point de degré maximal et tel que le degré décroît, lorsqu'on s'éloigne de ce maximum. Les modèles/formes des fonctions d'appartenance les plus, couramment, utilisées sont données par le tableau 1.2 et la figure 1.3.

Tab. 1.2 : Modèles des fonctions d'appartenance les plus utilisées.

Fonction	Modèle mathématique
Triangulaire (Fig. 1.3a)	$F(x;a,b,c)=\max\left(\min\left(\frac{x-a}{b-a},\frac{c-x}{c-b}\right),0\right)$
Trapézoïdale (Fig. 1.3b)	$F(x;a,b,c,d)=\max\left(\min\left(\frac{x-a}{b-a},1,\frac{d-x}{d-c}\right),0\right)$
Gaussienne (Fig. 1.3c)	$F(x;\delta,c)=\exp\left(-\left(\frac{x-c}{\sigma}\right)^2\right)$
Sigmoïde (Fig. 1.3d)	$F(x;a,c)=\frac{1}{1+\exp(-a(x-c))}$
Cloche (Fig. 1.3e)	$F(x;a,b,c)=\frac{1}{1+\left\lvert\frac{x-c}{a}\right\rvert^{2b}}$



(a)	(b)	(c)	(d)	(e)
(a) : Triangulaire.	**(b) :** Trapézoïdale.	**(c) :** Gaussienne.	**(d) :** Sigmoïde.	**(e) :** Cloche.

Fig. 1.3 : Formes des fonctions d'appartenance les plus utilisées.

1.1.3. Inférences floues

Les relations entre les variables floues sont représentées par des règles de la forme:

$$\underline{\mathrm{R}}:\mathrm{SI}\underbrace{(\mathrm{Condition(s)})}_{\text{Antécédents}}\mathrm{ALORS}\underbrace{(\mathrm{Action(s)})}_{\text{Conséquences}} \tag{1.15}$$

Généralement, les antécédents (prémisses) se composent de propositions de la forme '' x est A '', où x et A sont la variable et son terme 'label' linguistique, respectivement. Les conséquences sont aussi composées de propositions de la forme '' y est B '', où y et une variable et B l'un de ses termes linguistiques. Il est possible de tirer des conclusions à partir d'une règle floue en appliquant le concept de l'inférence floue.

Le raisonnement logique est le processus qui combine des propositions en utilisant les opérations de conjonction, disjonction et d'implication.

Soient $A_1, A_2, ..., A_n$ des sous ensembles flous définis, respectivement, dans les univers de discours $U_1, U_2, ..., U_n$. Le produit cartésien des $A_1, A_2, ..., A_n$ est un sous ensemble flou, $A = A_1 \times A_2 \times ... \times A_n$ dans $U_1 \times U_2 \times ... \times U_n$ de fonction d'appartenance :

$$\mu_A(x) = \min\left\{ \mu_{A_1}(x_1), \ ..., \mu_{A_n}(x_n) \right\} \qquad \forall x = (x_1, ..., x_n) \in X \tag{1.16}$$

Soit R et S 2 relations floues définies, respectivement, dans $U \times V$ et $V \times W$. La composition flou *max-min* $R \circ V$ est une relation floue dans $U \times W$ de fonction d'appartenance [19].

$$\mu_{R \circ V}(u, w) = \sup_{v \in V} \left(\min\left(\mu_R(u, v), \mu_S(v, w) \right) \right) \tag{1.17}$$

avec sup sélectionne la valeur maximale (supremum) d'un ensemble des éléments. La composition est formée de 2 paires d'éléments dans $U \times W$ avec leurs degrés d'appartenance calculés comme étant le maximum des valeurs minimales de la composition de différents éléments lorsqu'ils sont combinés avec tous les éléments de V. La composition de la règle d'inférence est définie comme suit [21] :

Soit R une relation floue définie dans $U \times V$ et $\widetilde{A}$ un ensemble flou de U, alors l'ensemble flou $\widetilde{B}$ de V est déduit de la composition :

$$\widetilde{B} = \widetilde{A} \circ R \tag{1.18}$$

où

$$\mu_{\widetilde{B}}(y) = \sup_{x \in \widetilde{A}} \left(\min\left(\mu_{\widetilde{A}}(x), \mu_R(x, y) \right) \right) \tag{1.19}$$

Cette stratégie de raisonnement peut être décrite en terme de MPG :

$$\begin{array}{ll} \text{Prémisse 1} & \left(x \text{ est } \widetilde{A}\right) \\ \text{Prémisse 2} & \text{SI } \left(x \text{ est } A\right) \text{ ALORS } \left(y \text{ est } B\right) \\ \hline \text{Conclusion} & \left(y \text{ est } \widetilde{B}\right) \end{array} \tag{1.20}$$

où la règle (prémisse 2) correspond à la relation floue R, définie avec un degré d'appartenance qui indique le degré de vérité de l'implication $A \Rightarrow B$.

1.1.4. Système de contrôle flou

D'un point de vue théorique, un modèle flou ou un SIF, peut être utilisé à la fois pour l'identification des fonctions (approximateur universel) et comme système de contrôle non linéaire. D'un point de vue pratique, la première application de la logique floue dans le domaine industriel a été effectuée en 1974 par Mamdani et son équipe [22]. De nombreux travaux de recherches et d'applications ont été développés dans les 2 dernières décennies [19] & [23] - [25].

Les systèmes flous sont des systèmes faciles à comprendre, simple à mettre en oeuvre et peu coûteux à se développer. Ces systèmes émulent les stratégies de contrôle humain. Ils sont accessibles même par ceux qui n'ont aucune formation de base en contrôle.

Un contrôleur flou est décrit par un ensemble de règles de type **SI** (condition (s)) **ALORS** (action (s)) permettant de convertir la stratégie de contrôle linguistique acquise auprès d'un expert humain en une stratégie de contrôle automatique bien adaptée au monde réel. Dans cette situation, l'opérateur humain peut être remplacé par une combinaison d'un ensemble de règles floues, comme le montre la figure 1.4 [26].

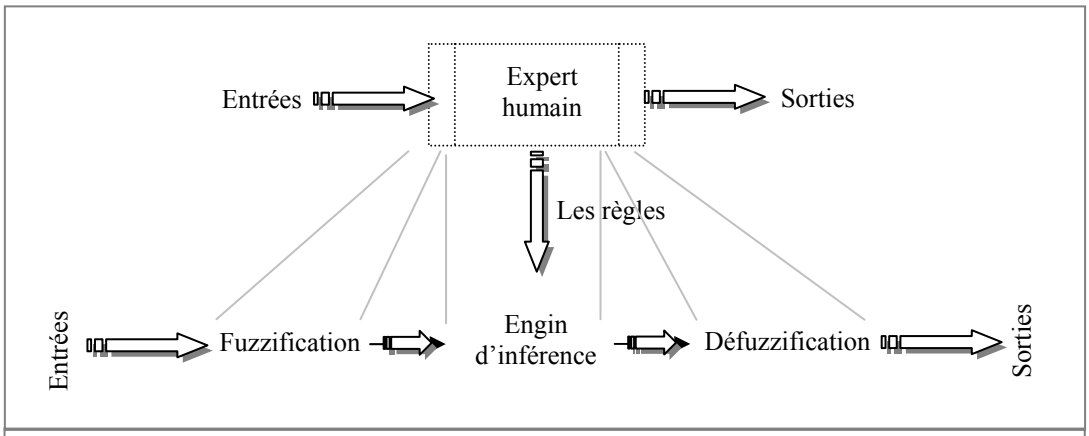


Fig. 1.4 : Définition conceptuelle d'un système de contrôle flou.

Le schéma synoptique général d'un contrôleur flou est représenté par la figure 1.5. Quelque soit le type d'application du contrôleur, on retrouve généralement la même configuration qui est composée de 4 blocs essentiels, à savoir :

- Une interface de fuzzification en entrée.
- Une base de connaissances (base de règles & engin d'inférence).
- L'engin d'inférence ou la logique de prise de décision.
- Une interface de défuzzification à la sortie.

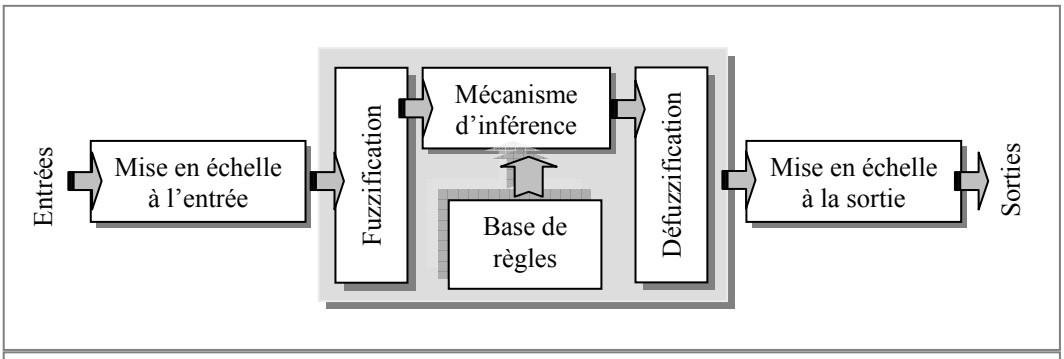


Fig. 1.5 : Structure de base d'un système de contrôle flou.

La mise en échelle (normalisation/dénormalisation) des grandeurs d'entrées/sorties permet d'adapter le traitement des signaux d'entrées/sorties. Par convention, la plage de variation des variables d'entrées/sorties est comprise entre –1 et +1. Les opérations de normalisation et de dénormalisation sont optionnelles.

La fonction de *la base de règles (base de connaissances floues)* est de présenter, de manière structurée, la stratégie (*politique*) de contrôle sous la forme d'un ensemble de règles de type :

SI (Etat (s) du processus) **ALORS** (Action (s) de contrôle)

Elle regroupe :

- La définition des règles d'inférence.
- Les ensembles flous associés aux variables d'entrées/sorties du contrôleur ainsi que leurs formes (triangulaire, gaussienne, trapézoïdale, ...etc.).

Prenons l'exemple d'une règle floue pour un système à 2 entrées x_1, x_2 et une sortie y (Fig. 1.6) de la forme :

$$\textbf{"SI}\,(x_1\ \textbf{est}\ A_1)\,\textbf{ET}\,(x_2\ \textbf{est}\ A_2)\,\textbf{ALORS}\,(y\ \textbf{est}\ B)\textbf{"}$$

où A_1 et A_2 sont des labels linguistiques associés aux variables d'entrées. Selon la structure particulière de la conséquence B, on peut distinguer 3 types de modèles flous basés sur des règles :

- *Modèle flou linguistique* (ou modèle de Mamdani) : La prémisse et la conséquence de la règle sont toutes les 2 des propositions floues qui utilisent des labels linguistiques (règles à conclusion symboliques) [22].
- *Modèle flou de Takagi-Sugeno* : La conséquence de la règle B est une valeur numérique constante (singleton) ou une équation mathématique bien précise. D'une manière plus générale, la partie conséquence d'une règle est une fonction polynomiale ou une équation différentielle dépendant des variables associées à la partie prémisse de la règle [5] (règles à conclusions algébriques).
- *Le modèle flou de Tsukamoto* [27] : Ces règles sont un cas particulier des règles de type Mamdani pour lesquelles les sous ensembles flous utilisés dans la partie des conséquences sont des fonctions d'appartenance monotones. L'emploi de ce modèle est très restreint [28] - [29].

La fuzzification : Elle permet d'assurer la conversion des grandeurs physiques d'entrées du SIF en variables linguistiques qui peuvent être traitées par les inférences.

Le mécanisme 'engin' d'inférence ou *logique de prise de décision* : Sa fonction est de calculer les valeurs finales des variables de sortie du contrôleur basées sur les contributions individuelles de chaque règle de *la base de règles*. Parmi les mécanismes d'inférence les plus utilisés, on note : *MAX-MIN, MAX-PROD* et *SOM-PROD*.

L'opération de défuzzification : Elle consiste à convertir le résultat de combinaisons des règles en une forme numérique bien adaptée au milieu du processus à commander. Plusieurs méthodes de défuzzification sont proposées dans la littérature [30]. La plus utilisée est celle du centre de gravité.

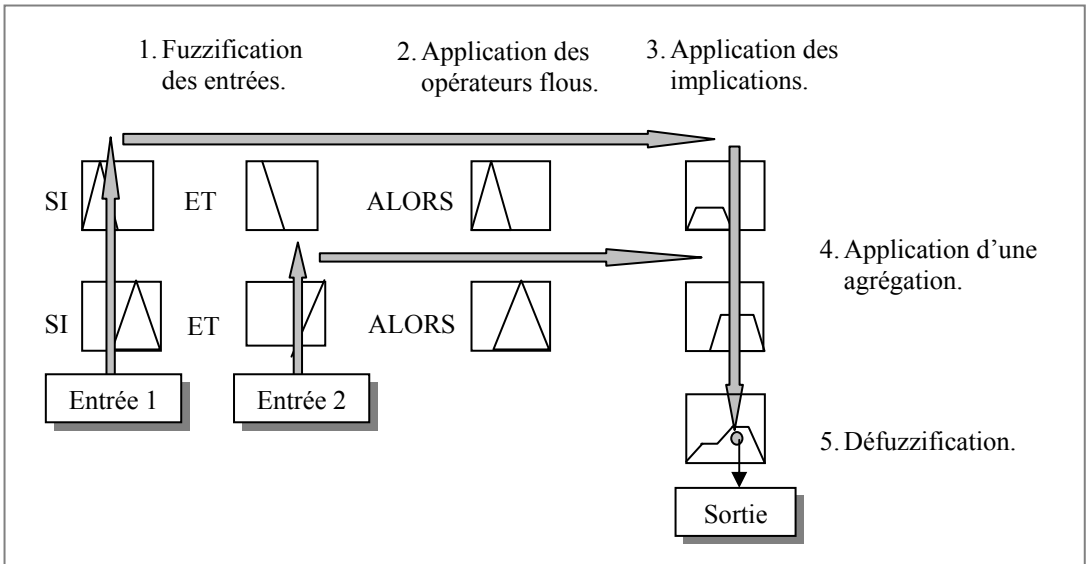


Fig. 1.6 : Digramme des inférences floues.

Généralement, 5 phases principales constituent le mécanisme de fonctionnement d'un SIF [31] :

- Fuzzification des variables d'entrées.
- Application des opérateurs flous (**ET** ou **OU**) dans la partie antécédent.
- Implication de l'antécédent à la conséquence.
- Agrégation des conséquences à travers les règles.
- Défuzzification à la sortie.

Les modèles de Mamdani reposent sur l'approche proposée par Zadeh. Ils permettent une description linguistique du système par une base de règles floues modélisant les relations entrées/sorties du système. L'inférence floue correspond aux 4 étapes suivantes :

- Calcul du degré d'appartenance de chaque entrée aux différents sous ensembles, $\mu_{X_j^i}$ pour $(j = 1, ..., n)$ et $(i = 1, ..., N)$.

- Calcul du degré d'activation de chaque règle, pour $(i = 1, ..., N)$, tel que :

$$\omega_i(x) = \min\left(\mu_{X_j^i}(x_j)\right) \qquad j = 1, ..., n \tag{1.21}$$

- Calcul de la contribution de chaque règle :

$$\mu_i(x) = \min\left(\omega_i(x), \mu_{B_i}(y)\right) \tag{1.22}$$

- Agrégation des règles :

$$\mu(y) = \max\left(\mu_i(y)\right) \tag{1.23}$$

Le résultat de l'agrégation est donc un sous ensemble flou caractérisé par sa fonction d'appartenance. Pour l'obtention d'une conclusion de nature précise y_0 à partir d'un ensemble flou résultant de l'agrégation, il faut utiliser une des méthodes de défuzzification.

Dans le modèle de TS, chacune des règles représente un modèle local sur une région floue d'entrée. Dans chaque région, le modèle flou est défini par la fonction :

$$F_i = \sum_{j=0}^{n} q_j^i \cdot x_j \tag{1.24}$$

Le modèle global est constitué par interpolation des modèles locaux. L'inférence floue dans ce modèle est composée de 3 étapes :

- Calcul du degré d'activation de chaque règle avec l'opérateur de conjonction implémenté par le produit arithmétique :

$$w_i(x) = \prod_{j=1}^{n} \mu_{X_j^i}(x_j) \qquad i = 1, ..., N \tag{1.25}$$

- Calcul des sorties individuelles : La sortie de la i^{eme} règle est donnée par sa fonction conséquente, c'est à dire :

$$y^i = F_i(x) \tag{1.26}$$

- Agrégation des sorties individuelles : La valeur finale de la sortie résultante de l'ensemble des règles est donnée par la moyenne des sorties individuelles pondérées par le degré d'activation des règles, soit :

$$y = \frac{\sum_{i=1}^{N} w_i(x).y^i}{\sum_{i=1}^{N} w_i(x)} \tag{1.27}$$

Les modèles de TS sont les plus utilisés pour extraire les connaissances à partir des données numériques. Ils correspondent en fait à une approche multimodèle dans laquelle le flou est utilisé pour agréger les sorties produites par chacun des modèles locaux [32] - [33].

En conclusion, les modèles de TS et les modèles de Mamdani sont 2 représentations complémentaires et non concurrentes. Le choix d'un modèle ou de l'autre dépend de l'application et du but visé.

1.1.5. Propriétés de la base de règles

Les paramètres d'un SIF peuvent être classés en 4 catégories présentées ci-dessous [34]. Le tableau 1.3 résume cette classification.

Tab. 1.3 : Classification des paramètres d'un SIF.

Classe	Paramètres
Logique	• Mécanisme de raisonnement
	• Opérateurs flous
	• Type des fonctions d'appartenance
	• Méthode de défuzzification
Structurelle	• Variables d'entrées/sorties
	• Nombre des fonctions d'appartenance
	• Nombre des règles floues
Connective	• Antécédents des règles
	• Conséquences des règles
	• Poids des règles dans certains cas
Opérationnelle	• Degrés d'appartenance

- **Paramètres logiques :** Ils caractérisent les fonctions et les opérateurs qui définissent le type de transformations subies pendant le processus d'inférence. Il s'agit notamment de la forme des fonctions d'appartenance, des opérateurs appliqués pour les connecteurs **ET** ou **OU**, des implications et de la méthode de défuzzification utilisée.
- **Paramètres structurels :** Ils sont liés, principalement, à la taille de la base de règles du système flou. Ils comprennent le nombre de variables d'entrées/sorties, le nombre des sous ensembles flous '*variables linguistiques*' pour chaque variable et le nombre de règles floues.

- **Paramètres de connectivité :** Ils sont liés à la topologie du système. Ces paramètres définissent le lien entre les différentes propositions dans la base de règles : les antécédents, les conséquences et les poids des règles dans certaines situations [35] - [36].
- **Paramètres opérationnels :** Ces paramètres définissent la cartographie entre les représentations numériques et linguistiques des variables. Ils caractérisent les degrés d'appartenance pour les variables linguistiques.

Une représentation graphique de l'ensemble des règles, appelée matrice d'inférence ou matrice de règles, couramment utilisée dans la littérature, permet la synthèse du cœur du système flou. La figure 1.7 représente une matrice d'inférence pour un système à 2 variables linguistiques d'entrée x_1, x_2 et une sortie u. La case en gris de la figure 1.7 représente la règle :

SI x_1 **est '**A_{12}**' ET** x_2 **est '**B_{12}**' ALORS** u **est '**R_5**'**

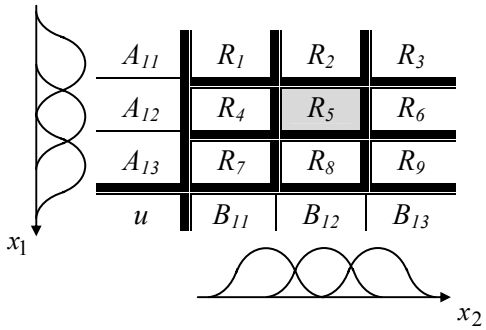


Fig. 1.7 : Fusion traditionnelle des règles : Matrice d'inférence complète.

Lorsque toutes les cases de la matrice d'inférence sont remplies, il s'agit de règles d'inférence complètes. Dans le cas contraire, il s'agit de règles d'inférence incomplètes (Fig. 1.8). La représentation matricielle de la base de règles devient plus complexe lorsque le nombre de variables linguistiques d'entrées est supérieur à 3.

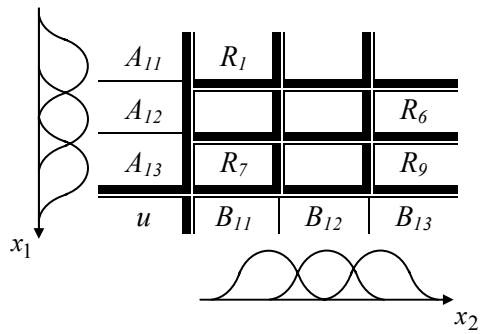


Fig. 1.8 : Fusion modifiée des règles : Matrice d'inférence incomplète.

Afin d'assurer une commande performante d'un système, la base de règles floues du SIF doit posséder certaines caractéristiques. La continuité, la consistance et la complétude de l'ensemble de règles sont les caractéristiques les plus significatives.

Une base de règles floues est complète si, quelle que soit la combinaison dans l'espace d'entrée, il existe une valeur de commande. Une base de règles est incomplète, si elle existe une situation de l'espace d'entrée pour laquelle aucune règle n'est activable. L'incomplétude conduit à une discontinuité indésirable dans la loi de commande. Cette propriété dépend de la partition des espaces et de l'écriture des règles, c'est-à-dire, les fonctions d'appartenance doivent couvrir toute la dynamique possible des variables. L. T. Kóczy propose l'interpolation entre les règles floues comme solution à ce problème [37]. Le tableau 1.4 donne les différentes situations possibles des bases des règles en fonction d'un important facteur : *Mesure de complétude*. Une mesure de complétude (C) d'une base de règles floues est définie par [38] :

$$C(\underline{x}) = \sum_{k=1}^{N} \left\{ \prod_{i=1}^{m} \mu_{X_i^k}(x_i) \right\} \tag{1.28}$$

Tab. 1.4 : Caractéristiques de la base de règles.

Valeur	Situation
$C(\underline{x}) = 0$	Base de règles *incomplète*.
$0 < C(\underline{x}) < 1$	Base de règles *sous-complète*.
$C(\underline{x}) = 1$	Base de règles *strictement complète*.
$C(\underline{x}) > 1$	Base de règles *sur-complète* (redondante).

Dans certaines situations, la prise en considération des surfaces '*hedges*' des fonctions d'appartenance est une nécessité pour permettre le contrôle de la forme de ces fonctions [39]. Des modificateurs linguistiques (*linguistic hedges*) sont introduits pour le réglage de la flexibilité des fonctions d'appartenance. Deux des modificateurs les plus connus sont le modificateur linguistique *de concentration* '**très**' et *de dilatation* '**plus-ou-moins**'. Leurs expressions sont :

- Concentration : $\mu_{A'}(x) = \mu_A(x)^p$ (1.29)
- Dilatation : $\mu_{A'}(x) = \mu_A(x)^{1/p}$ (1.30)

où $p > 1$ est le paramètre modificateur. Les effets des modificateurs de concentration et de dilatation sur la fonction d'appartenance triangulaire sont illustrés par la figure 1.9, dans le cas où $p = 2$.

Des modificateurs de l'intensité et de Blur sont aussi utilisés dans certains cas [40], tel que :

- Intensité : $$\mu_{A'}(x) = \begin{cases} 2^{p-1} \cdot \mu_A(x)^p & \text{si } \mu_A(x) \le 0.5 \\ 1 - 2^{p-1} \cdot \left(1 - \mu_A(x)^p\right) & \text{ailleurs} \end{cases} \tag{1.31}$$

- Blur : $$\mu_{A'}(x) = \begin{cases} (\mu_A(x))^2 & \text{si } \mu_A(x) \le 0.5 \\ 1-(1-\mu_A(x))^2 & \text{ailleurs} \end{cases} \quad (1.32)$$

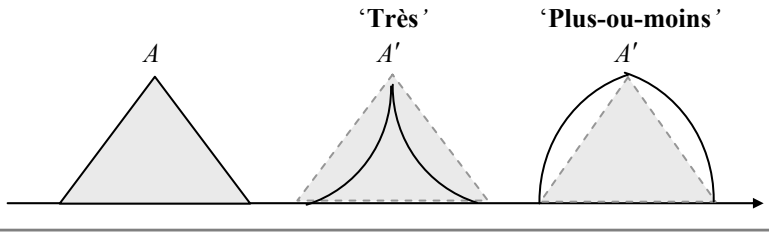


Fig. 1.9 : Effets de modificateurs linguistiques.

Zadeh a proposé 5 types de concentrations et 3 types de dilatations des surfaces des fonctions d'appartenance décrits dans le tableau 1.5 [41]. Bin-Da Liu et al. ont proposé un algorithme hybride pour améliorer les performances d'un contrôleur flou. Les AG sont utilisés pour la recherche des modificateurs linguistiques optimaux assurant les performances désirées [42] - [43].

Tab. 1.5 : Valeurs des modificateurs linguistiques proposés par L. A. Zadeh.

Modificateur p	Description linguistique
4	Absolument
2	Très
1.75	Beaucoup plus
1.5	Plus 'More'
1.25	Plus
1.0	Vide
0.75	Moins
0.5	Plus ou moins
0.25	Légèrement

En modélisation floue, la sélection de la structure implique, habituellement, le choix des facteurs suivants :

- Variables d'entrées et de sorties.
- Structure des règles y compris le type de raisonnement à utiliser.
- Nombre et type de fonctions d'appartenance pour chaque variable.
- Type de mécanisme d'inférence, opérateurs logiques et méthode de défuzzification.

Une fois que la structure est fixée, la performance de la méthode de modélisation peut être réglée avec précision en ajustant les paramètres. Les termes ajustables des modèles linguistiques sont les paramètres des fonctions d'appartenance des prémisses et des conséquences (détermination de sa forme et de sa position) ainsi que les règles (détermination de la correspondance entre les régions floues des antécédentes et des conséquences).

Pour leurs parts, les modèles flous de TS ont des paramètres tant dans les fonctions d'appartenance des antécédentes que dans les fonctions des conséquences (paramètres de

TS). Dans le deuxième chapitre, une étude détaillée de l'état de l'art des méthodes de modélisation des SIF a été introduite.

1.1.6. Conclusions

D'un point de vue mathématique, un SIF n'est qu'une fonction non linéaire d'un espace d'entrée vers un espace de sortie. D'un point de vue logique, un SIF est une structure de traitement parallèle et de prise de décisions. Sa structure consiste en 4 partie principales : La fuzzification, la base de règles, l'inférence et la défuzzification.

Pendant les vingt dernières années, les SIF, dont les bases relèvent de la théorie des ensembles flous proposés par Zadeh [2], sont devenus très populaires. Les applications dans la modélisation, le contrôle, le traitement de signal, la supervision des procédés et la prise de décision sont en effet autant d'application qui démontrent la capacité des SIF à traiter des problèmes hautement non linéaires grâce à l'utilisation de connaissances expertes. La logique floue aide à gérer des systèmes complexes de façon simple et facilement explicitable '*l'expertise humaine*'. La difficulté de sa mise en oeuvre réside dans la mise au point d'une base de connaissances floues (structure & paramètres du SIF). Ce problème peut être résolu par l'utilisation des méthodes d'extraction automatique des connaissances : Les méthodes analytiques, les TIA, ...etc.

1.2. Réseaux de neurones artificiels

1.2.1. Généralités sur le connexionnisme

Les RNA ont été développés comme généralisations des modèles mathématiques des systèmes nerveux biologiques. Les RNA sont une des composantes importantes des TIA. Un RNA est une structure de traitement parallèle et distribuée de l'information, avec une topologie spécifique. Les interactions mutuelles de ces éléments sont définies par les poids synaptiques des interconnexions. Un neurone artificiel (formel) N_F (Fig. 1.10), peut être défini par le quintuple suivant :

$$N_F = \{X, W, H, \Psi, G\} \tag{1.33}$$

tels que :

$X = [x_0, x_1, ..., x_n]^T$	:	Est le vecteur d'entrée avec $x_0 = 1$.
$W = [w_0, w_1, ..., w_n]^T$	:	Est le vecteur des poids synaptiques. Le poids w_0 est souvent appelé le seuil ou le biais.
H	:	Est la fonction d'entrée globale, telle que $H = \sum_{i=0}^{n} w_i \cdot x_i$.
Ψ	:	Est la fonction d'activation (de transfert) ou d'état.
$G = \Psi(H)$	:	Est la fonction de sortie.

La fonction d'activation *F* permet de définir l'état interne du neurone en fonction de son entrée. Tandis que la fonction *G* détermine la sortie du neurone en fonction de son état d'activation. La fonction de sortie est, généralement, considérée comme l'activation.

Chaque entrée d'un neurone est pondérée selon son importance par un poids synaptique et la somme de l'ensemble est présentée à l'entrée d'une fonction d'activation. On choisi d'habitude pour les neurones de la couche interne (cachée) des fonctions

d'activation non linéaires de type tangente hyperbolique ou une des variantes sigmoïdales. Un résumé des fonctions d'activation, couramment, utilisées est illustré par le tableau 1.6.

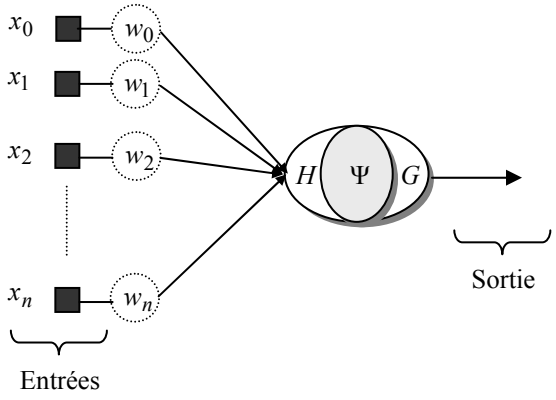


Fig. 1.10 : Structure générale d'un neurone formel.

Tab. 1.6 : Fonctions d'activation.

Type de la fonction	Modèle
Linéaire	$\Psi(x) = x$
Fonction seuil	$\Psi(x) = \begin{cases} +1 & \text{si} \quad x > 0 \\ 0 & \text{ailleurs} \end{cases}$
Fonction linéaire bornée	$\Psi(x) = \begin{cases} +1 & \text{si} \quad x > \Psi^+ \\ x & \text{si} \quad \Psi^- \le x \le \Psi^+ \\ -1 & \text{si} \quad x < \Psi^- \end{cases}$
Sigmoïde exponentielle	$\Psi(x) = \dfrac{1}{1+\exp(-\alpha.x)}$
Sigmoïde tangentielle	$\Psi(x) = \dfrac{1-\exp(-\alpha.x)}{1+\exp(-\alpha.x)}$
Fonction gaussienne	$\Psi(x) = \exp\left(-\dfrac{x^2}{\alpha^2}\right)$
Fonction logistique	$\Psi(x) = \dfrac{1}{1+\exp\left(-\dfrac{x^2}{\alpha^2}\right)}$
Tangente hyperbolique [44]	$\Psi(x) = \dfrac{2\cdot\beta}{1+\exp(-\alpha\cdot x)} - \beta \quad (\beta = 1.716, \alpha = 0.667)$

Le paramètre α détermine la raideur (la pente) de la région de transition (la forme) de la fonction d'activation ou de transfert des neurones. Ce type de fonction sera caractérisé par son gain qui affecte, sérieusement, les caractéristiques du réseau (Fig. 1.11) [45].

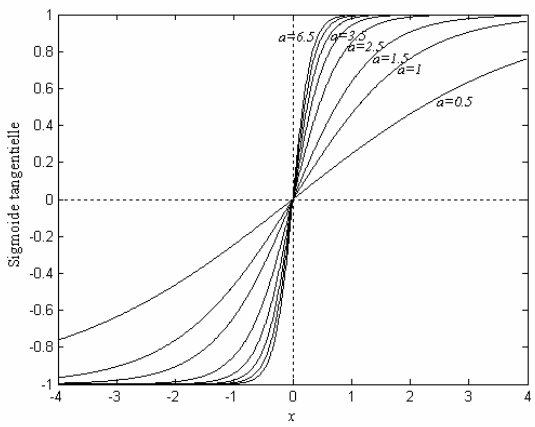


Fig. 1.11 : Allures de la sigmoïde tangentielle en fonction de α .

Les RNA regroupent un certain nombre de neurones formels connectés entre eux de diverses manières (topologies). Trois facteurs importants intervenant dans la définition d'un RNA :

- Sa topologie.
- Sa fonction d'activation ou de transfert.
- La méthode d'apprentissage.

La structure du réseau peut prendre 3 formes :

- Réseau multicouches ou du type feedforward (Fig. 1.12).
- Réseau totalement interconnecté (Fig. 1.13).
- Réseau récurrent, partiellement, connecté (Fig. 1.14).

L'analyse et la conception des RNA font appel à plusieurs outils mathématiques, en particulier :

- La théorie des modèles connexionnistes.
- La théorie des systèmes adaptatifs et des systèmes d'apprentissage.
- La théorie des systèmes dynamiques non linéaires.

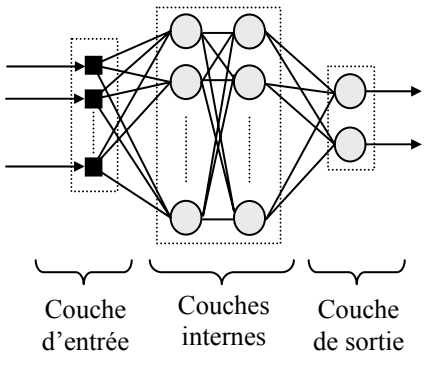


Fig. 1.12 : Architecture d'un réseau PMC.

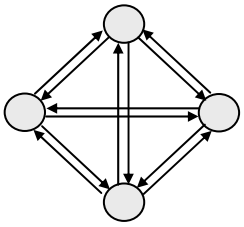
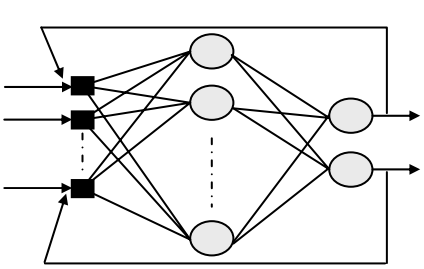


Fig. 1.13 : Réseau totalement connecté.



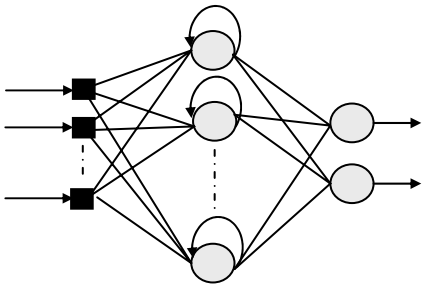


Fig. 1.14 : Réseaux dynamiques partiellement connectés.

Une vue schématique d'un RNA est représentée par la figure 1.15 pour une seule couche cachée. L'expression de la sortie du réseau en fonction des entrées s'exprime de la manière suivante :

$$y_i = \sum_{k=1}^{n_c} w_k^{(i)} \cdot \Psi\left(\sum_{j=1}^{m} v_{kj} x_j \right) \qquad i = 1, ..., n \tag{1.34}$$

n_c est le nombre des neurones dans la couche cachée. Les w_k et v_{kj} sont, respectivement, les poids des couches d'entrée et de sortie. La fonction Ψ représente la fonction d'activation du neurone (sigmoïde, gaussienne, tangente hyperbolique, ...etc.).

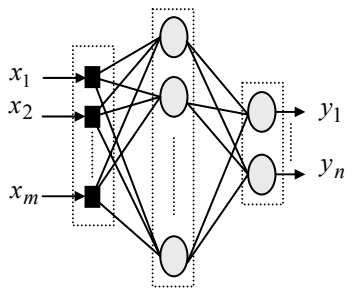


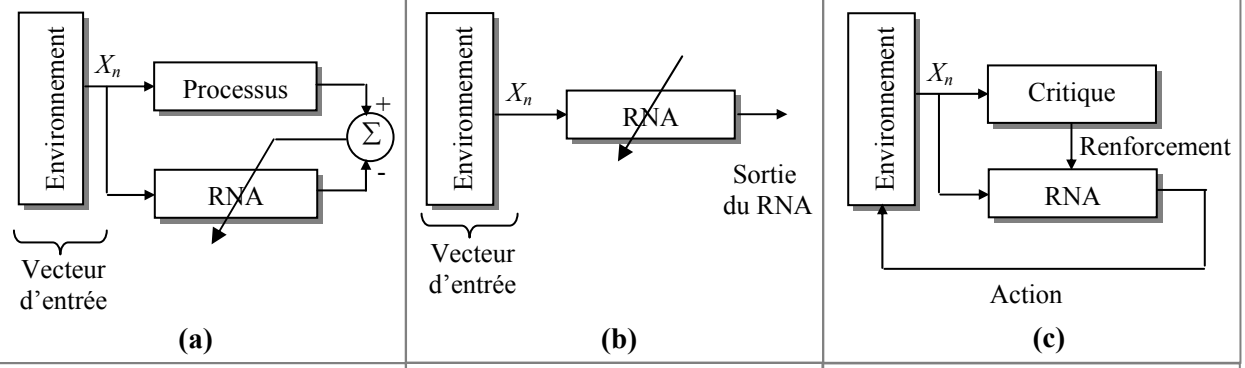
Fig. 1.15 : RNA à une couche cachée.

La synthèse d'un RNA comprend plusieurs étapes : Le choix du type de réseau, du nombre de neurones, du nombre de couches et de l'algorithme d'apprentissage.

Une fois, la topologie du réseau est choisie, il faut ensuite identifier les poids du réseau au cours de la phase d'apprentissage. L'apprentissage des RNA est une phase qui permet de déterminer ou de modifier les paramètres du réseau, afin d'adopter un comportement désiré. Plusieurs algorithmes d'apprentissage ont été développés depuis la première règle d'apprentissage de Hebb (1949). Ces algorithmes d'apprentissage sont classés en 3 catégories : S*upervisé*, *non supervisé* et *par renforcement*. Le tableau 1.7 résume les caractéristiques de chaque type.

Tab. 1.7 : Modes d'apprentissage des RNA.

Type	Description
Supervisé (Fig. 1.16a)	• Ce mode dispose d'un comportement de référence (présence d'un maître qui fournit la réponse désirée), vers lequel il tente de faire converger le réseau. Une fois le RNA présente des performances acceptables, il peut être utilisé pour répondre au besoin qui a été à l'origine de sa construction.
Non supervisé (Fig. 1.16b)	• Les données fournies en entrée ne contiennent pas d'information sur la sortie désirée. L'apprentissage est réalisé à l'aide de règles qui modifient les paramètres du réseau en fonction des exemples fournis en entrée. Les modèles de Kohonen et de Grossberg sont 2 exemples parfaits qui utilisent l'apprentissage non supervisé.
Par renforcement (Fig. 1.16c)	• Ce mode suppose qu'un comportement de référence n'est pas disponible. Mais en revanche, il est possible d'obtenir des indications qualitatifs (exemple : *correct / incorrect*), ou lacunaires (exemple : *une indication tous les N exemples d'entrées / sorties*) sur les performances du réseau.



(a) : Apprentissage supervisé. **(b) :** Non supervisé. **(c) :** Par renforcement.

Fig. 1.16 : Structures d'apprentissage des RNA.

Remarque :

La différence entre ces 3 types d'apprentissage réside dans le critère de performances implicite ou explicite.

L'apprentissage d'un PMC peut être vu comme un problème d'optimisation non linéaire '*nonlinear optimization problem*' [46]. Plusieurs variantes de méthodes peuvent servir, à savoir :

- Rétropropagation de l'erreur (gradient du premier ordre) [3].
- Méthodes de Newton, Levenberg-Marquardt et de Broyden-Fletcher-Goldfarb-Shanno (BFGS) (gradient du deuxième ordre) [47].
- Les méthodes évolutionnaires [48] - [51].
- Apprentissage adaptatif par la méthode du filtrage de Kalman [52].

Les interactions mutuelles des éléments d'un RNA sont définies par les poids des interconnexions. Une loi d'apprentissage permet d'ajuster les poids des connexions w tout en minimisant le critère de performances désirées. Dans les méthodes d'optimisation, le critère est, généralement, défini par :

$$J(w) = f(d_i - y_i, P) \tag{1.35}$$

d_i est la i^{eme} composante de la sortie désirée et y_i est la i^{eme} sortie du réseau PMC. $f(\cdot)$ est, le plus souvent, l'erreur quadratique sur l'ensemble des données d'apprentissage P dans le cas des méthodes du premier et du deuxième ordre. Les méthodes du gradient du premier ordre sont basées sur la règle d'adaptation suivante :

$$w(n+1) = w(n) - \eta_w(n) \cdot \nabla J(w(n)) \tag{1.36}$$

avec $w(n)$ est le vecteur des poids ajustables à l'itération n. $\eta_w(n)$ est le pas d'apprentissage et $\nabla J(w(n))$ est le Jacobien du réseau défini par :

$$\nabla J(w(n)) = \left[\frac{\partial J(w)}{\partial w_1}, \frac{\partial J(w)}{\partial w_2}, \ldots, \frac{\partial J(w)}{\partial w_M} \right]^T \tag{1.37}$$

Le problème d'optimisation non linéaire est ainsi résolu en employant le premier terme de la décomposition en série de Taylor (le gradient). Les méthodes du gradient du deuxième ordre utilisent, en plus, le second terme :

$$J(w) \approx J(w_0) + \nabla J(w_0)^T \cdot (w - w_0) + \frac{1}{2}(w - w_0)^T \cdot H(w_0) \cdot (w - w_0) \tag{1.38}$$

avec $H(w_0)$ est la matrice Hessienne dont les éléments sont les dérivées secondes de $J(w)$ par rapport aux poids du réseau. w_0 est le vecteur des poids initiaux.

$$H(w) = \frac{\partial^2 J(w)}{\partial w^2} \tag{1.39}$$

Remarque :

> Les méthodes du gradient du deuxième ordre présentent des résultats meilleurs par rapport aux méthodes du gradient du premier ordre [53].

1.2.2. Modélisation d'un PMC

Considérons un PMC constitué de *L* couches (*L-1* couches cachées). On désigne par n_l le nombre des neurones de la couche l, $z_i^{(l)}$ la sortie du neurone i de la couche l et

$w_{ij}^{(l)}$ le poids reliant le neurone j de la couche $l-1$ au neurone i de la couche l (Fig. 1.17). Les sorties d'un PMC se formulent sous la forme récursive suivante [54] :

$$z_i^{(l)}(t) = \begin{cases} x_i(t) & \forall i = 1, ..., n_0 \quad \text{si} \quad l = 0 \\ f\left(\sum_{j=1}^{n_l - 1} w_{ij}^{(l)} \cdot z_j^{(l-1)}(t) + \theta_j^{(l)} \right) & \forall i = 1, ..., n_l \quad \text{si} \quad l > 0 \end{cases} \tag{1.40}$$

où les $x_i(t)$ sont les variables explicatives du problème. $f(\cdot)$ est la fonction de transfert choisie pour les différentes couches et θ représentent des constantes ou *termes de biais*. Pour un PMC à une seule couche cachée, les sorties s'écrivent :

$$z_k(t) = f\left(\sum_{j=1}^{n_l - 1} w_{ij}^{(2)} \cdot f\left(\sum_{i=1}^{n_0} w_{ji}^{(1)} \cdot x_i(t) + \theta_j^{(1)} \right) + \theta_k^{(2)} \right) \qquad k = 1, ..., n_2 \tag{1.41}$$

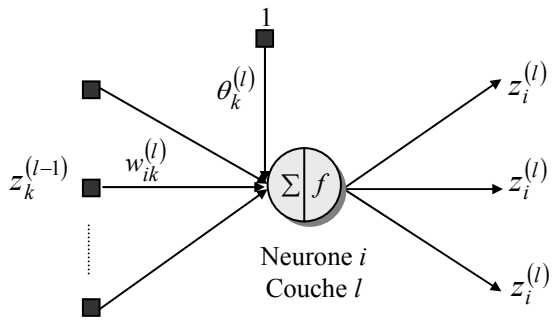


Fig. 1.17 : Structure générale d'un NF dans une couche cachée.

Les termes de biais sont souvent considérés comme des neurones supplémentaires associés à une entrée fixée à 1. Le modèle (1.40) peut prendre une forme réduite dans le cas où le vecteur seuil est nul :

$$z_i^{(l)}(t) = \begin{cases} x_i(t) & \forall i = 1, ..., n_0 \quad \text{si} \quad l = 0 \\ f\left(\sum_{j=1}^{n_{l-1}} w_{ij}^{(l)} \cdot z_j^{(l-1)}(t) \right) & \forall i = 1, ..., \ n_l \quad \text{si} \quad l > 0 \end{cases} \tag{1.42}$$

avec, par définition, $z_0^{(l)} = 1$ pour $(l = 1, ..., L-1)$.

De nombreux travaux ont permis de montrer qu'un tel modèle neuronal est un approximateur universel [3] & [55] - [57]. Une des conséquences pratiques de ces travaux montre qu'un PMC à une seule couche cachée est capable d'approcher, avec précision arbitraire, toute fonction continue, pourvu que le nombre de neurones dans la couche cachée soit suffisant. Les conditions initiales ainsi que l'ordre de présentation des exemples peuvent avoir une grande influence sur les résultats. De plus, le temps de calcul est un facteur très significatif et dépend de la taille du réseau. Travailler avec des RNA demande donc beaucoup d'expérience, malgré leur apparente facilité d'utilisation.

1.2.3. Apprentissage des paramètres d'un PMC

Cette partie est consacrée à la détermination des poids (biais θ compris) à partir de l'ensemble des données d'apprentissage. On peut déterminer les poids synaptiques tout en minimisant le critère de performances désirées C_{pd} (fonction coût dérivable) ou le risque empirique dans certaines références [54] & [58] - [60] par l'algorithme de la *rétropropagation* basée sur la descente du gradient, dont l'idée est la propagation des erreurs de la sortie vers l'entrée à travers le réseau [61]. Soit un PMC constitué d'un nombre arbitraire de couches (resp. de neurones) et pour un C_{pd} quelconque. On désigne par $s_i^{(l)}$ la sommation pondérée des entrées du neurone i sur la couche l :

$$s_i^{(l)} = \sum_{k=0}^{n_{l-1}} w_{ik}^{(l)} \cdot z_k^{(l-1)} \tag{1.43}$$

Soient :

$$\frac{\partial C_{pd}}{\partial w_{ij}^{(l)}} = \frac{\partial C_{pd}}{\partial s_i^{(l)}} \cdot \frac{\partial s_i^{(l)}}{\partial w_{ij}^{(l)}} \quad \forall l = 1, ..., L; \quad \forall i = 1, ..., n_l; \quad \forall j = 0, ..., n_{l-1} \tag{1.44}$$

sont les dérivées partielles du C_{pd} par rapport aux poids du réseau. On pose :

$$\alpha_i^{(l)} = \frac{\partial C_{pd}}{\partial s_i^{(l)}} \tag{1.45}$$

L'expression (1.44) devient :

$$\frac{\partial C_{pd}}{\partial w_{ij}^{(l)}} = \alpha_i^{(l)} \cdot z_j^{(l-1)} \qquad \forall l = 1, ..., L; \qquad \forall i = 1, ..., n_l; \qquad \forall j = 0, ..., n_{l-1} \tag{1.46}$$

En effet :

$$\alpha_i^{(L)} = \frac{\partial C_{pd}}{\partial s_i^{(L)}} = \frac{\partial C_{pd}}{\partial z_i^{(L)}} \cdot \frac{\partial z_i^{(L)}}{\partial s_i^{(L)}} = \frac{\partial C_{pd}}{\partial z_i^{(L)}} \cdot f'\left(s_i^{(L)}\right) \tag{1.47}$$

Remarque :

Dans le cas où la fonction d'activation f est de type logistique, sa dérivée est de la forme :

$$f'\left(s_i^{(L)}\right) = z_i^{(L)} \cdot \left(1 - z_i^{(L)}\right) \tag{1.48}$$

Si f est une tangente hyperbolique, sa dérivée est une fonction simple de la forme :

$$f'\left(s_i^{(L)}\right) = 1 - \left(z_i^{(L)}\right)^2 \tag{1.49}$$

Le calcul de $\alpha_i^{(l)}$ dans l'expression (1.47) se traduit par la règle de dérivation en chaîne suivante :

$$\begin{aligned}\alpha_i^{(l)} &= \frac{\partial C_{pd}}{\partial s_i^{(l)}} = \sum_{k=1}^{n_{l+1}} \frac{\partial C_{pd}}{\partial s_k^{(l+1)}} \cdot \frac{\partial s_k^{(l+1)}}{\partial s_i^{(l)}} = \sum_{k=1}^{n_{l+1}} \alpha_k^{(l+1)} \cdot \frac{\partial s_k^{(l+1)}}{\partial z_i^{(l)}} \cdot \frac{\partial z_i^{(l)}}{\partial s_i^{(l)}} \\ &= \sum_{k=1}^{n_{l+1}} \alpha_k^{(l+1)} \cdot w_{ki}^{(l+1)} \cdot f'\left(s_i^{(l)}\right)\end{aligned} \tag{1.50}$$

En conclusion, le calcul de $\alpha_i^{(l)}$ s'écrit sous la forme récursive suivante :

$$\alpha_i^{(l)} = \begin{cases} \dfrac{\partial C_{pd}}{\partial z_i^{(l)}} \cdot f'\left(s_i^{(l)}\right) & \text{pour les noeuds de sortie.} \\ f'\left(s_i^{(l)}\right) \cdot \sum\limits_{k=1}^{n_{l+1}} \alpha_k^{(l+1)} \cdot w_{ki}^{(l+1)} & \text{pour les autres noeuds.} \end{cases} \tag{1.51}$$

Le principe de l'algorithme de la rétropropagation consiste à évaluer les gradients de l'erreur sur la couche de sortie, d'abord, puis sur l'avant dernière couche et ainsi de suite jusqu'à la couche d'entrée, d'où le nom de *rétropropagation* du gradient de l'erreur. La procédure globale d'apprentissage d'un PMC par rétropropagation du gradient de l'erreur se résume à 4 étapes :

- Initialisation du vecteur des poids synaptiques par des petites valeurs $\underline{W} \in [-1,1]$. La pratique la plus courante, selon S. Haykin, consiste à initialiser les poids à des valeurs aléatoires uniformément distribuées dans un petit intervalle $\left[-2.4/F_i, 2.4/F_i\right]$, où F_i est le nombre total de signaux entrants du neurone *i* dans le réseau [56].
- Présentation d'un couple d'exemples entrée/sortie désirés.
- Propagation : Calcul des sorties du réseau par l'expression (1.42).
- Rétropropagation : Modification des poids du réseau suivant la règle d'adaptation des poids suivante :

$$w_{ij}^{(l)}(t+1) = w_{ij}^{(l)}(t) - \eta_w \cdot \frac{\partial C_{pd}\left(w(t)\right)}{\partial w_{ij}^{(l)}(t)} \tag{1.52}$$

où η_w est le pas d'adaptation de la méthode du gradient. Les 3 dernières étapes sont répétées jusqu'à la fin de la procédure d'apprentissage (satisfaction du critère d'arrêt ou d'une performance d'optimalité).

Si la simplicité de la rétropropagation a, sans doute, aidé à populariser les PMC, il n'en demeure pas moins que cet algorithme souffre de nombreux défauts, parmi lesquels, on cite [54] :

- Une extrême lenteur de convergence.
- Une grande sensibilité aux conditions initiales.
- Une possibilité de rester bloqué dans des minimums locaux.
- Une tendance aux oscillations.

De nombreux travaux ont été consacrés à l'amélioration des performances de l'algorithme :

- Le *momentum* (moment) ou des méthodes du gradient conjugué visent à lisser la trajectoire des poids dans l'espace de recherche (évitement des oscillations) en ajoutant aux termes du gradient une fraction des modifications appliquées au pas précédent. La loi (1.52) devient alors :

$$w_{ij}^{(l)}(t+1) = w_{ij}^{(l)}(t) - \eta_w \cdot \frac{\partial C_{pd}(w)}{\partial w_{ij}^{(l)}(t)} + \underbrace{\alpha_w \cdot \left(w_{ij}^{(l)}(t-1) - w_{ij}^{(l)}(t-2)\right)}_{\text{Correction}} \qquad 0 < \alpha_w < 1 \qquad (1.53)$$

- Les méthodes à pas η_w et α_w adaptatifs permettent d'accélérer la convergence en augmentant le pas d'adaptation tant que l'erreur décroît et en revenant à des valeurs plus faibles de pas en cas d'augmentation du critère d'erreur [62] - [66]. Yu-Ju Chen et al. ont proposé un SIF pour l'adaptation du pas d'apprentissage de la rétropropagation [67]. Sun Yan-jing et al. ont proposé un algorithme de rétropropagation à pas adaptatifs (BP-ALM) permettant de réduire le temps d'apprentissage et d'éviter d'être '*emprisonné*' dans des minima locaux. Les coefficients η_w et α_w dans l'expression (1.53) sont altérés à chaque itération [68]. L'analyse de la stabilité de la boucle de commande du système par l'approche de Lyapunov est utilisée pour l'adaptation des coefficients η_w et α_w est proposée par Amit Bhaya et al. [69]. Un pas variable est proposé par Sang-Min Kim sous forme [70] :

$$\eta_w(t) = \frac{\eta_{\max} - \eta_{\min}}{2} \cdot \left(\frac{1 - \exp\left(-(t - \varepsilon/2)\cdot\delta\right)}{1 + \exp\left(-(t - \varepsilon/2)\cdot\delta\right)} + 1 \right) + \eta_{\min}$$
$$\delta = \frac{2R}{\varepsilon} \qquad (1.54)$$

Les valeurs inférieure et supérieure de η_w sont définies, respectivement, par $\eta_{\min}$ et $\eta_{\max}$. ε est un paramètre positif prédéterminé $\varepsilon \approx 100$ et R permet de définir la forme de la tangente de η_w.

- Pour simplifier le problème de dimensionnement rencontré lors de la conception d'un PMC, 2 démarches ont été proposées dans la littérature :
  - √ La première dite de *sélection* ou *destructive* : Elle consiste à construire un réseau complexe, puis à tenter de le réduire en supprimant les cellules ou les connexions redondantes, au cours ou à la fin de l'apprentissage [56].
  - √ La seconde dite *incrémentale* ou *constructive* : Elle consiste à commencer par le réseau le plus simple possible, puis à ajouter des neurones ou, éventuellement, des couches pendant l'apprentissage (création des nœuds dynamiques) [71].
- La logique floue est aussi utilisée pour la conception des RNA. Kihwan Eom et al. ont introduit un SIF pour le réglage des formes des fonctions d'activation par action sur ses gains. Les performances de la rétropropagation sont prouvées sur des exemples d'approximation des fonctions et de la reconnaissance des formes [72]. Mendil a utilisé la descente du gradient pour adapter le gain de la sigmoïde qui affecte sa forme, c'est-à-dire, les caractéristiques du réseau [73].

- D'autres chercheurs ont proposé une hybridation de 2 types d'apprentissage, supervisé et non supervisé pour le même réseau [74]. Tandis que d'autres préfèrent l'hybridation avec d'autres techniques d'optimisation, telles que les AG 'méthodes directes'. Selon J. M. Renders, l'évolution des RNA par les AG peut se faire selon 4 axes [48] :
  - √ Recherche directe des poids de connexions.
  - √ Recherche des paramètres d'apprentissage optimaux intervenant dans l'algorithme de la rétropropagation.
  - √ Recherche des poids initiaux du réseau suivie d'une descente du gradient pour un ajustement 'plus fin' de ces poids 'couplage direct'.
  - √ Recherche d'une topologie optimale. Dans ce cas, l'évaluation d'un individu nécessite une étape interne d'apprentissage (rétropropagation) afin d'ajuster les poids du réseau pour une architecture donnée. Il s'agit donc d'un problème d'optimisation double :
    - Optimisation structurelle réalisée par l'AG.
    - Optimisation paramétrique, habituellement, par la descente du gradient.
- Suite aux idées de J. M. Renders, certains auteurs ont développé des AG pour la conception des RNA. Le chromosome de l'AG codifie la totalité des poids du réseau à structure prédéfinie ainsi que la forme des fonctions d'activation [75] - [76]. Le schéma de la figure 1.18 résume la stratégie du codage proposée sur un RNA à 2 entrées, 2 couches cachées et une sortie. Les gènes constituent le chromosome de l'AG sont codés en nombre réels avec le vecteur des poids $\underline{W} \in [-1,1]$ el le gain de la sigmoïde tangentielle $\sigma \in [2.0, 5.0]$.

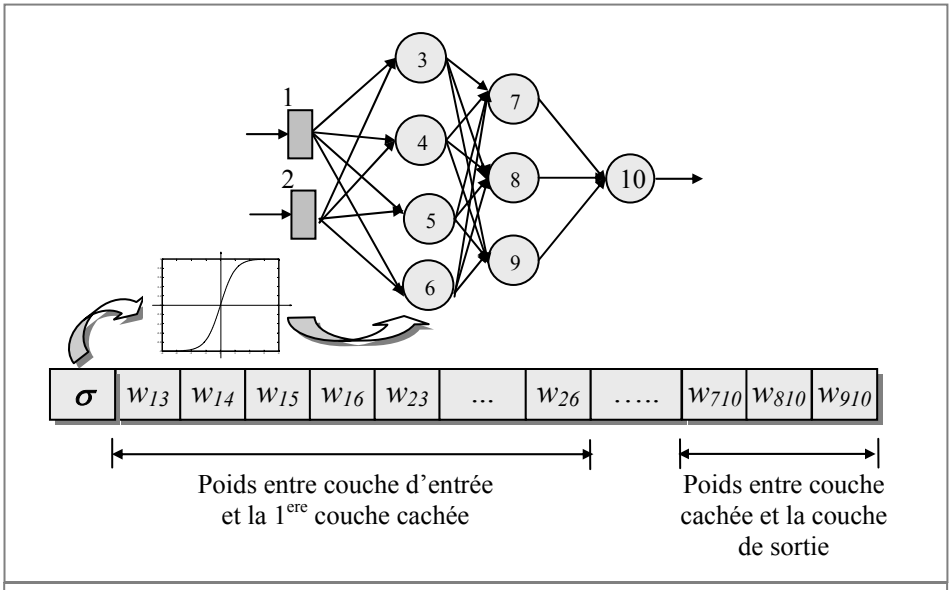


Fig. 1.18 : Stratégie de codage réel d'un RNA.

- D'autres travaux, enfin, préconisent l'utilisation des méthodes d'optimisation du second ordre fondée sur le calcul ou l'approximation de la matrice Hessienne (méthode de Newton, Levenberg-Marquardt, ... etc.) [47] & [52].

Les RNA peuvent être, également, implémentés en circuits électroniques, offrant ainsi la possibilité d'un traitement temps réel [77] - [79]. Leur utilisation est, principalement, guidée par leurs propriétés suivantes :

- Capacité d'apprentissage.
- Capacité de généralisation.
- Parallélisme dans le traitement (rapidité de traitement).
- Adaptés aux non linéarités des systèmes.

1.2.4. Conclusions

Un RNA est une structure de traitement parallèle et distribuée de l'information. Ce réseau est constitué de plusieurs éléments processeurs (neurones) avec une topologie spécifique. Les RNA sont dotés de règles d'apprentissage, mécanismes capables de modifier leurs poids, automatiquement, en fonction d'un critère interne ou externe.

Les RNA peuvent fournir une solution intéressante pour la modélisation et la commande des systèmes non linéaires. Leurs capacités de mémorisation, d'apprentissage et de généralisation, propriétés d'approximations universelles, d'adaptation et le parallélisme du calcul représentent des fonctions très utiles dans la théorie du contrôle des systèmes non linéaires [3], [48] & [55]. On peut, également, citer la facilité d'implémentation hardware et le fait que l'utilisation des RNA permet, parfois, d'éviter des analyses mathématiques très contraignantes.

1.3. Algorithmes Génétiques

1.3.1. Introduction

Les principes de base des AE, dont les plus connus sont les AG, s'inspirent de l'observation des phénomènes biologiques, plus précisément de la capacité des populations d'organismes vivants à s'adapter à leur environnement à l'aide de mécanismes de sélection et d'héritage génétique. En d'autres termes, ces AE représentent une version artificielle, informatique, de la théorie de l'évolution décrite par Charles Darwin.

Les AG, développés par John Holland et ses collaborateurs à l'université de Michigan puis Goldberg qui les a utilisés pour résoudre les problèmes d'optimisation de fonctions [4] & [80]. D'autres chercheurs ont suivi cette voie, en particulier Michalewics, Lance Chambers, M. Jamshidi, Randy L. Haupt, ...etc. [81] - [84]. Il s'agit d'une procédure de traitement parallèle souvent utilisée dans les problèmes d'optimisation.

1.3.2. Principe de fonctionnement des AG

Les AG, basés sur la métaphore de la discipline biologique de même nom, sont des méthodes d'optimisation d'ordre zéro. Ils ne font appel qu'à l'évaluation de coût en des points distincts et non à ses différentes dérivées. Les AG s'inspirent de la métaphore biologique des lois de la génétique naturelle qui président aux règles de l'évolution des espèces.

Chaque individu d'une population ($\boldsymbol{Pop.i}\big|_{i=1,\dots,P}$) est baptisé '*chromosome*' ($\boldsymbol{Ch}_{x_i}^{(j)}\Big|_{i=1,\dots,N}^{j=1,\dots,T}$) (individu ou variable). Chaque chromosome est constitué d'informations élémentaires appelées '*gènes*' ($\vartheta_i\big|_{i=1,\dots,m}$). Chaque gène peut prendre différentes valeurs appelées '*allèles*', dans une représentation (codage) donnée. Lorsque l'information est codée en binaire, les seuls allèles possibles sont 0 et 1. Mais, nous pouvons, également, coder l'information sous forme de nombres réels. La fonction coût est souvent appelée '*fonction d'évaluation*' ou '*fonction d'aptitude*' (*fitness function*) ou *fonction objectif*. Elle représente le lien entre l'AG et l'environnement de fonctionnement (phénotype).

A chaque itération, appelée '*génération*' (***gen***), une population va donner naissance à une nouvelle population. A chaque génération, on fait évaluer les performances des individus parents. Sur la base de cette évaluation, l'opération de sélection aura lieu. Les individus sélectionnés participent à la production d'une nouvelle génération. La figure 1.19 est une formulation graphique des vocabulaires des AG [85] - [86].

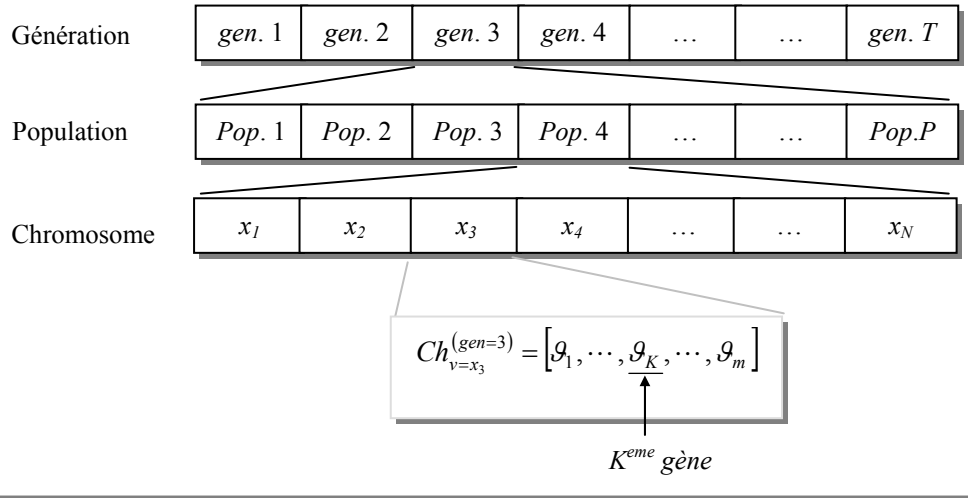


Fig. 1.19 : Structure des données dans un AG.

La reproduction se fait par croisement des gènes des chromosomes et peut s'accompagner de mutations. La nouvelle génération peut contenir des individus parents ou être, entièrement, constituée d'individus enfants.

L'AG de base peut être défini par un ensemble de paramètres fonctionnels assurant sa bonne exécution :

$$\mathbf{AG} \equiv \{ Type\_coding, Pop\_ini, Max\_Pop, \text{Parm\_op}\ (P_m(gen), P_c(gen)), gen, Fitness, Max\_gen \} \quad (1.55)$$

avec :

Type_coding	:	Le type de codage utilisé (réel, binaire, ...etc.).
Pop_ini	:	Population initiale.
Max_Pop	:	Taille maximale de la population.
Parm_op ($P_m(gen)$, $P_c(gen)$)	:	Probabilités des opérateurs génétiques de mutation et de croisement, respectivement.
Fitness	:	La fonction d'aptitude (Fitness).
gen	:	Génération en cours.
Max_gen	:	Nombre maximal de génération.

Tous les AG courants ne se différencient que par la taille de leur population, le mode du codage utilisé, les méthodes de sélection, de croisement et les formes de la mutation. L'implémentation d'un AG est spécifique au problème à résoudre. Les considérations principales sont les suivantes [87] :

- Détermination d'une configuration initiale de l'algorithme, c'est-à-dire, la taille de la population, les probabilités pour l'application des opérateurs génétiques, ...etc.
- Définir le mode de représentation des chromosomes (binaire, entier ou réel).
- Etablir la façon de générer la population initiale.
- Choix de la fonction d'évaluation.
- Choix de la méthode de sélection.
- Choix des opérateurs génétiques : Le croisement et la mutation doivent être adaptés au type de représentation choisie.

Partons d'une population initiale de N chromosomes générés, aléatoirement $\left(Ch_{x_i}^{(0)},\ (i=1,...,N)\right)$, où l'on suppose que ces N chromosomes couvrent, le mieux possible, l'ensemble du domaine de définition. On évalue leur performance relative (fitness) qui permet de quantifier sa qualité. Sur la base de ces performances, on crée une nouvelle population des individus potentiels en utilisant les opérateurs évolutionnaires simples : La sélection, le croisement et la mutation. On recommence ce cycle jusqu'à l'obtention d'une solution satisfaisante. Une description abstraite de l'AG de base est donnée par le schéma de la figure 1.20.

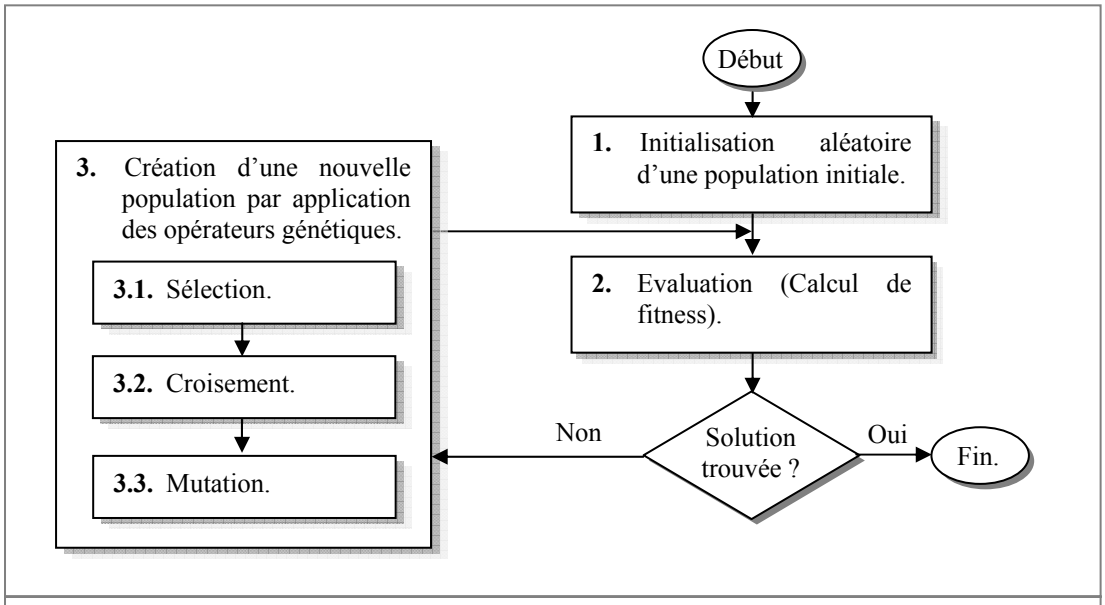


Fig. 1.20 : Organigramme général d'un AG de base.

1.3.3. Codage

Les AG diffèrent des autres méthodes d'optimisation car ils utilisent un codage des variables (solutions), plutôt que les variables elles-mêmes. Le codage désigne le processus qui transforme les variables en un chromosome.

$$x_c = \left[(x_c)_1, (x_c)_2, ..., (x_c)_m\right] \in \Re^m \rightarrow \left[\vartheta_1, \vartheta_2, ..., \vartheta_m\right] \tag{1.56}$$

L'AG de base emploi un codage binaire qui s'appuie sur la représentation $\{0,1\}$. Le codage binaire suppose en pratique la définition d'un intervalle admissible pour les variables et une discrétisation de cet intervalle. Pour chaque variable réelle $(x_c)_i$, on définit

une limite inférieure $\left(x_c^{\min}\right)_i$ et une limite supérieure $\left(x_c^{\max}\right)_i$ au domaine de variation. En supposant que la discrétisation fournisse une précision de π_i^{cs} chiffres significatifs, le nombre de gènes nécessaire au codage de la variable $\left(x_c\right)_i$ est le plus petit entier l_i vérifiant :

$$\left(\left(x_c^{\max}\right)_i - \left(x_c^{\min}\right)_i\right) \cdot 10^{\pi_i^{cs}} \leq 2^{l_i} - 1 \tag{1.57}$$

Un chromosome étant constitué des gènes associés à chaque variable mis bout à bout, le nombre total de gènes d'un chromosome, caractérisant un individu et un point dans l'espace de recherche est alors :

$$l = \sum_{i=1}^{m} l_i \tag{1.58}$$

La transformation d'un chromosome en variables réelles peut se faire de manière naturelle par la relation :

$$\left(x_c\right)_i = \left(x_c^{\min}\right)_i + \frac{\left(x_c^{\max}\right)_i - \left(x_c^{\min}\right)_i}{2^{l_i} - 1} \sum_{i_{bit}=1}^{l_i} 2^{i_{bit}-1} \cdot val_{bin}\left(i_{bit}\right) \tag{1.59}$$

où $val_{bin}\left(i_{bit}\right)$ représente la valeur binaire, 0 ou 1, du gène i_{bit} associé à la variable $\left(x_c\right)_i$.

1.3.4. Evaluation

En raison de son analogie avec le processus de l'évolution naturelle, l'AG est naturellement formulé en terme de maximisation. Etant donnée une fonction F réelle à une ou plusieurs variables, le problème d'optimisation dans un espace de recherche V s'écrit comme suit :

$$\max_{\vartheta \in V}\left\{F(\vartheta)\right\} \tag{1.60}$$

Dans beaucoup de situations, l'objectif est exprimé sous forme de minimisation d'une fonction coût ou d'un critère de performances $J(\cdot)$:

$$\min_{\vartheta \in V}\left\{J(\vartheta)\right\} \tag{1.61}$$

Pour passer d'un problème de minimisation à un problème de maximisation bien adapté à la nature de l'AG, plusieurs méthodes de transformations sont proposées, connues sous l'appellation de *techniques d'aptitudes* «en anglais : fitness techniques» [80]. La plus utilisée est :

$$\underset{\vartheta \in V}{F\left(\vartheta\right)} = \frac{C^{te1}}{\left(1 + J\left(\vartheta\right)\right)^{C^{te2}}} \tag{1.62}$$

où C^{te2} est une constante qui affecte la performance. $C^{te1} \in]0,+\infty]$ est, convenablement, choisie de sorte que $F(\cdot)$ ne soit pas trop petite.

Le processus d'interaction entre l'AG et la fonction à optimiser, c'est-à-dire, l'environnement est schématisé par la figure 1.21.

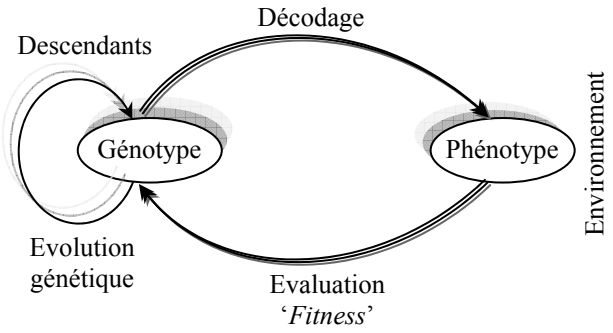


Fig. 1.21 : Processus d'interaction AG - Environnement.

Il est à noter qu'aucune certitude concernant la convergence de l'algorithme vers un optimum global n'est assurée et la solution 'en temps fini' ne constitue qu'une approximation de l'optimum. L'AG s'arrête quand un indice de performances donné ou un nombre maximum de génération est atteint ou quand la diversité de la population est trop faible :

$$\max_{\substack{\vartheta \in V \\ k \le Max\_Pop}} F_k(\vartheta) - \min_{\substack{\vartheta \in V \\ k \le Max\_Pop}} F_k(\vartheta) < \varepsilon \qquad (1.63)$$

où ε est un seuil donné.

En se basant sur le principe de fonctionnement des AG, certains de ses avantages sont tirés [80] & [84] :

- L'optimisation s'effectue dans le domaine continu comme dans le domaine discret.
- Les AG travaillent avec des données générées numériquement, données expérimentales, analytiques ou fonctions.
- Les AG, algorithmes itératifs de recherche globale, explorent en *parallèle* un ensemble de solutions possibles à un problème donné et non une seule solution. Les AG sont bien adaptés à l'implémentation hardware sur des machines parallèles [88] - [90].
- L'optimisation des variables avec des coûts, extrêmement, complexes.
- Fournir une liste de solutions optimales et non seulement une solution unique.
- Les AG possèdent une représentation codée et cherchent une représentation dans l'espace des solutions et non pas directement dans le domaine original.
- Les AG n'utilisent que les valeurs de la fonction à optimiser, pas sa dérivée ou une autre information auxiliaire.
- Les AG utilisent des règles de transitions probabilistes (pseudo aléatoires) et non déterministes comme outils pour guider l'exploration à travers les régions de l'espace de recherche.

1.3.5. Mécanisme d'un AG standard

A partir d'une population initiale de chromosomes créée, aléatoirement, l'AG génère de nouveaux chromosomes pour construire une nouvelle génération, par application des opérateurs de sélection et de la reproduction.

1.3.5.1. Sélection

La sélection permet d'identifier les individus susceptibles à la reproduction dans une population. Il y a plusieurs façons d'exercer une pression sélective sur la population. La plus radicale consiste à ne conserver que les meilleurs individus. Mais, les risques de perte de la diversité sont importants avec ce type de stratégie. Des stratégies de sélection plus souples ont été développées [80] - [81] & [91] :

a. Sélection par roue de fortune (proportionnelle)

Son principe est équivalent à une recherche linéaire sur une roue de fortune 'de loteries' dont les sections ont des tailles proportionnelles aux valeurs des fonctions d'évaluation des individus. Dans un problème d'optimisation de maximisation, on associe à chaque individu *i* une probabilité de sélection, notée P_{Sel}^{i}, proportionnelle à sa valeur F^{i} de la fonction objectif, donnée par :

$$P_{Sel}^{i} = \frac{F^{i}}{\sum_{j=1}^{Max\_Pop} \left(F^{j}\right)} \tag{1.64}$$

Chaque individu est alors reproduit avec la probabilité P_{Sel}^{i}. Certains individus (les 'bons') ont 'plus' de chance de participer à la reproduction et d'autres (les 'mauvais') seront éliminés.

b. Sélection par rang

Dans cette méthode, les individus sont classés en fonction de leurs fonctions d'évaluation. La probabilité de sélection d'un individu est déterminée en fonction de son ordre dans la population. Pour un problème de minimisation, les individus sont ordonnés selon l'ordre décroissant. Les probabilités de sélection sont indexées sur les rangs des individus :

$$P_{Sel}\left(Parent_{i}\right) = \frac{Rang\left(Parent_{i}\right)}{\sum_{j=1}^{Max\_Pop} Rang\left(Parent_{j}\right)} \tag{1.65}$$

c. Sélection aléatoire

La sélection des individus se fait d'une manière aléatoire sans prise en compte de leurs fonctions d'évaluation. Chaque individu a une chance d'être sélectionné avec une probabilité uniforme de :

$$P_{Sel}^{i} = \frac{1}{Max\_Pop} \tag{1.66}$$

Généralement, cette méthode est moins utilisée parce que la convergence de l'algorithme est trop lente.

d. Sélection par compétition 'tournoi'

Une compétition aura lieu entre les individus d'une sous population de taille $N\left(N \le Max\_Pop\right)$ prise au hasard. Le meilleur élément du groupe est choisi comme vainqueur et sera sélectionné à la reproduction. Si $N=1$, la méthode de sélection correspond à la sélection aléatoire. Si $N = Max\_Pop$, l'algorithme sera réduit à un algorithme de recherche locale fonctionnant sur une seule solution à la fois.

1.3.5.2. Reproduction

La reproduction représente la phase de l'évolution génétique. Cette opération permet la construction d'une nouvelle population 'génération', par application des opérateurs de croisement et de mutation.

a. Opérateurs de croisement

Le croisement a pour but d'assurer la diversité des solutions dans la population, tout en manipulant les composantes 'gènes' des chromosomes. Dans l'AG de base, les croisements sont envisagés avec 2 parents et génèrent 2 enfants. Ce n'est que l'échange partiel du matériel génétique entre parents pour obtenir des enfants qui prennent les caractéristiques de leurs parents.

L'opération de croisement est appliquée avec une probabilité P_c. Le mode de représentation des chromosomes joue un rôle important dans le choix du type de croisement : simple (en un point ou multipoints), arithmétique (simple ou entier) ou heuristique.

a.1. Croisement simple

Deux pairs de chromosomes Ch_ϑ et Ch_s sont sélectionnés aléatoirement. Un échange partiel de matériel génétique aura lieu à partir d'un site choisi aléatoirement. Une représentation schématique de cette opération est donnée par le schéma de la figure 1.22. Ce type de croisement est unique quelque soit le mode de codage des chromosomes.

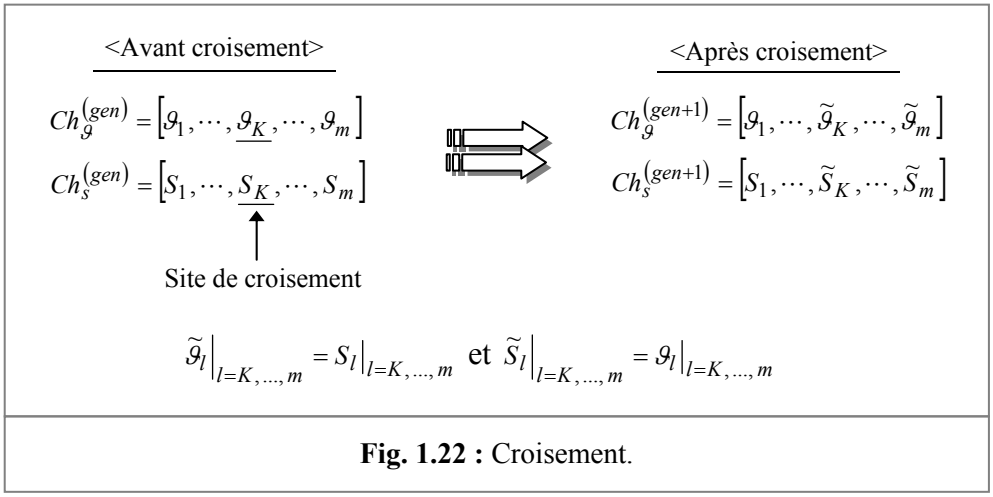


Fig. 1.22 : Croisement.

a.2. Croisement arithmétique simple

Dans ce mode de croisement, les valeurs des gènes sélectionnés à l'opération de croisement sont calculées par les expressions suivantes :

$$\widetilde{\vartheta}_j\Big|_{j=K,\ldots,m} = \alpha \cdot \vartheta_j + \beta \cdot S_j \tag{1.67}$$

$$\widetilde{S}_j\Big|_{j=K,\ldots,m} = \alpha \cdot S_j + \beta \cdot \vartheta_j \tag{1.68}$$

où α et β sont des nombres aléatoires $\in [0,1]$.

a.3. Croisement arithmétique entier 'whole arithmetical crossover'

Une combinaison linéaire des 2 chromosomes $Ch_{\vartheta}^{(gen)}$ et $Ch_{s}^{(gen)}$ aura lieu. Les chromosomes résultants sont donnés par :

$$Ch_{\vartheta}^{(gen+1)} = r \cdot Ch_{s}^{(gen)} + (1-r) \cdot Ch_{\vartheta}^{(gen)} \tag{1.69}$$

$$Ch_{s}^{(gen+1)} = r \cdot Ch_{\vartheta}^{(gen)} + (1-r) \cdot Ch_{s}^{(gen)} \tag{1.70}$$

où r est un nombre aléatoire $\in [0,1]$.

a.4. Croisement heuristique

Un nouveau chromosome $Ch_{\text{new}}^{(gen+1)}$ est généré à partir de croisement de $Ch_{\vartheta}^{(gen)}$ et de $Ch_{s}^{(gen)}$. Sa formule est :

$$Ch_{\text{new}}^{(gen+1)} = r\left(Ch_{s}^{(gen)} - Ch_{\vartheta}^{(gen)}\right) + Ch_{\vartheta}^{(gen)} \tag{1.71}$$

Dans ce mode de croisement, il faut réarranger les chromosomes afin d'obtenir des fonctions d'objectives $F_{it}\left(Ch_{\vartheta}^{(gen)}\right) \leq F_{it}\left(Ch_{s}^{(gen)}\right)$ pour des problèmes de minimisation et de maximisation.

b. Opérateurs de mutation

L'opérateur de mutation inverse ou modifie 'en appliquant des perturbations aléatoires ou de petites modifications' une ou plusieurs composantes 'gènes' d'un chromosome. Cet opérateur permet d'avoir une large diversité de solutions dans la population. La fréquence d'application de cet opérateur est contrôlée par un paramètre connu sous la probabilité '*taux*' de mutation P_m. Cette valeur est fixée par l'utilisateur dans certaines applications des AG. Par contre, elle peut changer, dynamiquement ou peut être adaptée automatiquement, par un processus d'évolution. La figure 1.23 schématise, clairement, le processus de mutation.

Dans la figure 1.23, $\widetilde{\vartheta}_K$ est la composante mutée de ϑ_K. Il existe un certain nombre d'opérateurs de mutation définis dans la littérature qui, évidemment, dépend du mode de représentation des chromosomes utilisé. Le schéma de la figure 1.24 donne le classement par catégories de la représentation à laquelle ces opérateurs sont appliqués.

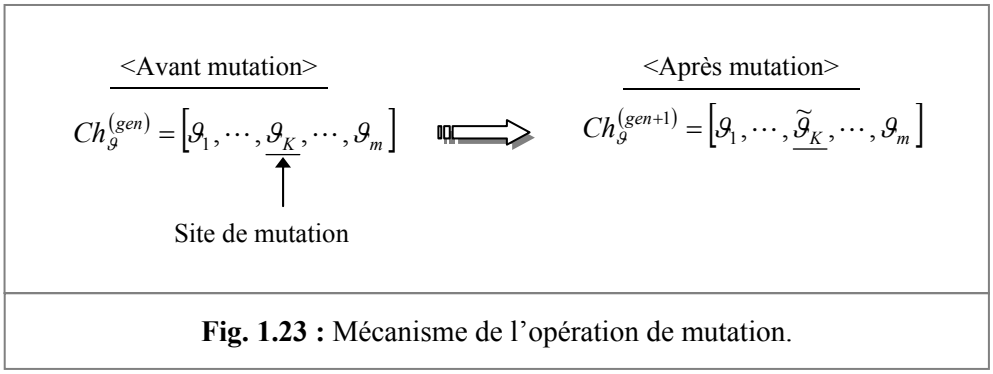


Fig. 1.23 : Mécanisme de l'opération de mutation.

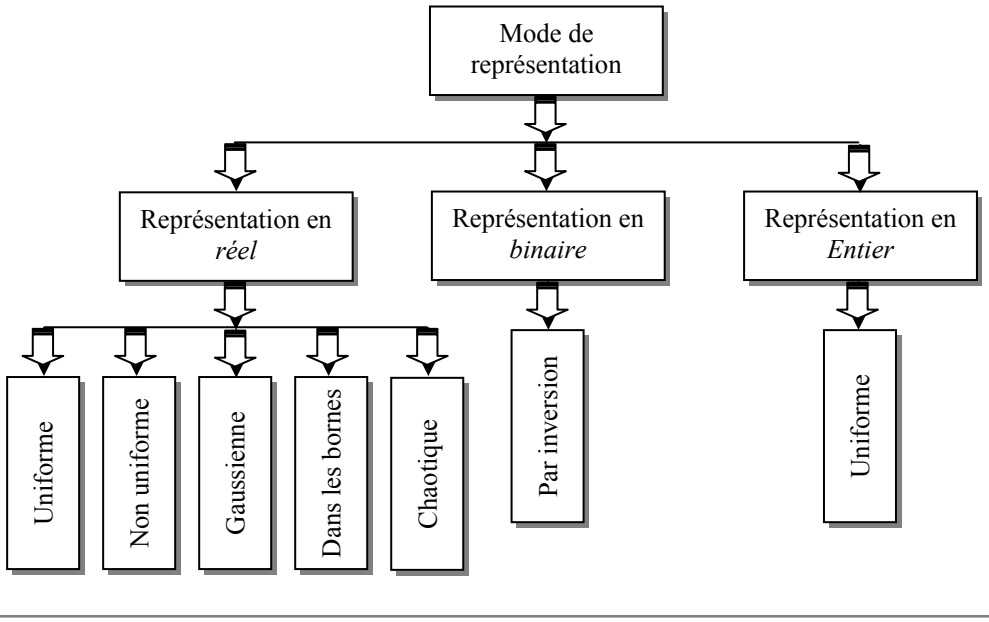


Fig. 1.24 : Différents modes de codage des chromosomes & leurs opérateurs de mutation appropriés.

b.1. Mutation uniforme

La mutation uniforme est identique à celle du codage binaire, c'est-à-dire, inversion de bit. Ainsi, chaque variable $\vartheta_i \in V$ est changée selon une certaine probabilité en un nombre aléatoire tiré dans une distribution uniforme dans l'intervalle $\left[\vartheta_i^{\min}, \vartheta_i^{\max}\right]$, avec $\vartheta_i^{\min}$ et $\vartheta_i^{\max}$ sont les bornes inférieure et supérieure de ϑ_i, respectivement.

b.2. Mutation non uniforme

La mutation non uniforme (non linéaire) revient à changer la variable ϑ_i en un nombre tiré dans une distribution non uniforme [81]. Cette nouvelle variable $\tilde{\vartheta}_i$ est telle que :

$$\tilde{\vartheta}_i = \begin{cases} \vartheta_i + \left(\vartheta_i^{\max} - \vartheta_i\right) \cdot f\left(gen\right) & \text{si} \quad \delta < 0.5 \\ \vartheta_i - \left(\vartheta_i + \vartheta_i^{\min}\right) \cdot f\left(gen\right) & \text{si} \quad \delta \geq 0.5 \end{cases} \tag{1.72}$$

$$f(gen) = \tilde{\delta} \cdot \left(1 - \frac{gen}{Max\_gen}\right)^{\tau} \tag{1.73}$$

δ & $\tilde{\delta}$: Nombres aléatoires $\in [0,1]$.
τ : Un paramètre déterminant le degré de non uniformité.

b.3. Mutation dans les bornes 'boundary mutation'

Avec cet opérateur, chaque variable $\vartheta_i \in V$, choisie pour la mutation, prend pour valeur l'une des 2 bornes $\vartheta_i^{\min}$ ou $\vartheta_i^{\max}$ avec équiprobabilité.

$$\tilde{\vartheta}_i = \begin{cases} \vartheta_i^{\min} & \text{si} \quad r < 0.5 \\ \vartheta_i^{\max} & \text{si} \quad r \geq 0.5 \end{cases} \tag{1.74}$$

r est un nombre aléatoire $\in [0,1]$. Il faut noter que cet opérateur n'a d'intérêt et d'efficacité que si la solution est proche des bornes de l'espace de recherche.

b.4. Mutation gaussienne [92]

Tous les éléments du chromosome $Ch_{\vartheta}^{(gen)}$ sont sujets de l'opération de mutation. Etant donné $Ch_{\vartheta}^{(gen)} = (\vartheta_1, ..., \vartheta_K, ..., \vartheta_m)$, le chromosome résultant est $Ch_{\vartheta}^{(gen+1)} = (\tilde{\vartheta}_1, ..., \tilde{\vartheta}_K, ..., \tilde{\vartheta}_m)$ avec :

$$\tilde{\vartheta}_k = \vartheta_k + f_k \qquad k = 1, ..., m \tag{1.75}$$

où f_k est un nombre aléatoire qui a une distribution gaussienne à moyenne nulle et une variance σ_k. Parfois, une variance adaptative est appliquée avec la forme [82] :

$$\sigma_k = \frac{Max\_gen - gen}{Max\_gen} \cdot \frac{\left(\vartheta_k^{\max} - \vartheta_k^{\min}\right)}{3} \tag{1.76}$$

b.5. Mutation chaotique

La mutation chaotique est introduite comme alternative pour maintenir une diversité dans la population durant le processus de l'évolution de l'AG [93]. L'opérateur de mutation chaotique est défini par :

$$\tilde{\vartheta}_K = \vartheta_K^{\min} + Mut(\vartheta_K) \cdot \left(\vartheta_K^{\max} - \vartheta_K^{\min}\right) \tag{1.77}$$

où $Mut(\cdot)$ est la fonction de mutation chaotique, définie par la fonction logistique donnée par :

$$Mut(\vartheta_K) = A \cdot \vartheta_K \cdot (1 - \vartheta_K) \tag{1.78}$$

$A \in [3.56, 4]$ est un paramètre de contrôle ou l'attracteur chaotique. L'efficacité de cette forme de mutation a été provée [93].

1.3.6. Amélioration des performances de l'AG standard

L'efficacité d'un algorithme évolutionnaire, en particulier un AG, dépend principalement du mode de représentation des chromosomes et des opérateurs génétiques adaptés à ses modes. Toutefois, le choix des paramètres tels que la probabilité de mutation P_m, la probabilité de croisement P_c, la taille de la population, ...etc., est un élément clé dans le processus de configuration d'un algorithme évolutif [94]. Plusieurs améliorations ont été proposées et introduites sur l'AG standard dont l'objectif est d'assurer la convergence de l'algorithme vers un optimum global en un temps optimal.

- Afin d'améliorer les performances des algorithmes de recherches, de nombreuses méthodes d'hybridation ont été apportées. Les techniques hybrides représentent une façon d'utiliser l'information auxiliaire pour accélérer le processus d'exploration génétique d'un côté. De l'autre côté, elles permettent d'exploiter, efficacement, les connaissances sur la procédure conventionnelle d'optimisation et l'efficacité de l'AG dans un sens coopératif [80] & [95]. Ceci améliore la convergence.
- La sélection des meilleurs individus pour la reproduction et la diversité dans la population sont 2 facteurs recommandés pour le bon déroulement de l'exécution de l'AG. Ces 2 dernières sont liées, directement, à la taille de la population et aux valeurs de P_m et P_c. Grenfenstette a proposé l'idée d'utiliser un AG de taille réduite $(\mu - GA)$ pour la supervision d'un AG maître [96]. D'où la notion d'*auto-régulation* avec l'idée de *contrôle auto-intégré* des paramètres de l'AG par un mico-GA. Michalwics a mentionné un AG avec des tailles variables des populations, similaire à l'évolution naturelle. Il a introduit le concept de l'âge du chromosome qui est équivalent au nombre de génération où le chromosome reste vivant, comme base de sélection [81]. T-Bâck a étudié l'effet d'adaptation des paramètres de l'AG sur son bon déroulement. Il a aussi mentionné une relation permettant d'avoir des paramètres optimaux de l'AG, donnée par [97] :

$$\ln(Max\_Pop) + 0.93 \cdot \ln(P_m) + 0.456 \cdot \ln(L\_Ch_g) = 0.56 \tag{1.79}$$

où L_Ch_g est la longueur du chromosome. L'expression (1.79) peut être approximée par :

$$Max\_Pop \cdot P_m \cdot \sqrt{L\_Ch_g} = 1.7 \tag{1.80}$$

Srinivas et al. ont proposé des probabilités de mutation et de croisement adaptatives, données par [98] :

$$P_c = \begin{cases} K_1 \cdot \dfrac{(F_{\max} - F_c)}{(F_{\max} - \overline{F})} & F_c \geq \overline{F} \\ K_3 & F_c < \overline{F} \end{cases} \tag{1.81}$$

$$P_m = \begin{cases} K_2 \cdot \dfrac{(F_{\max} - F_m)}{(F_{\max} - \overline{F})} & F_m \geq \overline{F} \\ K_4 & F_m < \overline{F} \end{cases} \tag{1.82}$$

où K_i, $(i = 1, ..., 4)$ sont des nombres aléatoires $\in [0, 1]$. $F_{\max}$ et $\overline{F}$ représentent le maximum et la moyenne de la population courante, respectivement. F_c est la plus grande valeur des 2 aptitudes associées aux 2 chromosomes candidats à l'opération de croisement. F_m est l'aptitude des solutions qui peuvent être subit à l'opération de mutation. Dans cette méthode d'adaptation, un problème peut être apparu lorsque la solution optimale avec l'aptitude maximale se produit, c'est-à-dire P_c et P_m restent toujours nulles. Pour remédier à ce problème, Chang-Kuo Chen et al. ont proposé des améliorations des lois d'adaptation des probabilités P_c et P_m [99] :

$$P_c = \begin{cases} K_1 \cdot \dfrac{(F'_{\max} - F'_c)}{(F'_{\max} - \overline{F'} + \delta)} \cdot a + P_{c\_last} \cdot (1 - a) & F'_c \geq \overline{F'} \\ K_3 \cdot a + P_{c\_last} \cdot (1 - a) & F'_c < \overline{F'} \end{cases} \tag{1.83}$$

$$P_m = \begin{cases} K_2 \cdot \dfrac{(F'_{\max} - F'_m)}{(F'_{\max} - \overline{F'} + \delta)} \cdot a + P_{m\_last} \cdot (1 - a) & F'_m \geq \overline{F'} \\ K_4 \cdot a + P_{m\_last} \cdot (1 - a) & F'_m < \overline{F'} \end{cases} \tag{1.84}$$

P_{c_last} et P_{m_last} sont les probabilités de croisement et de mutation dans la dernière génération, respectivement. a est un facteur de pondération 'weight of probability' et δ est une petite valeur constante telles que les premières valeurs de (1.83) et de (1.84) sont bien définies. F' dénote l'aptitude modifiée, donnée par la relation :

$$F'(ch) = \frac{F(ch) - F_{\min}}{F_{\max} - F_{\min}} \tag{1.85}$$

$F_{\min}$ et $F_{\max}$ représentent, respectivement, l'aptitude minimale et maximale dans la génération en cours.

Le taux de mutation commencera avec une valeur élevée pour produire une large diversité au début de l'exécution de l'AG, puis elle diminuera au cours de l'évolution et d'exploration génétique. Le taux de mutation pour cette opération sera donné par K.C. Tan et al. [100] :

$$P_m(gen) = \begin{cases} \alpha_1 \left[1 - \left(\dfrac{gen}{Max\_gen} \right)^2 \right] + \alpha_2 & 0 \leq gen \leq \alpha_3 \\ \alpha_1 \left[0.1 \left(\dfrac{(gen - Max\_gen)}{Max\_gen} \right)^2 \right] + \alpha_2 & \alpha_3 \leq gen \leq Max\_gen \end{cases} \tag{1.86}$$

avec $\alpha_1 \in [0.5,\ 0.8]$, $\alpha_2 = 1/L\_Ch_g$ & $\alpha_3 = Max\_gen/4$.

De plus, pour accélérer le processus de convergence, les probabilités de croisement P_c et de mutation P_m sont dynamiques et évoluent d'une génération à l'autre selon Teo Lian Seng et al. [101]. La figure 1.25 indique, clairement, le processus de l'évolution des opérateurs génétiques : croisement et mutation. Les probabilités de croisement et de mutation sont définies par les expressions suivantes :

$$P_c(gen) = \exp\left(-\frac{gen}{Max\_gen}\right) \tag{1.87}$$

$$P_m(gen) = \exp\left(0.05 \cdot \frac{gen}{Max\_gen}\right) - 1 \tag{1.88}$$

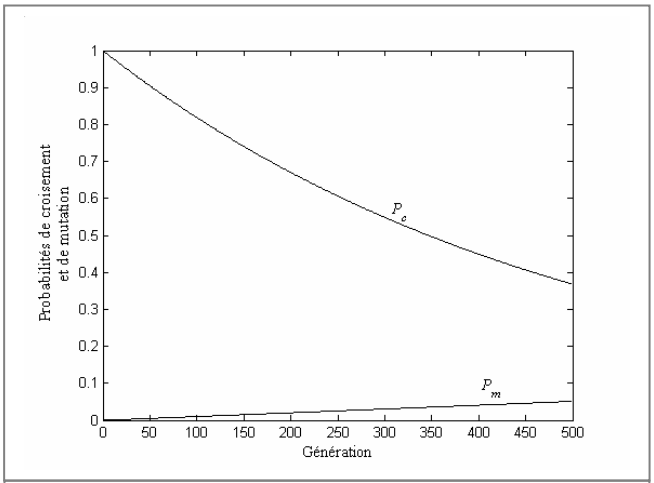


Fig. 1.25 : Evolution des probabilités de croisement et de mutation.

Un système flou a été utilisé pour l'ajustement automatique des paramètres de l'AG [102] - [104]. Les probabilités de croisement et de mutation sont ajustées d'une manière dynamique par la formule :

$$P_c(gen+1) = P_c(gen) + \Delta c(gen) \qquad \text{avec} \quad P_c(0) = 0.5 \tag{1.89}$$

$$P_m(gen+1) = P_m(gen) + \Delta m(gen) \qquad \text{avec} \quad P_m(0) = 0.0 \tag{1.90}$$

où $\Delta c(gen)$ et $\Delta m(gen)$ sont les variations des probabilités de croisement et de mutation, respectivement, générées par un SIF.

Zhiming Liu et al. ont proposé un AG adaptatif, de sorte que les probabilités de sélection P_s, de croisement P_c et de mutation P_m sont adaptatives en fonction du classement des chromosomes dans la population [105]. Après l'opération de croisement et de mutation, on aura $2 \times Max\_Pop$ solutions (parents et leurs enfants). Ces solutions sont rangées d'une manière croissante en fonction de leurs

aptitudes (compétition). Soit $k = \{0, 1, ..., 2 \cdot Max\_Pop\}$ est le rang des individus, la probabilité de sélection de k^{eme} solution sera donnée par :

$$P_s(k) = \begin{cases} 1 - \dfrac{1}{Max\_Pop - 1} \cdot k & 0 \le k \le Max\_Pop - 1 \\ \dfrac{1}{Max\_Pop - 1} \cdot (k - Max\_Pop) & Max\_Pop \le k \le 2 \cdot Max\_Pop - 1 \end{cases} \quad (1.91)$$

Cette expression permet aux chromosomes de bonnes et de mauvaises aptitudes de bien participer à la sélection pour la reproduction. Les probabilités de croisement et de mutation sont celles données par les expressions (1.81) & (1.82).

Une autre façon d'adapter les probabilités de croisement et de mutation dépendant du rangement des chromosomes dans la population s'intitule comme suit :

$$P_c = \begin{cases} \dfrac{\tilde{k}}{rang} & 0 \le \tilde{k} \le rang \\ 1.0 & rang \le \tilde{k} \le Max\_Pop - 1 \end{cases} \quad (1.92)$$

$$P_m = \begin{cases} \dfrac{\tilde{k}}{2 \cdot rang} & 0 \le \tilde{k} \le rang \\ 0.5 & rang \le \tilde{k} \le Max\_Pop - 1 \end{cases} \quad (1.93)$$

où $rang$ est le nombre de classement de la solution dans la population dont l'aptitude est plus étroite à la moyenne de la population. $\tilde{k}$ est le nombre de rangement de la solution ayant la plus grande aptitude des 2 individus sélectionnés pour l'opération de croisement.

Li-Min Su et al. ont proposé une méthode adaptative d'ajustement des paramètres de l'AG [106]. Ces valeurs sont adaptées en fonction de la fonction objective :

$$P_c = \begin{cases} k_c - (k_c - k_{c0}) \cdot \dfrac{(F_{\max} - F_c)}{(F_{\max} - F)} & F_c \ge F \\ k_c & F_c < F \end{cases} \quad (1.94)$$

$$P_m = \begin{cases} k_m - (k_m - k_{m0}) \cdot \dfrac{(F_{\max} - F_c)}{(F_{\max} - F)} & F_m \ge F \\ k_m & F_m < F \end{cases} \quad (1.95)$$

avec $[k_c \quad k_{c0} \quad k_m \quad k_{m0}] = [0.95 \quad 0.6 \quad 0.15 \quad 0.001]$. F_c est la plus grande valeur d'aptitude entre 2 individus sélectionnés à l'opération de croisement. F_m est la valeur d'aptitude de l'individu subi à l'opération de mutation. $F_{\max}$ est la plus grande aptitude dans la population et F est l'aptitude moyenne.

Hanan A. Kamal a présenté un nouvel opérateur adaptatif de mutation qui fournit de nouveaux éléments de solution et maintient en même temps les meilleurs schémas dans l'ancienne population [107].

- L'élitisme est un mécanisme optionnel particulier. Il permet de garder l'individu(s) le(s) mieux adapté(s) d'une génération à la suivante. Conserver ces solutions pour les générations futures permet d'améliorer les performances des algorithmes sur certains problèmes. Les opérateurs génétiques avancés permettent d'avoir une diversité dans l'espace de recherche, d'une part et d'assurer une convergence, d'autre part. La dominance, la diploïde, l'opérateur de réarrangement, ...etc., sont des exemples de ces opérateurs [80]. Parmi les stratégies d'élitismes, on peut y inclure les méthodes de sélections finales. Une fois la reproduction est terminée, au lieu de remplacer les descendants dans la prochaine population et pour aider l'AG de se converger, plusieurs méthodes sont proposées pour construire la nouvelle génération. La *sélection à l'état permanent* '*steady-state selection*', *sélection par compétition,* s*élection par élevage sélective* '*selective breeding selection'* sont des techniques compétitives qui permettent aux meilleurs individus de participer à la construction de la nouvelle génération.
- Pour éviter une convergence prématurée due à un trop grand nombre de copies des individus 'extraordinaires', la solution proposée est de changer, dynamiquement, l'échelle de la fonction d'évaluation ou d'objective. Cela est fait afin de garder un niveau de compétitivité adéquat tout au long de l'exécution de l'AG [80].
- Une version des AG proposée par Xuefeng F. Yan appelée AG-chaotique (CGA) exploite les propriétés des systèmes chaotiques [108]. Une autre version appelée AG à base d'une recherche chaotique (CSGA) proposée par Gwo-Ching Liao, est utilisée pour la conception des modèles neuronaux flous [109]. La fonction logistique est un outil performant pour la conception des opérateurs génétiques chaotiques [110] - [111].

Ces mécanismes d'adaptation et de recherche des paramètres optimaux de l'AG permettent de maintenir une large diversité dans la population et d'empêcher, donc, la convergence prématurée dans l'exploration et de la recherche génétique [112]. Ces techniques et stratégies devront, aussi, conduire à de nouvelles améliorations de l'efficacité et de l'applicabilité des AG.

1.3.7. Conclusions

Les AG sont des méthodes de recherche stochastique '*très flexibles*' basées sur des abstractions des processus d'évolution naturelle. Elles constituent des outils d'optimisation efficaces permettant de résoudre toute sorte de problème, de l'optimisation des fonctions au modélisation et contrôle des procédés industriels [99] & [113] - [114], tout en choisissant une représentation des individus, des opérateurs et une fonction d'évaluation adéquate. Ces choix sont fortement dépendants du problème. L'aspect parallèle de fonctionnement des AG leur permet l'implémentation sur des architectures parallèles [88] - [90]. Néanmoins, leur aspect magique et leur potentiel ne doivent pas masquer de nombreuses limitations :

- Les outils théoriques disponibles pour effectuer les choix optimaux des paramètres structurels et fonctionnels des AG sont quasi-inexistants. Des propositions sont citées dans la littérature du domaine des AG. Au final, il est

nécessaire d'expérimenter de nombreuses combinaisons pour obtenir de *meilleurs* résultats et il est souvent difficile de justifier les choix effectués.

- Si les solutions trouvées sont généralement *bonnes*, il est rare que les AG trouvent le minimum global *souhaité* de manière certaine.

1.4. Conclusions

Dans ce chapitre, les fondations nécessaires à la maîtrise des TIA sont établies. Après avoir introduit des concepts de base concernant la structure générale et les différents types des modèles flous, nous avons étudié, plus particulièrement, les caractéristiques structurelles et paramétriques des ces modèles.

Les principaux avantages des techniques floues sont l'approche naturelle de la modélisation et la bonne interprétabilité de la description tout en employant des règles linguistiques. Cependant, comme il n'y a pas de méthode formelle pour déterminer ses paramètres, c'est-à-dire sa base de connaissances, il serait intéressant de disposer d'algorithmes permettant l'apprentissage automatique de ces paramètres.

Dans les systèmes experts, les connaissances de l'expert ont une forme énumérée. Elles sont exprimées sous forme de règles linguistiques. Dans le cas des modèles connexionnistes 'neuronales', les connaissances ont une forme distribuée. Elle sont codées dans les poids synaptiques des connexions.

La topologie du réseau, les fonctions d'activation des neurones, le seuil de ces fonctions et l'algorithme d'apprentissage utilisé sont des facteurs significatifs dans l'emploi des modèles connexionnistes.

Nous avons aussi exposé en détail les différentes étapes nécessaires à l'exécution d'un AG, configuration paramétriques, initialisation de la population, évaluation, sélection pour la reproduction, test d'arrêt et les améliorations de l'AG de base pour l'aider à se converger vers une solution optimale.

Les algorithmes évolutionnistes, grâce à la simplicité et la souplesse de leurs principes, peuvent être un outil d'optimisation et de conception des systèmes de contrôle des processus complexes dont la dynamique n'est pas encore maîtrisée. Leur souplesse leur permet d'être utilisés à plusieurs niveaux, depuis l'optimisation de paramètres pour un contrôleur dont la structure a été prédéfinie, jusqu'à l'étude ou même la conception complète d'un système de contrôle.

Le parallélisme intrinsèque de traitement des informations 'données' représente un point commun entre les différentes TIA. Ce paradigme constitue un facteur primordial dans des problèmes réels dans des domaines aussi variés que la recherche d'information, l'aide à la décision, la robotique, l'identification, le contrôle de systèmes complexes, ...etc.

Enfin, une caractéristique importante des TIA apparaît, implicitement, dans cette partie de la thèse : La facilité d'utilisation/implantation par des non spécialistes '*non mathématiciens*'. Cette caractéristique permet aux chercheurs dans différents domaines de la science d'utiliser des méthodes puissantes d'approximation, d'optimisation, ...etc., sans passer beaucoup de temps à comprendre ces méthodes et leurs conditions d'application.

Chapitre 2

Techniques Intelligentes : Vers une approche hybride

Résumé :

Dans ce chapitre, un état de l'art bien détaillé des méthodes de modélisation des SIF est effectué. Nous avons commencé par les différentes techniques d'optimisation des SIF par les AG. Ensuite, l'hybridation des RNF, avec différentes structures et algorithmes d'apprentissage proposés dans la littérature, est aussi synthétisée. Le chapitre se termine par une conclusion.

« You must understand the process before you can control it & the simplest control system that will do the job is the best. »

William L. Luyben (1990)

Sommaire

2. Techniques Intelligentes : Vers une approche hybride

2.1. Introduction

Les systèmes flous ont connu un succès considérable dans la modélisation et la commande des procédés industriels complexes, du fait de leurs caractères approximatif et qualitatif inspirés de la pensé humaine. Cependant, leurs performances sont liées à 2 facteurs importants :

- La disponibilité de l'expertise, c'est-à-dire, le savoir-faire de l'opérateur humain.
- La validité des techniques d'acquisition des connaissances et la justesse des données acquises.

La conception d'une base des règles floues est le processus qui conduit à la formalisation, sous forme de règles et/ou des relations apprises, à partir d'un ensemble d'exemples qui existe entre les entrées et les sorties d'un procédé. Les principales méthodes décrites dans la littérature sont présentées dans cette partie de la thèse. Ces méthodes sont, principalement, pilotées et évaluées par rapport à un critère de performances (mécanisme d'adaptation).

Dans beaucoup de cas, la structure est déterminée, empiriquement, en choisissant à priori le type de raisonnement approximatif linguistique ou relationnel, le nombre des sous-ensembles flous pour chaque variable d'entrée et en prenant toutes les combinaisons possibles pour construire la base de règles.

Le réglage par essais successifs de ces nombreux paramètres étant assez long et fastidieux. Diverses techniques d'auto - réglage, d'optimisation et d'apprentissage ont été développées ces dernières années. On peut citer, les techniques de la programmation mathématique, les réseaux connexionnistes et les algorithmes évolutionnaires. Ces techniques permettent à la fois l'extraction automatique des connaissances sous forme de règles floues à partir d'un ensemble de données numériques.

Les méthodes hybrides intègrent toutes sortes d'outils, dont les plus utilisés sont les RNA et les AE. Les AE sont susceptibles de trouver un optimum global et permettent d'optimiser la structure et les paramètres d'un SIF. Les RNA ont apporté aux SIF leurs algorithmes d'apprentissage et leur précision dans l'ajustement numérique. Ce type d'outils peut être appréciable lorsque la sémantique est secondaire par rapport aux performances numériques et pour la gestion des problèmes pour lesquelles aucune connaissance experte n'est disponible.

L'hybridation des techniques intelligentes dans un sens coopératif permet d'exploiter les caractéristiques de chacune pour accomplir une tâche performante et efficace dans la modélisation et la commande des systèmes non linéaires. Elle peut, aussi, fournir des solutions bien adaptées à la complexité de ces systèmes.

2.2. Outils de conception : Etat de l'art

Les SIF sont des approximateurs universels des fonctions. Pour toute fonction continue $f(x)$, généralement non linéaire, définie sur un ensemble compact Ω et pour toute constante positive ε, il existe un SIF $S_{IF}(x)$ tel que : $\sup_{x\in\Omega}|f(x)-S_{IF}(x)|<\varepsilon$

La propriété d'approximation universelle ne donne pas une méthode de conception 'construction' du SIF. Elle garantie seulement son existence. La modélisation de ces systèmes n'était pas toujours évidente. C'est pourquoi le flou adaptatif a fait son apparition. Les lois d'adaptation, proposées dans la littérature, agissent sur le comportement global du SIF : Sa configuration structurelle et paramétrique. La conception d'un SIF nécessite le passage par les étapes suivantes :

- Définir le type de raisonnement utilisé : relationnel ou linguistique.
- Sélection des univers de discours des variables linguistiques.
- Adoption d'une stratégie de fuzzification appropriée, tout en incluant :
  - √ Partitionnement de l'espace d'entrée/sortie.
  - √ Sélection des fonctions d'appartenance utilisées.
  - √ Mise en échelle des variables d'entrée/sortie (normalisation & dénormalisation).
- Construction de la base des règles :
  - √ Sélection des variables d'entrée et de commande.
  - √ Etablir la matrice d'inférence.
- Choix de la logique de prise de décision :
  - √ Définition des implications.
  - √ Interprétation des connexions des règles : Conjonction et disjonction.
  - √ Mécanismes d'inférence.
  - √ Stratégie de défuzzification.

Il existe 2 approches pour la conception des SIF. La première, dite traditionnelle, est basée sur le savoir faire de l'opérateur humain [11] & [115]. La deuxième est basée sur la notion d'extraction automatique des connaissances (adaptation) à partir des données numériques par application des techniques d'optimisation et d'apprentissage (auto - apprentissage 'self - learning' ou auto - réglage 'self - tuning') : SOFLC.

La figure 2.1 schématise la structure d'une boucle de contrôle à base d'un SIF à auto - apprentissage (SIF adaptatif). Deux niveaux hiérarchiques constituent le SOFLC : Un niveau d'apprentissage et l'autre de décision ou d'action.

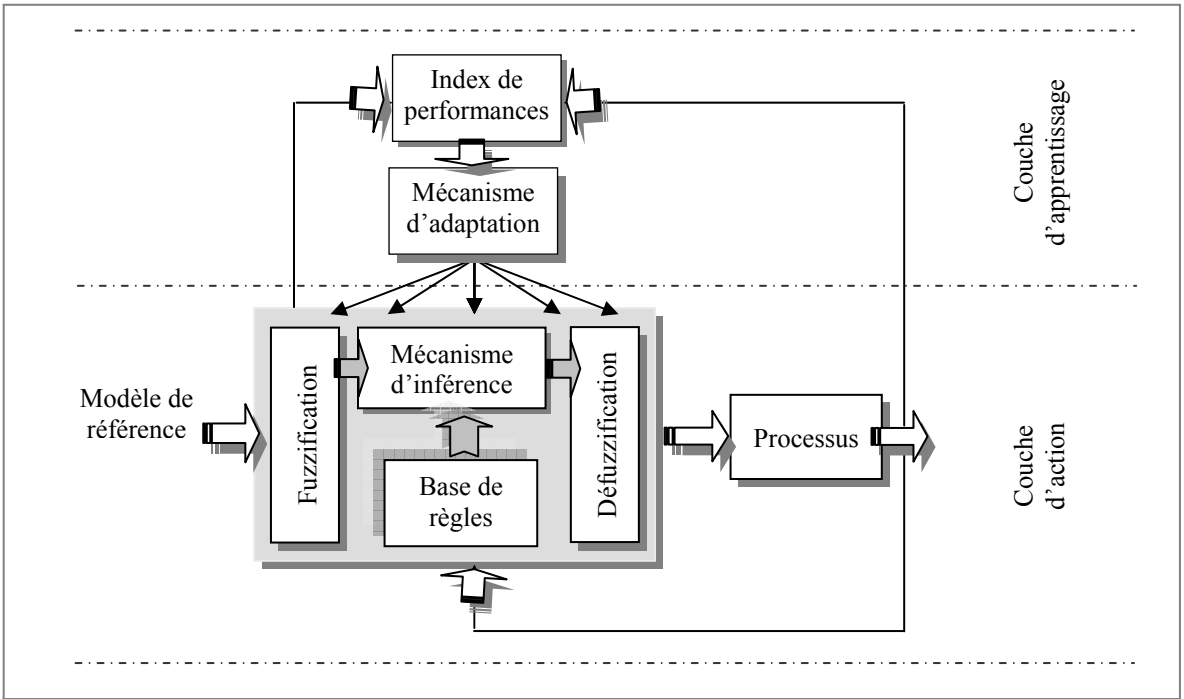


Fig. 2.1 : Niveaux hiérarchiques d'un SOFLC.

Le classement des méthodes d'optimisation s'effectue suivant le nombre de dérivées de la fonction coût $J(\vartheta)$ intervenant dans l'élaboration de l'algorithme de calcul. Sans perte de généralité, nous pouvons toujours considérer que nous cherchons un minimum :

$$\min_{\vartheta} J(\vartheta) \tag{2.1}$$

Les méthodes du premier ordre font appel au gradient de la fonction $\partial J(\vartheta)/\partial \vartheta$. Les méthodes du deuxième ordre font appel, en plus du gradient de la fonction, à son Hessien $\partial^2 J(\vartheta)/\partial \vartheta \partial \vartheta^T$. La loi d'adaptation est donnée par l'expression (1.36).

Enfin, nous pouvons encore citer les méthodes d'ordre zéro, qui ne font appel ni au gradient, ni au Hessien de la fonction coût, mais uniquement aux valeurs prises par cette fonction de coût [12].

Les méthodes de premier et du deuxième ordre sont très efficaces, lorsque la fonction coût a la forme d'un paraboloïde, ou au moins, elle est unimodale. Lorsque la fonction coût n'est pas unimodale, le risque de tendre vers un minimum local est loin d'être négligeable lorsque nous utilisons ces méthodes.

L'avantage des méthodes d'ordre zéro, outre le fait qu'elles ne nécessitent pas la connaissance des dérivées de la fonction, réside dans le fait qu'elles convergent toujours vers l'*optimum optimorum* [12]. Quatre grandes familles de méthodes d'ordre zéro peuvent être citées :

- Les méthodes de programmation linéaire.
- Les recuits simulés [116].
- La recherche avec tabous (Search Tabou) [117] - [118].
- Les algorithmes évolutionnaires.

Les différents niveaux d'optimisation possibles des SIF sont, habituellement, classés en 3 catégories [119] - [120] :

- Optimisation structurelle.
- Optimisation paramétrique.
- Optimisation globale.

L'optimisation structurelle permet de déterminer le nombre optimal de règles, le nombre des sous ensemble flous pour les variables d'entrées/sorties, les opérateurs d'inférence et la méthode de défuzzification. On suppose que la structure est bien définie. Le reste consiste à optimiser les différents paramètres du SIF. La figure 2.2 présente les différentes approches proposées dans la littérature.

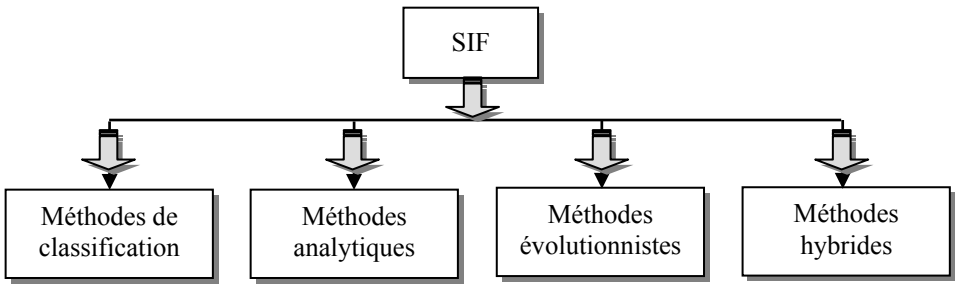


Fig. 2.2 : Méthodes d'optimisation et d'apprentissage d'un SIF.

L'un des premiers algorithmes proposés pour construire, automatiquement, une partition floue d'un ensemble de vecteurs de $\Re^d$ est l'algorithme FCM ou l'algorithme des centres mobiles flous, introduit par Bezdek [121]. Les algorithmes de groupage '*clustering*' flou sont utilisés pour organiser un ensemble de données. Leur application conduit à une partition des données en groupes homogènes suivant un certain critère. Le partitionnement de l'espace est dérivé de la partition des données et conduit à une règle par groupe dans le cas général. Donc les ensembles flous résultants sont propres à une règle, ils ne sont pas partagés par l'ensemble des règles. Les méthodes de groupage (classification) sont largement utilisées à la fois pour l'apprentissage supervisé ou non supervisé des SIF [119] & [122].

Les méthodes analytiques sont basées sur des méthodes d'optimisation du premier et de deuxième ordre (descente du gradient et la méthode des MCR) [123] - [124]. Ces techniques sont applicables quand les fonctions coût (mesure des performances) considérées sont dérivables.

Considérons le SIF du type TS en fonction du vecteur d'entrée $\underline{x} = [x_1, x_2, ..., x_m]$, de la i^{eme} sortie b_i et les sous ensembles flous $\left(A_1^i, A_2^i, ..., A_m^i\right)$ associés aux variables d'entrées. La forme de la i^{eme} règle est donnée par :

$$R^i : \textbf{ SI } \left(x_1 \textbf{ est } A_1^i\right) \textbf{... ET } \left(x_m \textbf{ est } A_m^i\right) \textbf{ ALORS } b_i = g_i(x_1, x_2, ..., x_m) \tag{2.2}$$

Le choix de la fonction $g_i(\cdot)$ dépend de l'application en cours. Le modèle linéaire de TS sera donné par la formule :

$$g_i(x_1, x_2, ..., x_m) = a_{i0} + a_{i1} \cdot x_1 + a_{i2} \cdot x_2 + ... + a_{im} \cdot x_m \tag{2.3}$$

La sortie du SIF sera calculée en appliquant la méthode du centre de gravité, donnée par l'expression :

$$y = \frac{\sum_{i=1}^{N} a_{i0} \cdot \mu_i(\underline{x})}{\sum_{i=1}^{N} \mu_i(\underline{x})} + \frac{\sum_{i=1}^{N} a_{i1} \cdot \mu_i(\underline{x})}{\sum_{i=1}^{N} \mu_i(\underline{x})} + ... + \frac{\sum_{i=1}^{N} a_{im} \cdot \mu_i(\underline{x})}{\sum_{i=1}^{N} \mu_i(\underline{x})} \tag{2.4}$$

La méthode des MCR est souvent utilisée pour l'adaptation des paramètres des conséquences. Supposons que :

$$\xi_i(\underline{x}) = \frac{\mu_i(\underline{x})}{\sum_{i=1}^{N} \mu_i(\underline{x})} \tag{2.5}$$

$$\xi(\underline{x}) = [\xi_1(\underline{x}), ..., \xi_N(\underline{x}), x_1 \cdot \xi_1(\underline{x}), ..., x_1 \cdot \xi_N(\underline{x}), x_m \cdot \xi_1(\underline{x}), ..., x_m \cdot \xi_N(\underline{x})]^T \tag{2.6}$$

$$\theta = [a_{10}, ..., a_{N0}, a_{11}, ..., a_{N1}, ..., a_{1m}, ..., a_{Nm}]^T \tag{2.7}$$

De sorte que :

$$f(\underline{x} \mid \theta) = \theta^T \cdot \xi(\underline{x}) \tag{2.8}$$

La loi d'adaptation itérative du vecteur des paramètres θ à l'itération k est formulée par [125] :

$$\begin{aligned} P(k) &= \frac{1}{\lambda} \cdot \left(I - P(k-1) \cdot \xi^k \cdot \left(\lambda I + \left(\xi^k\right)^T \cdot P(k-1) \cdot \xi^k \right)^{-1} \cdot \left(\xi^k\right)^T \right) \cdot P(k-1) \\ \theta(k) &= \theta(k-1) + P(k) \cdot \xi^k \cdot \left(y^k - \left(\xi^k\right)^T \cdot \theta(k-1) \right) \end{aligned} \tag{2.9}$$

Le coefficient λ est le facteur d'oubli (forgetting factor). L'initialisation de la méthode s'effectue par un choix régulier de $P(0)$ et $\theta(0)$. Généralement, $\{P(0) = \alpha \cdot I\}$ et $\{\theta(0) = 0\}$. En absence d'informations initiales sur les paramètres à estimer, on choisi un gain d'adaptation α grand (la valeur typique est de l'ordre de 10000). Par contre si on dispose d'une estimation initiale des paramètres, on choisi un gain faible ($\alpha \leq 1$).

L'algorithme de la descente du gradient est, aussi, utilisé pour l'optimisation des paramètres des conséquences. Un apprentissage supervisé est visé pour la minimisation de la fonction coût $J(\cdot)$ donnée par :

$$J(k) = \frac{1}{M} \sum_{k=1}^{M} \left(y(k) - y^d(k) \right)^2 \tag{2.10}$$

$y(k)$ est la sortie du processus à commander et $y^d(k)$ est la sortie désirée à l'itération k. M est le nombre d'échantillons des données. De l'expression (2.8), la fonction coût (2.10) peut être réécrite sous la forme :

$$J = \theta^T \cdot R \cdot \theta - 2 \cdot \theta^T \cdot P + \frac{Y^T \cdot Y}{M} \tag{2.11}$$

avec $R = (1/M)\xi^T \xi$ est la matrice d'autocorrélation. $\xi_{M \times m}$ est donnée par l'expression (2.6) et $P = (1/M) \cdot \xi^T \cdot Y$ est le vecteur de corrélation (cross correlation) de dimension m et $Y = Y^d$ est le vecteur des sorties désirés. La dérivée de l'équation (2.11) donne :

$$\frac{\partial J}{\partial \theta} = -2 \cdot R \cdot \theta - 2 \cdot P \tag{2.12}$$

Les équations linéaires suivantes sont obtenues pour la minimisation de la fonction coût J :

$$R \cdot \theta = P \tag{2.13}$$

Dans la méthode de la descente du gradient, la direction des changements des paramètres sera dans le sens opposé du gradient de la fonction coût :

$$\Delta\theta(k) = -\frac{\partial J}{\partial \theta(k)} = 2 \cdot P - 2 \cdot R \cdot \theta \tag{2.14}$$

$$\theta(k+1) = \theta(k) + \eta_\theta \cdot \Delta\theta(k) \tag{2.15}$$

où η_θ est la vitesse d'apprentissage. La descente du gradient peut aussi être applicable pour l'optimisation des fonctions d'appartenance sous la condition que celles-ci soient dérivables [119] & [123].

Certains chercheurs ont introduit les méthodes de commande adaptatives directes et indirectes. Dans ces techniques, la propriété d'approximation universelle des SIF est exploitée [7] - [10] & [126] - [128]. Des travaux de recherche ont porté sur la commande floue par mode de glissement [129] - [131]. Le backstepping, méthode appliquée aux systèmes triangulaires avec des paramètres inconnus, est proposée comme méthode d'analyse et de synthèse des SIF [132] - [133].

La conception des SIF sur la base de l'analyse de la stabilité et de la robustesse de la boucle de commande est étudiée par l'approche énergétique de Lyapunov [134].

Pour plus d'informations et compléments algorithmiques, nous citons l'ouvrage de KAZUO TANAKA qui regroupe l'ensemble des travaux dans ce contexte [134]. La commande prédictive généralisée pour la détermination des paramètres du SIF du type TS dans certain cas est aussi abordée [135] - [137].

Les méthodes évolutionnistes recouvrent en particulier les AG, PSO et les stratégies d'évolution [138]. Elles permettent à la fois l'optimisation paramétrique et structurelle. La méthode des PSO représente une alternative d'optimisation des SIF. Cihan Karakuzu a proposé l'utilisation des PSO pour l'optimisation de l'ensemble des paramètres d'un SIF du type TS [139]. Les performances de l'algorithme sont testées sur 2 exemples de systèmes non linéaires : Un réacteur chimique et un oxcillateur de Van der Pol.

Les méthodes hybrides intègrent toutes sortes d'outils, dont les plus utilisés sont les RNA (approche connexionniste) et les AE. Les AE sont susceptibles de trouver un optimum global et permettent d'optimiser la structure et les paramètres d'un SIF.

Les méthodes d'ordre zéro sont, souvent, combinées avec celles du premier et du deuxième ordre. Une initialisation suivie par un apprentissage plus fin est la règle la plus utilisée dans la littérature [125] & [140] - [143]. Vincenzo Giordano et al. ont proposé une approche hybride pour la conception des SIF adaptatifs pour lesquels 2 algorithmes d'apprentissage avec différentes caractéristiques sont fusionnés pour l'amélioration des performances du SIF [144] - [145]. L'approche combine un AG, conçu pour optimiser l'ensemble structure/paramètres du SIF et une loi d'adaptation basée sur l'approche de Lyapunov permettant un ajustement plus fin des conséquences (pics) du modèle flou. Zne-Jung Lee a introduit une approche systématique à 2 niveaux hiérarchiques pour l'approximation des fonctions [146]. Une phase d'initialisation consiste en une identification de la structure et une estimation des paramètres du SIF par la méthode FCM. Dans la deuxième phase, un algorithme hybride AG-PSO est utilisé pour un réglage plus fin de l'ensemble des paramètres des prémisses et des conséquences obtenus dans la première phase.

Les RNA et les AG ont apportés aux SIF leurs structures, leurs algorithmes d'apprentissage et leurs précisions dans l'ajustement numérique. Ce type d'outils peut être appréciable lorsque la sémantique est secondaire par rapport aux performances numériques et pour la gestion des problèmes pour lesquelles aucune connaissance experte n'est disponible.

L'interaction entre les techniques intelligentes est schématisée par la figure 2.3. Plusieurs possibilités se présentent parmi lesquelles, on cite :

- Un SIF optimisé par les AG.
- Un RNA optimisé par les AG.
- Un modèle RNF.
- Une structure RNF à optimisation génétique.

Le reste du chapitre est consacré à une description détaillée de l'interaction SIF-AG et RNF, respectivement.

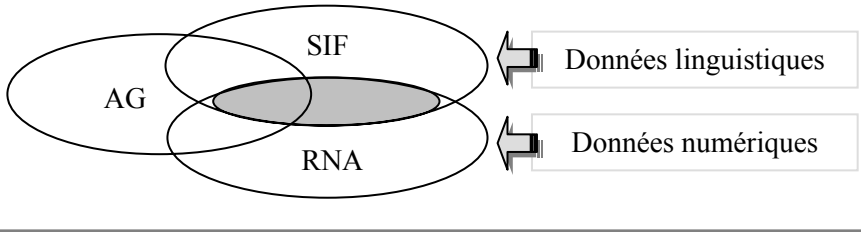


Fig. 2.3 : Interaction entre les différentes TIA.

2.2.1. Optimisation des SIF par les AG

L'utilisation des AG dans le contexte d'optimisation des modèles flous a pour objectif de trouver une structure avec paramètres convenables. Les éléments candidats à l'optimisation sont :

- Les fonctions d'appartenance.
- Les règles floues.
- Les opérateurs de connexion des règles (conjonction et disjonction), la procédure d'inférence et la méthode de défuzzification [20].

Les étapes principales de l'emploi des AG dans l'optimisation des SIF sont :

1. Choix d'une stratégie de codage appropriée, la taille de la population et l'initialisation des chromosomes.
2. Définition de la fonction objective.
3. Evaluation des individus de la population.
4. Sélection aléatoire des individus pour la reproduction en fonction de leur aptitude.
5. Création de nouveaux individus par application des opérateurs génétiques : croisement et mutation.
6. Aller à l'étape 3.

Ce processus se répète jusqu'à ce qu'un critère soit satisfait, généralement, un nombre maximal de génération.

L'application des AG pour faciliter l'extraction automatique d'un ensemble optimal des paramètres d'un SIF a été étudiée par plusieurs chercheurs depuis 1990. Sept catégories se sont découlées de ces recherches :

- Les règles d'inférence sont définies soit à partir de l'expérience, soit par une méthode telle que la méthode de plan de phase ou par la fonction génératrice des règles basée sur le gradient négatif d'une mesure de performance proposée par Zong-Mu Yeh. Les fonctions d'appartenance sont choisies, uniformément réparties, sur les univers de discours et l'AG est utilisé pour l'optimisation des facteurs d'échelles des variables d'entrées/sorties [147].
- Les règles d'inférence **SI - ALORS** sont, complètement, définies au préalable. Seules les fonctions d'appartenance associées aux espaces d'entrés/sorties sont

ajustées. Les travaux de Karr représentent une référence importante dans ce contexte [148] - [150].

- Les prémisses des règles et les fonctions d'appartenance sont fixées. Seules les conclusions des règles sont optimisées [151] - [154].
- Les conclusions des règles sont définies par l'opérateur humain et l'AG est utilisé pour l'optimisation de la partie prémisse des règles d'inférence floue [155].
- Les fonctions d'appartenance et les conclusions des règles sont optimisées, simultanément, par étapes successives. La première est l'optimisation des conclusions avec la condition que les fonctions d'appartenance des prémisses sont fixées. Un ajustement plus fin des fonctions d'appartenance des prémisses aura lieu dans la deuxième étape [156].
- Apprentissage simultané des conclusions et des prémisses des règles d'inférence. Les conditions sur les règles sont spécifiées à l'avance par l'utilisateur [157] - [160].
- La structure et les paramètres du SIF sont déterminés par l'AG. L'intervention de l'utilisateur est, pratiquement, négligeable [93] & [161] - [162].

En 2000 Juang a proposé la notion de l'évolution symbiotique d'un contrôleur flou (SEFC) comme méthode de conception inspirée du processus de l'évolution symbiotique dans la nature [163] - [164]. Chaque règle floue est codée dans un chromosome, en spécifiant les centres et les bases des fonctions d'appartenance correspondantes à cette règle. Chaque règle est un gène du chromosome. L'ensemble des règles constitue le chromosome dans la population de l'AG, comme l'indique la figure 2.4.

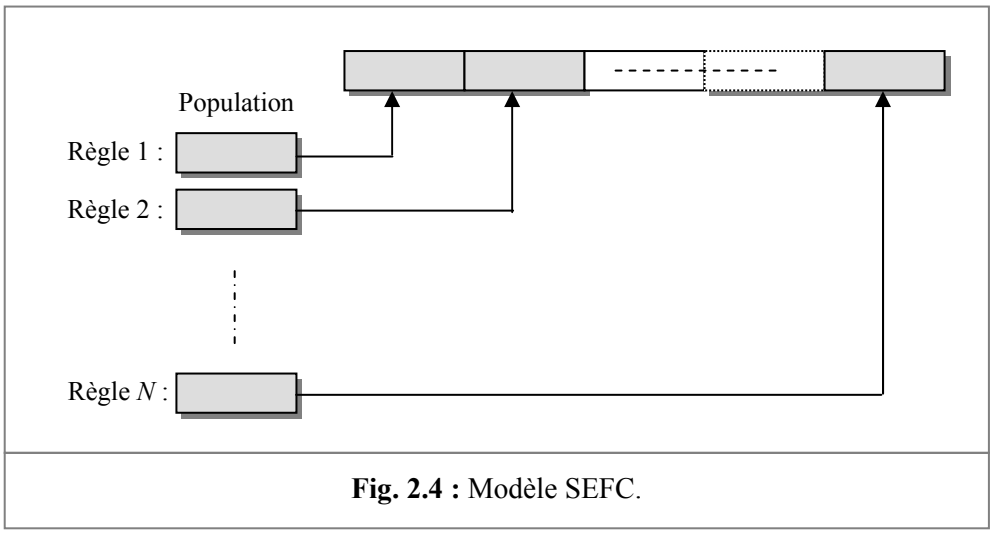


Fig. 2.4 : Modèle SEFC.

Les fonctions d'appartenance gaussiennes avec le raisonnement approximatif de TS ont été utilisées, par Chia-Feng Juang & M. Jamei et al. pour valider le modèle SEFC [165] - [166]. La figure 2.5a montre le codage d'une règle du type TS0, c'est à dire, de la forme :

$$\textbf{SI } x_1 \textbf{ est } \mu(m_1,\sigma_1) \textbf{ ET } x_2 \textbf{ est } \mu(m_2,\sigma_2) \textbf{ ET ... ET } x_m \textbf{ est } \mu(m_m,\sigma_m) \textbf{ ALORS } y \textbf{ est } w_0 \quad (2.16)$$

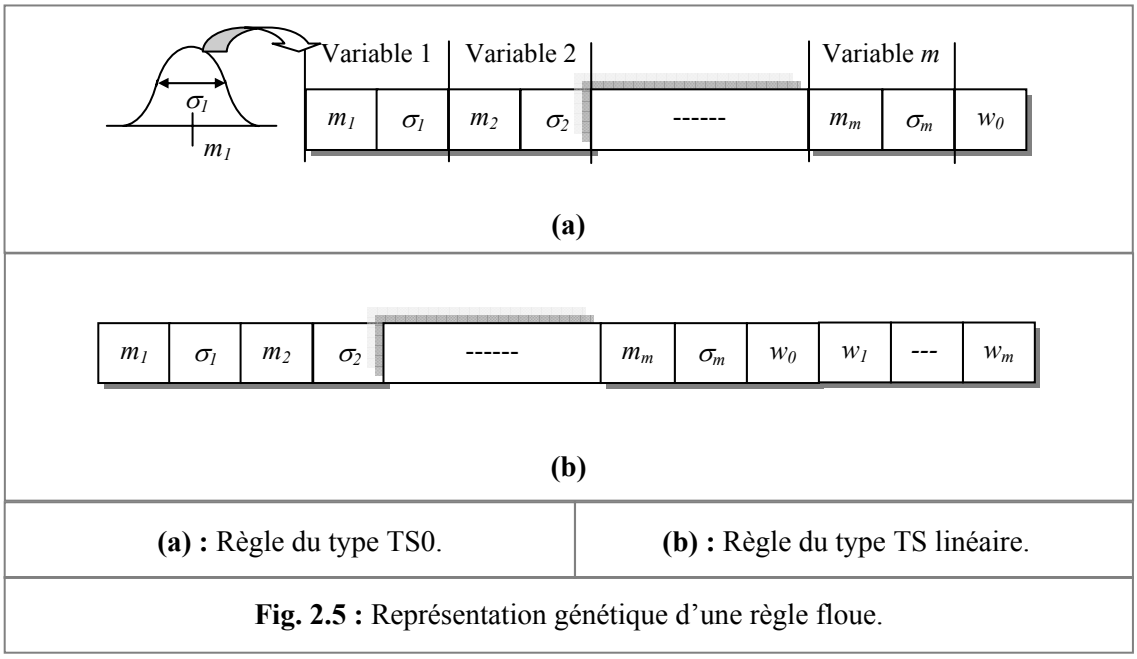
$\mu(m_i,\sigma_i)$ représente la fonction d'appartenance gaussienne de centre m_i et de variance σ_i^2. La figure 2.5b est une représentation d'une règle du type TS de la forme :

$$\textbf{SI } x_1 \textbf{ est } \mu(m_1,\sigma_1) \textbf{ ET } x_2 \textbf{ est } \mu(m_2,\sigma_2) \textbf{ ET ... ET } x_m \textbf{ est } \mu(m_m,\sigma_m) \quad \textbf{ALORS } y = w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + ... + w_m \cdot x_m \tag{2.17}$$

w_i, $(i = 0,1,...,m)$ sont les coefficients du modèle flou de TS.

L'évolution du processus d'apprentissage s'effectue en appliquant les opérateurs génétiques adaptés au type de codage tout au long d'un horizon spécifié, généralement, par un nombre maximal de générations.

La puissance d'un algorithme d'optimisation est mesurée par sa capacité de se converger vers un optimum global. Dans le cas des AG, les stratégies d'élitismes sont utilisées comme guide de convergence de l'algorithme.



(a) : Règle du type TS0. **(b) :** Règle du type TS linéaire.

Fig. 2.5 : Représentation génétique d'une règle floue.

2.2.2. RNF : Une approche hybride

L'approche connexionniste consiste à combiner la théorie des RNA avec celle de la LF, de manière à tirer profits des avantages de chacune de ces 2 techniques. L'objectif est de concevoir ce qu'on appelle les RNF.

Un RNF est un RNA qui est, topologiquement, équivalent à la structure d'un SIF. Les entrées/sorties du réseau ainsi que les poids sont des nombres réels. Par contre, les noeuds implémentent des opérations spécifiques aux systèmes flous : fuzzification, opérateurs flous (conjonction, disjonction) et défuzzification. En d'autres termes, un RNF peut être vu comme un système flou pour lequel les opérations sont implémentées de façon parallèle par un RNA. Il existe 2 manières pour construire des RNF [167] - [168] :

- Soit, en construisant un SIF sous forme d'un PMC dans lequel les poids du réseau correspondent aux paramètres du SIF (approche structurelle).
- Soit, en construisant un RNA qui peut incorporer le processus de raisonnement approximatif (approche fonctionnelle).

Une autre méthode neuro - floue est l'association en série ou en parallèle des RNA et des SIF. Leurs champs d'application est la classification, la reconnaissance des formes et l'aide à la décision [167].

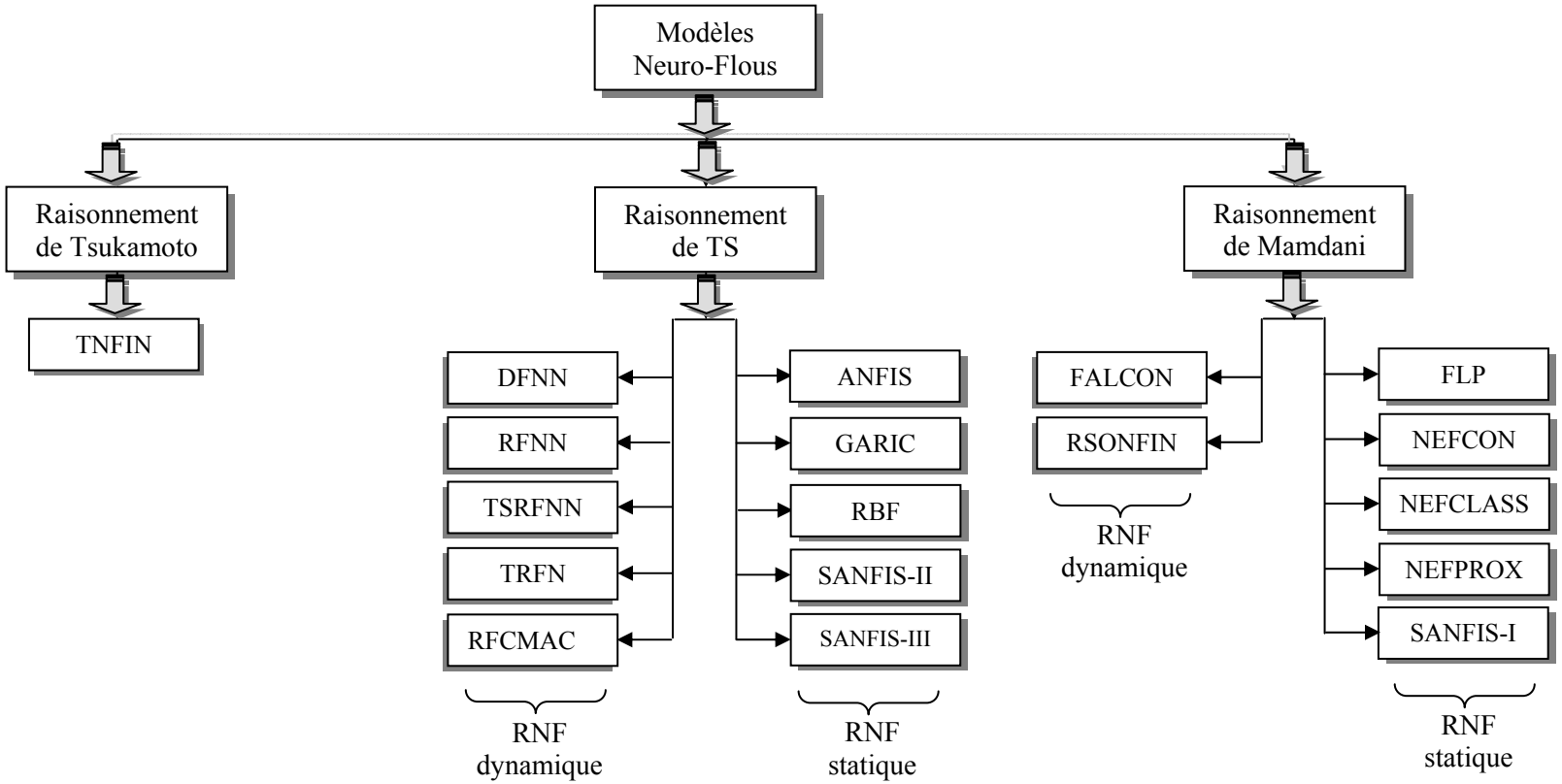


Fig. 2.6 : Arborescence de différentes structures RNF.

La principale propriété des RNF est leur capacité à traiter sur un même outil des connaissances numériques et symboliques d'un système. Ils permettent, donc, d'exploiter les capacités d'apprentissage des RNA, d'une part et les capacités de raisonnement de la LF, d'autre part. Diverses combinaisons des méthodes neuro - floues ont été développées depuis 1990. Elles sont le plus souvent orientées vers le contrôle des procédés complexes, l'approximation des fonctions (l'identification) et les problèmes de classification. Le classement des modèles RNF est basé, principalement, sur le type de raisonnement utilisé : Généralement, Mamdani ou TS, avec bien sur la prise en considération des dynamiques que présentent ces modèles [169]. Un autre modèle basé sur le raisonnement flou de Tsukamoto est aussi utilisé tout en restant très restreint [28] - [29]. L'arborescence schématisée par la figure 2.6 présente le résultat d'une recherche bibliographique approfondie sur les différents modes d'hybridation neuro - flous.

2.2.2.1. Modèles basés sur le raisonnement de TS

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

a. Modèles statiques

Ce type de modèles regroupe les catégories ANFIS, GARIC, RBF, SANFIS-II &III.

a.1. Modèle ANFIS

L'architecture ANFIS proposée par J. R. Jang a pour objectif d'exploiter les capacités d'apprentissage des RNA pour la détermination d'une base optimale des règles floues du raisonnement approximatif du type TS [170] - [171]. La figure 2.7 illustre l'architecture d'un ANFIS à 2 règles d'inférence, 2 variables d'entrée x_1, x_2 et une sortie f. Les règles sont données par :

Règle 1 : **SI** x_1 **est** A_1 **ET** x_2 **est** A_2 **ALORS** $f = r_1 + p_1 \cdot x_1 + q_1 \cdot x_2$
Règle 2 : **SI** x_1 **est** A_2 **ET** x_2 **est** B_2 **ALORS** $f = r_2 + p_2 \cdot x_1 + q_2 \cdot x_2$ (2.18)

***Couche 1* :** Chaque neurone de cette couche réalise l'opération de fuzzification. La sortie est donnée par :

$$O_{1,i} = \mu_{A_i}(x_1)\big|_{i=1,2} \;\&\; O_{1,i} = \mu_{B_{i-2}}(x_2)\big|_{i=3,4} \tag{2.19}$$

Les fonctions d'appartenance de l'espace des entrées varient d'une forme à l'autre (triangulaire, gaussienne, ...etc.). Dans la version de base de l'ANFIS, la fonction de cloche généralisée est utilisée. Le degré d'appartenance est calculé par :

$$\mu_{A_i}(x) = \frac{1}{1 + \left[\left(\frac{(x - c_i)}{a_i}\right)^2\right]^{b_i}} \tag{2.20}$$

Couche 2 : Couche d'inférence : Chaque noeud calcule le degré d'activation de chaque règle, appliquant une conjonction *Min* ou *prod* :

$$O_{2,i} = \min\left(\mu_{A_i}(x_1), \mu_{B_i}(x_2)\right)\Big|_{i=1,2} \quad \text{ou} \quad O_{2,i} = \mu_{A_i}(x_1)\cdot\mu_{B_i}(x_2)\Big|_{i=1,2} \tag{2.21}$$

Couche 3 : Couche de normalisation. La sortie de chaque neurone est donnée par :

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2} \qquad i = 1, 2 \tag{2.22}$$

Couche 4 : Un neurone de cette couche est appelé neurone adaptatif. Sa sortie est :

$$O_{4,i} = \overline{w}_i \cdot f_i = \overline{w}_i \cdot \left(r_i + p_i \cdot x_1 + q_i \cdot x_2\right)\Big|_{i=1,2} \tag{2.23}$$

où les $\{p_i, q_i, r_i\}$ sont les paramètres des conséquences.

Couche 5 : Le neurone de sortie calcule la somme des signaux :

$$O_{5,i} = \overline{w}_1 \cdot f_1 + \overline{w}_2 \cdot f_2 \Big|_{i=1,2} \tag{2.24}$$

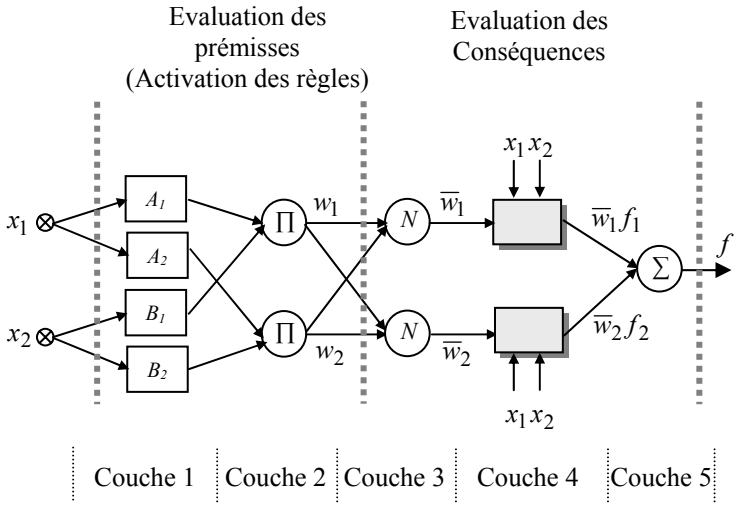


Fig. 2.7 : Modèle ANFIS.

Dans la structure ANFIS, l'adaptation ne concerne que les éléments de la couche 1 et la couche 5, c'est à dire, les paramètres des prémisses et des conséquences. L'apprentissage s'effectue, généralement, par un algorithme hybride [172]. Une fois les paramètres des prémisses sont fixés, les paramètres des conséquences sont identifiés par un algorithme d'identification paramétrique.

La méthode la plus utilisée est celle des MCR dans le sens direct « *forward* » du réseau. La deuxième phase est l'ajustement des paramètres des prémisses par l'algorithme

de la rétropropagation basée sur la descente du gradient dans le sens inverse « backward » [173]. Le tableau 2.1 résume les différentes étapes de l'algorithme d'apprentissage.

Tab. 2.1 : Phases d'apprentissage hybride de l'ANFIS.

Sens / Paramètres	Direct	Inverse
Paramètres des prémisses	Fixés	Descente du gradient
Paramètres des conséquences	MCR	Fixés
Type de signal	Signal de sortie	Signal erreur

Une autre alternative pour l'apprentissage des réseaux ANFIS est l'emploi des méthodes d'optimisation d'ordre zéro à savoir les AG, le recuit simulé et la méthode des essaims particulaires [174] - [176].

Dans le cas où les conséquences sont constantes (TS0), l'algorithme de la rétropropagation est suffisant pour l'ajustement de la totalité des paramètres adaptatifs du réseau ANFIS [177]. Mahdi Jalili-Kharaajoo a utilisé un algorithme d'apprentissage émotionnel (emotional learning) pour l'amélioration des performances de l'ANFIS [178].

a.2. Modèle GARIC

Le modèle GARIC a été développé par Berenji en 1992 comme l'indique la figure 2.8 [179]. Trois parties principales constituent cette architecture :

- Réseau statique de sélection des actions (ASN).
- Réseau statique de sélection des évaluations (AEN).
- Modificateur stochastique des actions (SAM).

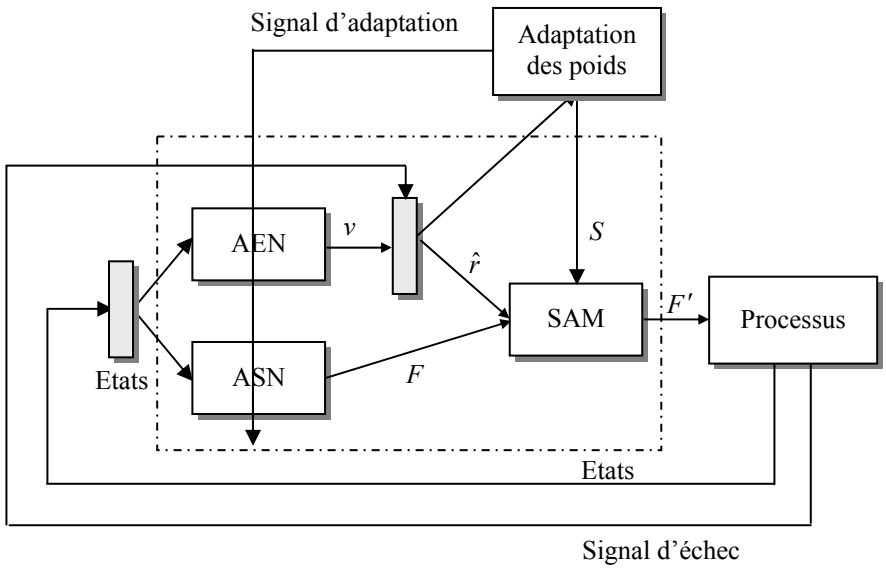


Fig. 2.8 : Architecture du modèle GARIC.

L'ASN consiste à mettre un vecteur d'état dans une action recommandée F, en utilisant le réseau ANFIS dans sa version originale. L'AEN explore un vecteur d'état et un signal d'échec dans des points scalaires qui indiquent la qualité d'état. Ceci est, également,

employé pour produire le signal de renforcement interne $\hat{r}$. Le SAM utilise F et $\hat{r}$ pour produire le signal de commande F' à appliquer sur le processus [180]. Dans certaines applications, le bloc ASN est construit à partir du réseau donné par la figure 2.9 [169] & [181]. Dans cette situation, on fait appel au raisonnement approximatif de Mamdani.

L'apprentissage le mieux adapté à l'architecture GARIC est l'apprentissage par renforcement. Les robots mobiles représentent des exemples concrets de l'application du modèle GARIC. Pour plus de détails, nous citons la référence [182].

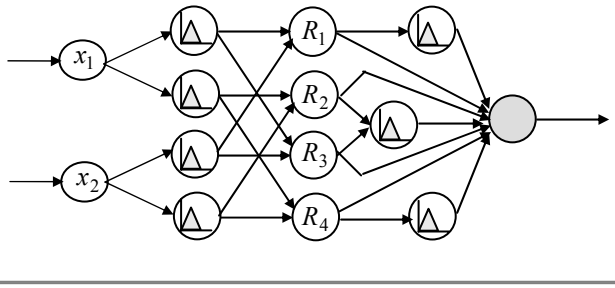


Fig. 2.9 : Réseau du bloc ASN.

a.3. RBF-flou

Le réseau de neurones à fonction de base radiale ou tout simplement réseau à base radiale a été introduit par Powell [183] - [184] comme alternative des RNA. Un RBF est défini par un réseau à 2 couches, représenté par la figure 2.10. Sa fonction est déterminée par :

$$y = f(x) = \sum_{i=1}^{M} w_i \cdot \Phi_i(x) \tag{2.25}$$

Des exemples de non-linéarités $\Phi_i(x)$ pour des RBF sont récapitulés dans le tableau 2.2 [185]. N'importe quelle fonction radiale peut être utilisée, le choix habituel pour la fonction de base $\Phi_i(x)$ est la gaussienne, donnée par le modèle :

$$\Phi_i(x) = \exp\left(-\left\| \frac{x - c_i}{2\sigma_i} \right\|^2 \right) \tag{2.26}$$

$\|\cdot\|$ est une norme (en général la norme Euclidienne ou la distance de Mahalanobis). Les vecteurs σ_i^2 et c_i, $(i = 1, ..., M)$ sont les variances et les centres des gaussiennes, respectivement. J. S. R. Jang a montré la similarité de l'approche RBF avec les SIF [186]. L'équivalence s'obtient pour un choix bien spécifique des différents opérateurs flous et des fonctions d'appartenance définissant les termes linguistiques [48].

Remarque :

Le modèle du neurone flou n'a pu être formulé que grâce à la propriété de la fonction exponentielle : $\exp\left(\sum_{i=1}^{n} (\alpha_i) \right) = \prod_{i=1}^{n} \exp(\alpha_i)$.

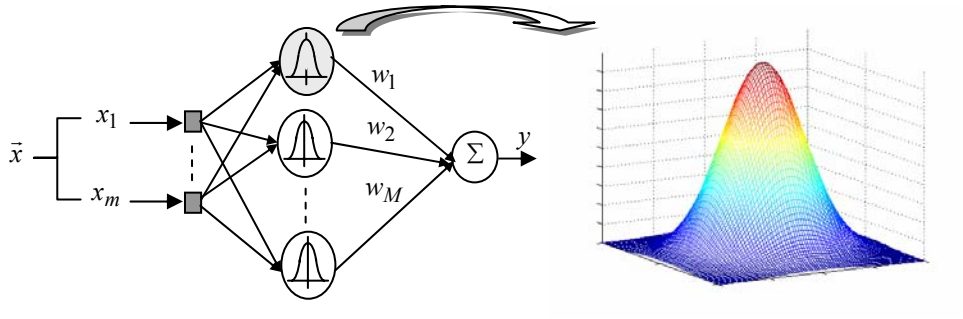


Fig. 2.10 : RBF-flou (activation gaussienne).

Tab. 2.2 : Exemples des fonctions utilisées dans les RBF.

Type	Expression de la fonction $\Phi(r,\sigma)$ (σ = constant)
Linéaire	r
Cubique	r^3
Fonction d'Epanechnikov	$\max\left(0, 1-r^2\right)$
Triangulaire	$\max(0, 1-r)$
Spline à feuille épaisse	$r^2 \log r$
Gaussienne	$\exp\left(-\frac{r^2}{\sigma^2}\right)$
Logistique	$\frac{1}{1+\exp\left(-\frac{r^2}{\sigma^2}\right)}$
Multiquadratique	$\left(r^2+\sigma^2\right)^{\beta} \quad 0<\beta<1$
Multiquadratique inverse	$\left(r^2+\sigma^2\right)^{-\alpha} \quad \alpha>0$

Noter que les poids ajustables sont, seulement, présents dans la couche de sortie. Les liens entre la couche d'entrée et la couche cachée sont fixés à un. Les paramètres du RBF sont les poids w_i et les paramètres de la fonction à base radiale. D'après l'expression (2.25), la sortie du réseau est fonction linéaire de w_i. Ces poids peuvent être estimés par la méthode des MCR.

L'adaptation des paramètres du RBF $\{c_i, \sigma_i\}$ peut être vue comme un problème d'optimisation non linéaire dont la solution est l'utilisation de la méthode de la descente du gradient [187]. Une extension de cette architecture RBF est obtenue en associant une fonction linéaire à la sortie de chaque nœud :

$$w_i = a_i \cdot x + b_i \qquad i = 1, 2, ..., M \tag{2.27}$$

où a_i sont des vecteurs des paramètres et les b_i sont des constantes scalaires. Il faut remarquer que dans la première forme, le RBF est équivalent à un modèle de TS0 [186]. Dans l'extension de l'expression (2.27), le RBF correspond à un modèle de TS linéaire [188].

Les AG présentent une solution pour l'apprentissage des paramètres du RBF-flou. Un exemple typique d'application de cette forme est présenté dans [101]. Un état de l'art des combinaisons RBF-GA est présenté par C. Harpham et al. Dans [189]. Dans certains travaux, l'apprentissage hybride, c'est-à-dire, combinaison des méthodes d'optimisation : AG, descente du gradient et MCR et orthogonales est utilisé pour préserver les capacités d'apprentissage et de généralisation des réseaux RBF [190] - [191]. Roman Neruda et al. ont présenté une série des algorithmes d'apprentissage des RBF avec différentes combinaisons possibles [192]. Leurs performances sont examinées sur l'exemple du voyageur de commerce.

a.4. Modèle SANFIS

Une architecture à 3 modèles flous est proposée par Jeen-Shing Wang et al. : SANFIS-I, SANFIS-II et SANFIS-III [193]. Les 2 types du raisonnement de Mamdani et de TS sont exploités. La j^{eme} règle est donnée par :

$$R^j : \textbf{SI}\left(x_1 \textbf{ est } A_1^j\right)\textbf{... ET}\left(x_n \textbf{ est } A_n^j\right) \textbf{ ALORS } \left(y_1 \textbf{ est } f_1^j\right)\textbf{... ET}\left(y_m \textbf{ est } f_m^j\right) \qquad (2.28)$$

où

$$f_k^j = \begin{cases} B_k^j & \text{(SANFIS - I)} \\ \theta_k^j & \text{(SANFIS - II)} \\ b_{0k}^j + b_{1k}^j \cdot x_1 + \ldots + b_{nk}^j \cdot x_n & \text{(SANFIS - III)} \end{cases} \qquad (2.29)$$

La topologie de la fonction floue de base (fuzzification, inférence, normalisation) est représentée par le schéma de la figure 2.11.

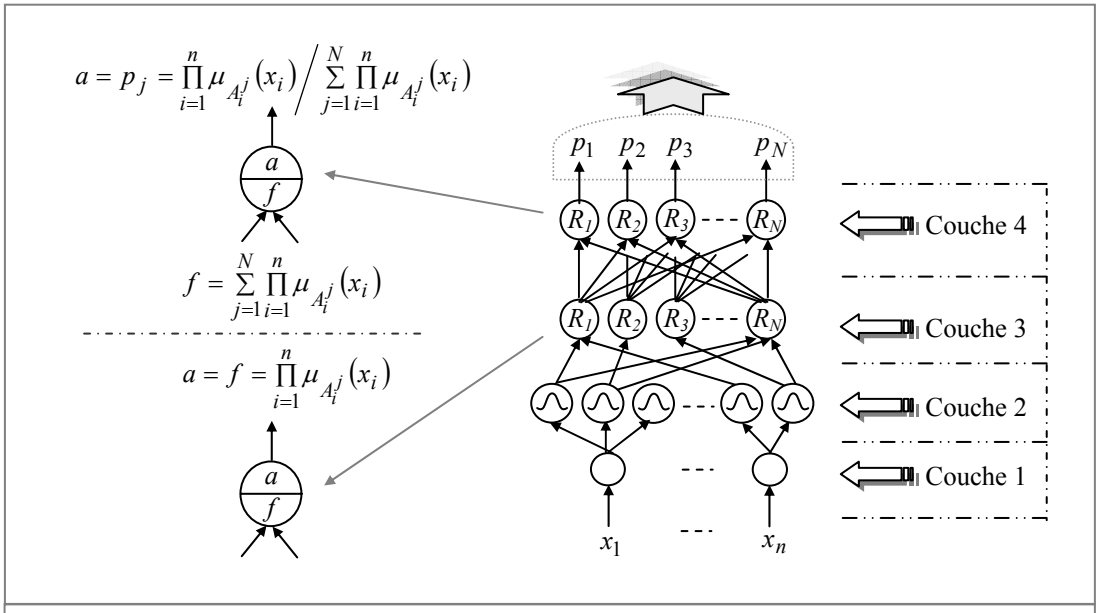


Fig. 2.11 : Topologie de la fonction floue de base.

$\left(B_k^j\right)$, $\left(\theta_k^j\right)$ et $\left(b_{0k}^j + b_{1k}^j \cdot x_1 + ... + b_{nk}^j \cdot x_n\right)$ représentant, respectivement, les ensembles flous de sortie, les composants singletons (pics) et les combinaisons linéaires des variables d'entrées. L'ajout d'une couche de sortie à la fonction floue de base, de la figure 2.11, donne naissance aux modèles SANFIS-I, SANFIS-II et SANFIS-III (Fig. 2.12).

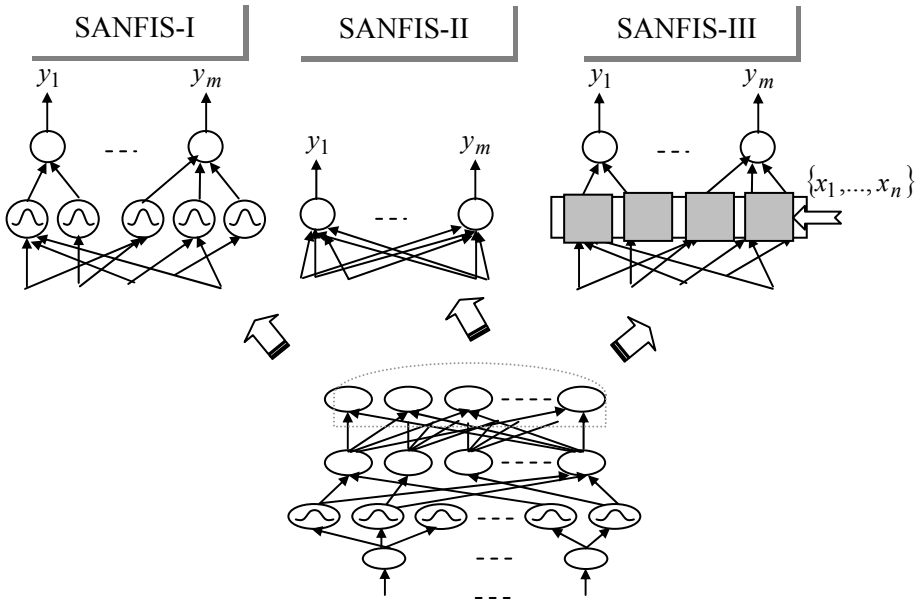


Fig. 2.12 : Structure SANFIS.

L'apprentissage s'effectue par un algorithme hybride, la technique de groupage pour l'initialisation de la structure du réseau SANFIS suivie par la méthode des MCR et l'algorithme de Levenberg–Marquardt modifié [193].

b. Modèles dynamiques

Afin d'identifier des systèmes non linéaires, on fait appel, généralement, à des structures NFR. Les modèles NFR sont des systèmes NF avec des rétroactions de connexions 'feedback' de différentes formes '*topologies*' dans la structure du réseau. Cet asservissement a pour but de mémoriser le passé du modèle.

b.1. Modèle RFNN

Cette structure est une généralisation du la FNN, proposée par Ching-Hung Lee [194]. Les principales caractéristiques de la structure RFNN sont :

- L'aspect dynamique.
- La capacité de stockage temporel des informations.
- Approximation universelle des fonctions.

Le diagramme schématique d'un RFNN à 4 couches, *n* entrées, *p* sorties, *m* partitions floues pour chaque variable d'entrée et $n \times m$ règles est présenté par la figure 2.13. Les notations : I_i^k, O_i^k et w_i^j sont respectivement, la i^{eme} entrée du neurone de la couche *k*, la sortie de neurone *i* dans la couche *k* et les poids synaptiques entre les couches *i* et *j*.

***Couche 1* :** Couche de transfert des signaux d'entrée vers la prochaine couche directement, c'est-à-dire :

$$O_i^{(1)} = w_i^{(1)} \cdot I_i^{(1)} \quad \& \quad w_i^{(1)} = 1 \tag{2.30}$$

***Couche 2* :** La fonction d'activation de chaque neurone dans cette couche est la gaussienne donnée par :

$$O_{ij}^{(2)} = \exp\left\{ -\frac{\left(I_{ij}^{(2)} - m_{ij}\right)^2}{\left(\sigma_{ij}\right)^2} \right\} \quad \& \quad \begin{cases} I_{ij}^{(2)}(t) = O_i^{(1)}(t) + O_{ij}^{(f)}(t) \\ O_{ij}^{(f)}(t) = O_{ij}^{(2)}(t-1) \cdot \theta_{ij} \end{cases} \tag{2.31}$$

L'indice *ij* signifie le terme *j* de la i^{eme} entrée x_i et θ_{ij} représente le poids du lien de l'unité de rétroaction. A noter que cette couche est la source des effets mémoires $O_{ij}^{(2)}(t-1)$, qui stockent l'information passée du réseau.

***Couche 3* :** Cette couche représente la base des règles floues. La sortie de chaque neurone est calculée par l'opérateur de conjonction **ET** :

$$O_i^{(3)} = \prod_{j=1} O_j^{(2)} \tag{2.32}$$

Les liens entre la couche 2 et la couche 3 sont égaux à l'unité.

***Couche 4* :** Les neurones effectuent la sommation linéaire (processeur de TS). Le modèle mathématique du nœud *i* est :

$$O_j^{(4)} = \sum_{i=1}^{m} w_{ij}^3 \cdot O_j^{(3)} \tag{2.33}$$

où $w_{ij}^{(3)}$ sont les paramètres des conséquences à identifier par un processus d'apprentissage adéquat.

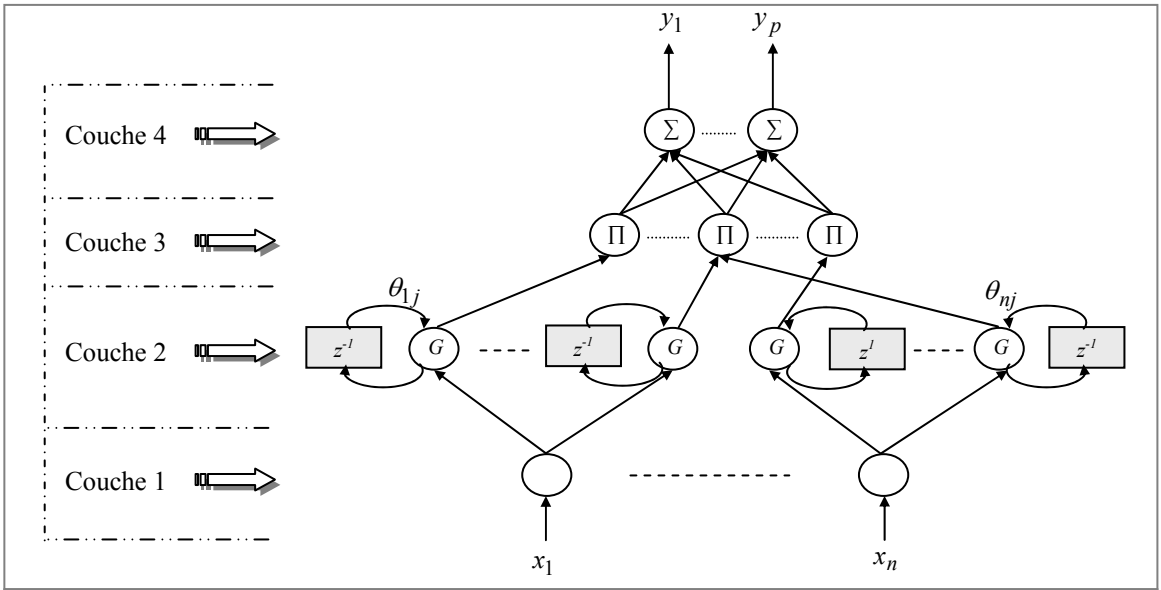
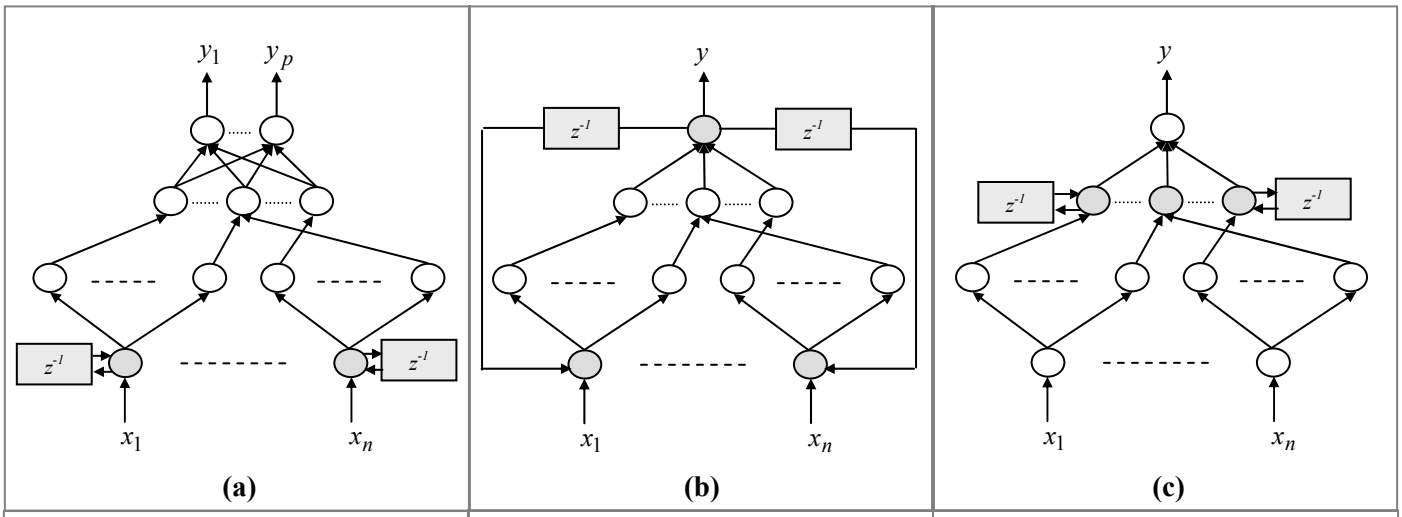


Fig. 2.13 : Structure du modèle RFNN de base.

La rétropropagation du gradient et les AG sont souvent utilisés comme outils d'apprentissage des paramètres de la structure RFNN [194] - [196]. Chih-Min Lin et al. ont proposé un apprentissage on-line du modèle RFNN basé sur la descente du gradient et la théorie de stabilité de Lyapunov [197]. Certains auteurs proposent des versions de la structure RFNN où l'effet mémoire temporel se trouve dans les nœuds de la couche d'entrée, de la sortie du réseau vers ses entrées et dans la couche des inférences des règles [198] - [200]. Ces types sont illustrés par la figure 2.14.



(a) : Proposé dans [198].	**(b) :** Proposé dans [199].	**(c) :** Proposé dans [200].

Fig. 2.14 : Structure du modèle RFNN améliorée.

b.2. Modèle TSRFNN

Le modèle TSRFNN combine un réseau récurrent multicouches avec un modèle TS flou dynamique [201]. La configuration de TSRFNN est donnée par la figure 2.15. La propagation des informations et la fonction de chaque couche seront détaillées ci-dessus.

***Couche 1* :** Couche d'entrée réalise le transfert direct des signaux d'entrées externes. Son modèle est :

$$I_i^{(1)}(t) = x_i(t) \quad \& \quad O_i^{(1)}(t) = f_i^{(1)}\left(I_i^{(1)}(t)\right) = I_i^{(1)}(t) \tag{2.34}$$

***Couche 2* :** Couches des prémisses. Chaque neurone de cette couche représente un terme linguistique pour une variable d'entrée et calcule le degré d'appartenance. La fonction gaussienne utilisée est donnée par :

$$I_{ij}^{(2)}(t) = -\left(\frac{O_i^{(1)}(t) - m_{ij}}{\sigma_{ij}}\right)^2 \quad \& \quad O_{ij}^{(2)}(t) = f_{ij}^{(2)}\left(I_{ij}^{(2)}(t)\right) = \exp\left(I_{ij}^{(2)}(t)\right) \tag{2.35}$$

***Couche 3* :** Couche des règles. Chaque neurone dans cette couche représente une règle d'inférence floue. L'opérateur de conjonction **ET** est utilisé pour calculer la sortie du neurone k :

$$\begin{cases} I_k^{(3)}(t) = \prod_{i=1}^{M} O_i^{(2)} = \exp\left\{ -\left[D_i\left(x_j^{(2)} - c_j\right)\right]^T \left[D_i\left(x_j^{(2)} - c_j\right)\right]\right\} \\ O_k^{(3)}(t) = f_k^{(3)}\left(I_k^{(3)}(t)\right) = I_k^{(3)}(t) \end{cases} \tag{2.36}$$

avec $D_j = \text{diag}\left(1/\sigma_{1j}, 1/\sigma_{2j}, ..., 1/\sigma_{mj}\right)$ & $c_j = \left(c_{j1}, c_{j2}, ..., c_{jN}\right)$. Les liens entre la couche 2 et la couche 3 sont égaux à l'unité.

***Couche 4* :** Couche d'état (state layer). Les neurones effectuent la sommation linéaire. Le modèle mathématique du neurone *i* est :

$$\begin{cases} I_i^{(4)}(t) = \sum_{k=1}^{M} w_{ik}^{(4)} \cdot \dfrac{O_k^{(3)}(t)}{\sum_{k=1}^{M} O_k^{(3)}(t)} = \sum_{k=1}^{M}\left(\sum_{l=1}^{m} a_{il} \cdot h_l(t) + \sum_{l=1}^{m} b_{il} \cdot x_l(t)\right) \dfrac{O_k^{(3)}(t)}{\sum_{k=1}^{M} O_k^{(3)}(t)} \\ h_i(t+1) = O_i^{(4)}(t) = f_i^{(4)}\left(I_i^{(4)}(t)\right) = I_i^{(4)}(t) \end{cases} \tag{2.37}$$

où a_{il} et b_{il} sont les paramètres des conséquences à identifier par un processus d'apprentissage adéquat.

***Couche 5* :** Couche de sortie. La fonction de chaque neurone est donnée par :

$$I_o^{(5)}(t) = \sum_{v=1}^{n} w_{0v}^{(5)} \cdot O_v^{(4)}(t-1) \quad \& \quad y_0(t) = O_0^{(5)}(t) = f_0^{(5)}\left(I_0^{(5)}(t)\right) = I_0^{(5)}(t) \tag{2.38}$$

avec $w_{0v}^{(5)}$ est un poids de valeur fixe.

***Couche de rétroaction* :** Chaque neurone dans cette couche est un nœud récurrent. $O_i^{(r)}(t)$ est la sortie de nœud *i* à l'instant *t* et $I_i^{(r)}(t)$ est son entrée. Selon la figure 2.15 :

$$\begin{cases} I_i^{(r)}(t) = h_i^{(r)}(t) \\ O_i^{(r)}(t) = I_i^{(r)}(t)\Big|_{1 \le i \le n} = O_i^{(4)}(t-1) \end{cases} \tag{2.39}$$

Finalement, une formulation globale du TSRFNN sera donnée par :

$$\begin{aligned} h_v(t+1) = O_v^{(4)}(t) &= \frac{\sum_{k=1}^{M} w_{ik}^{(4)} \cdot O_k^{(3)}(t)}{\sum_{k=1}^{M} O_k^{(3)}(t)} \\ y_0(t) = O_0^{(5)}(t) &= \sum_{v=1}^{n} w_{ov}^{(5)} \cdot O_v^{(4)}(t-1) \end{aligned} \tag{2.40}$$

L'apprentissage des paramètres du TSRFNN nécessite l'ajustement des fonctions d'appartenance des prémisses ainsi que l'identification des paramètres des conséquences.

L'approche hybride basée sur une combinaison de la descente du gradient et de l'algorithme des MCR est proposée par Ying-Chung Wang et al. [201].

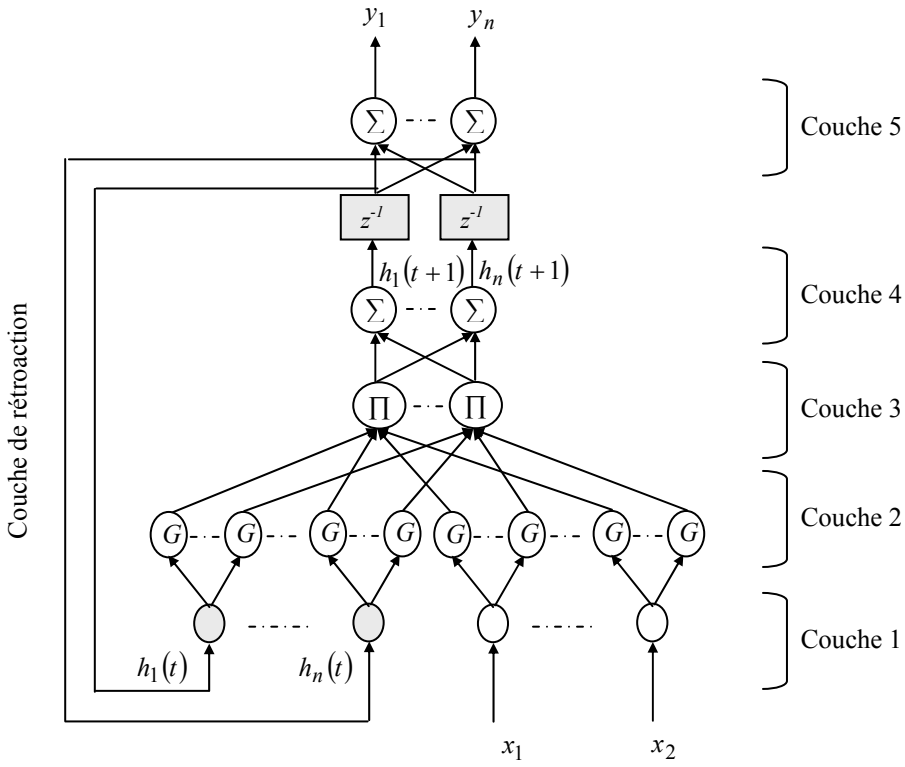


Fig. 2.15 : Structure du modèle TSRFNN.

b.3. Modèle DFNN

DFNN proposé par Paris A. Mastorocostas & John B. Theocharis est un modèle flou dont les conséquences des règles sont implémentées par des réseaux de neurones récurrents [202]. Chaque réseau représente un approximateur local dans le domaine d'application défini par les antécédents de la règle correspondante. La i^{eme} règle est définie par :

$$\textbf{SI} \quad \left(x(t) \textbf{ est } A^{(i)}\right) \quad \textbf{ALORS} \quad \left(\tilde{y}^{(i)}(t) = RNR^{(i)}(x(t))\right) \tag{2.41}$$

$x(t)$ est le vecteur d'entrée. $A^{(i)}$ est la partition floue dans la partie prémisse et $RNR^{(i)}$ est le RNA correspondant à la partie conséquence de la i^{eme} règle (basé sur l'approche classique du modèle flou de TS). La forme générale du modèle DFNN est représentée par la figure 2.16.

$\mu_i(t)$, $(i = 1, ..., M)$ est le degré d'activation de la i^{eme} règle et M est le nombre global des règles floues. Chaque $RNR^{(i)}(x(t))$ dans (2.41) est une $m - H - 1$ structure neuronale récurrente schématisée par la figure 2.17, où m est le nombre des entrées. H est le nombre des neurones dans la couche cachée et 1 représente le neurone de la sortie. Les neurones dans les couches cachée et sortie sont de type G-FGS.

L'architecture d'un G-FGS à une entrée $\zeta(t)$ et une sortie $\psi(t)$ est représentée par la figure 2.18. Basée sur cette architecture, la sortie du modèle G-FGS est donnée par :

$$\psi(t) = f\left(\overbrace{\sum_{j=1}^{O_\zeta} \left(b_j \cdot \zeta(t-j) \right)}^{FIR} + \underbrace{\sum_{i=1}^{O_\psi} \left(a_i \cdot \psi(t-i) \right)}_{IIR} + c \right) \tag{2.42}$$

où

$\underbrace{b_j}_{j=0,\ldots,O_\zeta}$ & $\underbrace{a_i}_{i=1,\ldots,O_\psi}$: Sont des poids synaptiques correspondant aux filtres FIR et IIR constituant le modèle récurrent du neurone G-FGS, respectivement.

c : Est une constante représentant le seuil du neurone.

$f(\cdot)$: Est la fonction d'activation du neurone G-FGS.

O_ζ & O_ψ : Sont le degré du retard de l'entrée et des signaux de retour '*local output feedback*', respectivement.

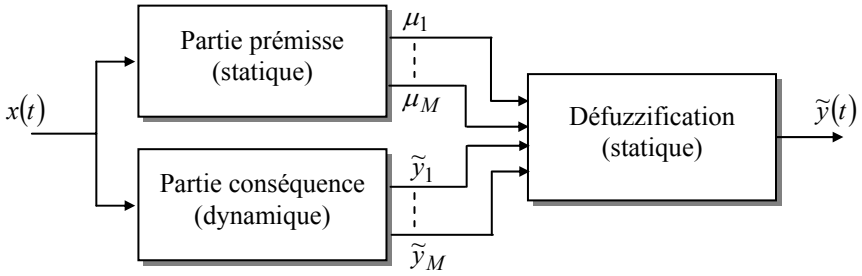


Fig. 2.16 : Structure générale du modèle DFNN.

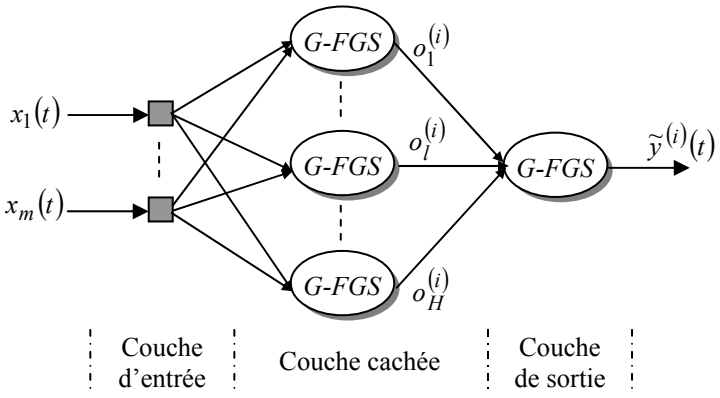


Fig. 2.17 : Partie dynamique du modèle DFNN.

Le tableau 2.3 donne une description des différents paramètres intervenant dans la modélisation des G-FGS.

Tab. 2.3 : Description des paramètres fonctionnels d'un G-FGS.

Paramètres	Description
f_1 & f_2	Sont les fonctions d'activation des neurones des couches cachées et de sortie, respectivement. Paris A. Mastorocostas et al. ont proposé d'utiliser les fonctions du type tangente hyperbolique $\tanh(.)$ comme fonction d'activation [202].
$O_l^{(i)}$	Est la sortie du neurone *i* dans la couche cachée à l'instant *t*.
$\tilde{y}^{(i)}(t)$	Est la sortie du neurone de la couche de sortie, $\tilde{y}^{(i)}(t) = RNN^{(i)}(x(t))$.
O_x & O_{y1}	Sont le degré du retard des entrées des G-FGS et de la couche cachée, respectivement.
O_{y3} & O_{y2}	Sont le degré du retard des entrées et de G-FGS de sortie, respectivement.
$w_{ljq}^{1(i)}$ & $w_{lj}^{2(i)}$	Sont les poids synaptiques des filtres FIR et IIR dans la couche cachée.
$w_{jq}^{4(i)}$ & $w_j^{5(i)}$	Sont les poids synaptiques des filtres FIR et IIR dans la couche de sortie.
$w_l^{3(i)}$ & $w^{6(i)}$	Sont les seuils des neurones cachés et de sortie, respectivement.

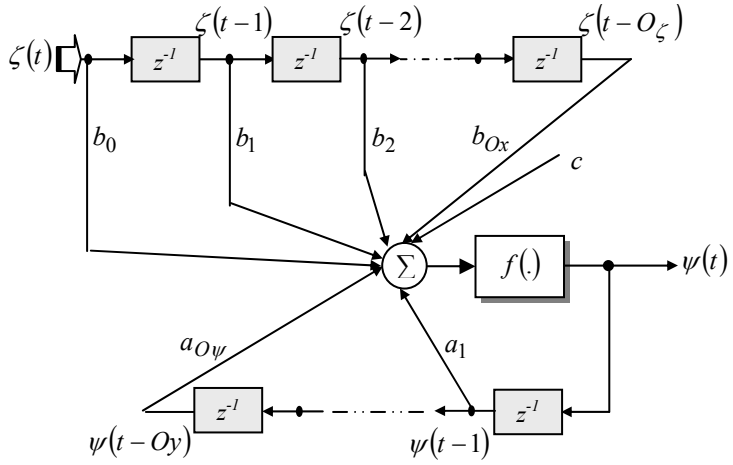


Fig. 2.18 : Topologie de neurone G-FGS.

La partie conclusion de la i^{eme} règle sera calculée par la formule suivante :

$$\tilde{y}^{(i)}(t) = f_2\left(\sum_{j=1}^{H}\sum_{q=0}^{O_{y3}} w_{jq}^{4(i)} \cdot O_j^{(i)}(t-q) + \sum_{j=1}^{O_{y2}} w_j^{5(i)} \cdot \tilde{y}^{(i)}(t-j) + w^{6(i)}\right) \quad i = 1, \ldots, M \tag{2.43}$$

$$O_l^{(i)}(t) = f_1\left(\sum_{j=1}^{m}\sum_{q=0}^{O_x} w_{ljq}^{1(i)} \cdot x_j(t-q) + \sum_{j=1}^{O_{y1}} w_{lj}^{2(i)} \cdot O_l^{(i)}(t-j) + w_l^{3(i)}\right) \quad l = 1, \ldots, H, \quad i = 1, \ldots, M \tag{2.44}$$

Due à l'effet mémoire de la structure DFNN, l'expression (2.43) sera transformée en :

$$\tilde{y}^{(i)}(t) = F^{(i)}\left(x(t), x(t-1), ..., \tilde{y}^{(i)}(t-1), \tilde{y}^{(i)}(t-2), ...\right) \qquad i = 1, ..., M \tag{2.45}$$

$F^{(i)}$ est une transformation non linéaire définie par le réseau neuronal. La sortie du DFNN à l'instant t, $\tilde{y}(t)$, sera calculée par l'utilisation de la méthode de défuzzification des moyennes pondérées (1.26). En conclusion, la sortie du réseau de la figure 2.17 sera donnée par la formule généralisée du contrôleur flou de type TS :

$$\tilde{y}(t) = F\left(x(t), x(t-1), ..., \tilde{y}(t-1), \tilde{y}(t-2), ...\right) \tag{2.46}$$

L'apprentissage s'effectue par application d'un algorithme d'adaptation des poids synaptiques du RNA basé sur l'optimisation contrainte (constrained optimisation) proposé par Paris Mastorocostas et al.. Ce modèle est bien adapté à l'identification des systèmes et aux problèmes d'annulation de bruit [202]. Une étude comparative entre différentes méthodes d'apprentissage des DFNN est présentée par Meng Joo Er et al. & T.G. Barbounis et al. [203] - [204]. La méthode d'estimation des MCR, filtre de Kalman et filtre de Kalman étendu sont les méthodes de base pour l'algorithme d'ajustement des poids du DFNN.

b.4. Modèle TRFN

La figure 2.19 schématise la structure d'un TRFN défini par 2 entrées x_1 et x_2, une sortie y_1 et 2 règles d'inférence R_1 et R_2 [165]. La i^{eme} règle sera formulée par :

$$\begin{aligned} \textbf{Règle } i: \ \textbf{SI} \quad & x_1(t) \ \textbf{est} \ A_{i1} \ \textbf{ET ...} \ \textbf{ET} \ x_m(t) \ \textbf{est} \ A_{im} \ \textbf{ET} \ h_i(t) \ \textbf{est} \ G \\ \textbf{ALORS} \quad & y(t+1) = a_{i0} + a_{i1} \cdot x_1(t) + ... + a_{im} \cdot x_m(t) + a_{i(m+1)} \cdot h_i(t) \ \textbf{ET} \\ & h_i(t+1) \ \textbf{est} \ w_{i1} \ \textbf{ET ....} \ \textbf{ET} \ h_{Nr}(t+1) \ \textbf{est} \ w_{iNr} \end{aligned} \tag{2.47}$$

A_{ij} et G sont des sous ensembles flous et w_{ij} et a_{ij} sont les paramètres de conséquences des sorties h et y, respectivement. Le nombre des variables internes est défini par Nr. Six couches forment le réseau TRFN. On note par $\zeta_i^{(k)}$ l'entrée du nœud *i* dans la couche *k* et $O_i^{(k)}$ sa valeur de sortie.

***Couche 1* :** Couche de transfert des signaux d'entrée vers la deuxième couche.

***Couche 2* :** Deux types de fonctions d'appartenance sont utilisés dans cette couche. Pour les variables d'entrées externes x_i, la fonction gaussienne utilisée est donnée par :

$$O_i^{(2)} = \exp\left\{ -\frac{\left(\zeta_j^{(2)} - m_{ij}\right)^2}{\sigma_{ij}^2} \right\} \quad \& \quad \zeta_j^{(2)} = O_j^{(1)} \tag{2.48}$$

m_{ij} et σ_{ij} sont, respectivement, le centre et l'écart type de la fonction gaussienne de la i^{eme} partition floue de la variable d'entrée x_j. Pour la variable interne h_i, la fonction sigmoïdale est utilisée :

$$O_i^{(2)} = \frac{1}{1+\exp\left(-\zeta_j^{(2)}\right)} \quad \& \quad \zeta_i^{(2)} = O_i^{(5)} \tag{2.49}$$

Les liens entre la couche d'entrée et la couche 2 sont égaux à l'unité.

***Couche 3* :** La sortie de chaque neurone dans cette couche est calculée par l'opérateur de conjonction **ET**. La fonction de chaque règle est :

$$O_i^{(3)} = \prod_{j=1}^{m+1} O_j^{(2)} = \left(1+\exp\left(-O_i^{(5)}\right)\right)^{-1} . \exp\left(-\sum_{j=1}^{m} \frac{\left(O_j^{(1)} - m_{ij}\right)^2}{\sigma_{ij}^2}\right) \tag{2.50}$$

Les liens entre la couche 2 et la couche 3 sont égaux à l'unité.

***Couche 4* :** Les neurones effectuent la sommation linéaire (processeur de TS). Le modèle mathématique du nœud *i* est :

$$O_i^{(4)} = \sum_{j=0}^{m+1} a_{ij}\zeta_j^{(4)} = a_{i0} + \sum_{j=1}^{m} a_{ij}x_j + a_{i(n+1)}h_i \tag{2.51}$$

a_{ij}, $(j = 0, 1, ..., m)$ sont les paramètres à identifier. Les liens entre cette couche et la couche 6 sont égaux à l'unité.

***Couche 5* :** Le noeud du contexte fonctionne comme défuzzificateur pour les règles floues avec l'inférence h_i. Les poids synaptiques représentent les valeurs des singletons dans la partie conséquence des règles floues internes. La méthode des sommes pondérées est bien adaptée à cette situation. Elle est donnée par :

$$h_i = O_i^{(5)} = \sum_{j=1}^{Nr} O_j^{(3)} \cdot w_{ij} \tag{2.52}$$

***Couche 6* :** Le nœud dans cette couche permet de calculer le signal de sortie de TRFN. Il est donné par :

$$y = O^{(6)} = \frac{\sum_{j=1}^{Nr} O_j^{(3)} \cdot O_j^{(4)}}{\sum_{j=1}^{Nr} O_j^{(3)}} \tag{2.53}$$

En résumé, l'inférence et la sortie interne de TRFN sont calculées, respectivement, par :

$$y(t+1) = \frac{\sum_{i=1}^{Nr} \mu_i(x(t), h_i(t)) \cdot f_i(t)}{\sum_{i=1}^{Nr} \mu_i(x(t), h_i(t))}$$
$$h_i(t+1) = \sum_{k=1}^{Nr} \mu_k(x(t), h_i(t)) \cdot w_{ik} \quad (2.54)$$

avec :

$$f_i(t) = a_{i0} + \sum_{j=1}^{m} a_{ij} \cdot x_j + a_{i(n+1)} \cdot h_i(t)$$
$$\mu_i(x(t), h_i(t)) = (1 + \exp(-h_i(t)))^{-1} . \exp\left(- \sum_{j=1}^{m} \frac{(x_j(t) - m_{ij})^2}{\sigma_{ij}^2} \right) \quad (2.55)$$

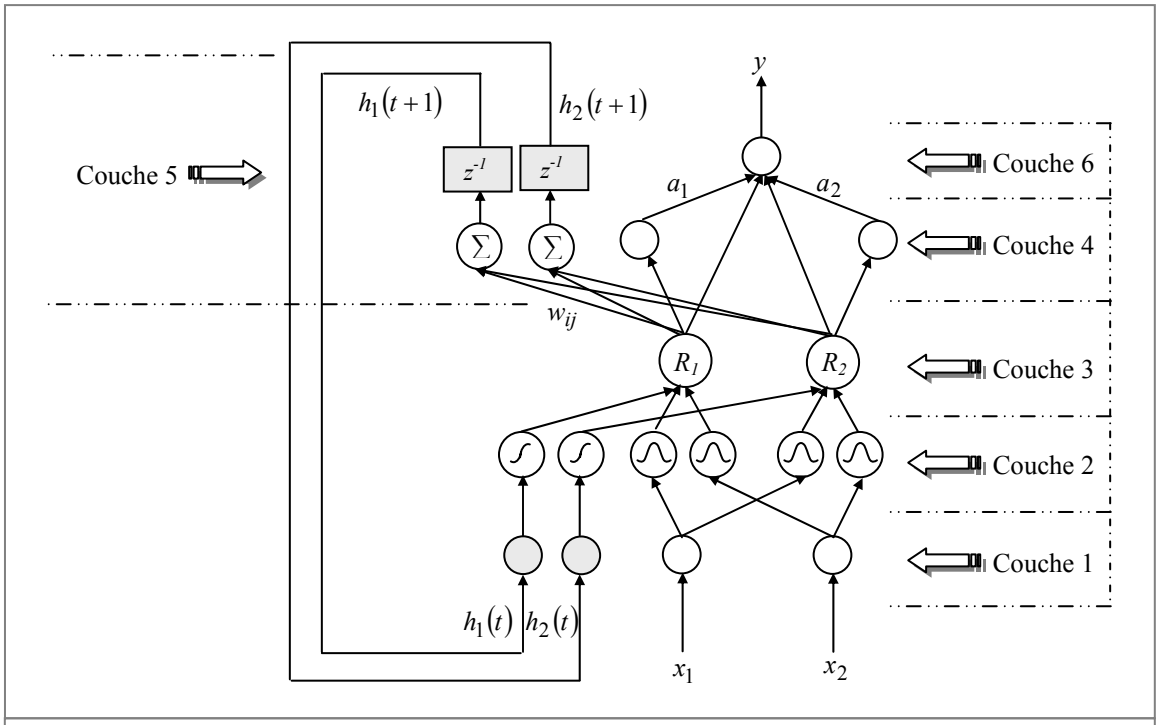


Fig. 2.19 : Structure du modèle TRFN.

La conception de TRFN nécessite 2 phases d'apprentissage et d'optimisation :

- Optimisation structurelle.
- Optimisation paramétrique.

L'objectif de l'optimisation structurelle est de définir le nombre des règles floues, les positions initiales des fonctions d'appartenance ainsi que les valeurs initiales des paramètres des conséquences (phase de configuration et d'initialisation).

Une fois la structure du réseau est spécifiée, le réglage des paramètres permet d'avoir un TRFN à caractère optimal. Dans les méthodes analytiques, l'apprentissage permet l'ajustement des fonctions d'appartenance ainsi que les paramètres des conséquences tout en minimisant le critère de performances désirées (erreur quadratique).

Deux algorithmes sont proposés pour l'ajustement des poids de la couche 4, la méthode des MCR pour l'adaptation du vecteur des paramètres des conséquences et la méthode de la descente du gradient pour l'ajustement des poids a_{ij} [205]. Cependant, si le système flou est bien initialisé, une descente du gradient est très efficace et beaucoup moins gourmande en mémoire.

Les paramètres des fonctions d'appartenance $\{m_{ij}, \sigma_{ij}\}$ ainsi que les poids synaptiques dans la couche 5 $\{w\}$ sont adaptés par l'algorithme d'apprentissage récurrent en temps réel proposé par R. J. Williams [206].

Les méthodes évolutionnaires proposées pour l'adaptation des poids du réseau TRFN $\{m_{ij}, \sigma_{ij}, a_{ij}, w_{ij}\}$ sont basées sur les algorithmes génétiques et la méthode des PSO [165] & [207] - [209]. La structure du chromosome de l'AG qui codifie la structure TRFN sera représentée par la figure 2.20.

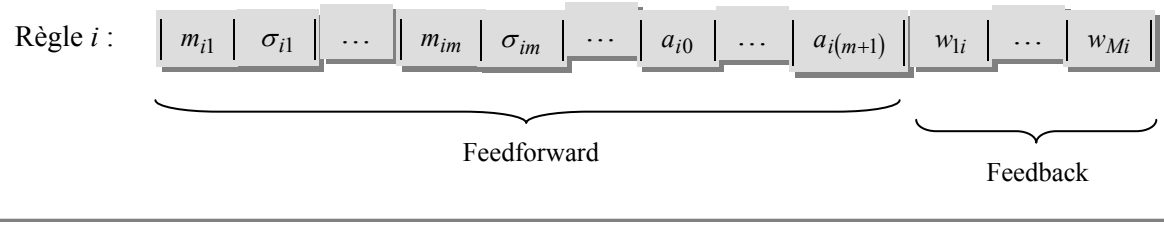


Fig. 2.20 : Représentation génétique du modèle TRFN.

Les paramètres optimaux sont obtenus à la fin de l'exécution de l'algorithme d'optimisation qui se déroule par application des opérateurs adaptés au type du codage utilisé.

b.5. Modèle RFCMAC

Ce modèle, proposé par Jinzhu Peng et al., contient une couche de rétroaction, une mémoire associative, des hypercubes, mémoire des poids et une couche de sortie (Fig. 2.21) [210].

***Couche 1* :** Couche de rétroaction. Chaque neurone effectue le transfert de signal d'entrée vers la prochaine couche. Des liaisons de rétroaction sont ajoutées aux neurones de cette couche permettant la réalisation des temporisations dans la fonction globale du réseau. Pour le neurone i de cette couche, la relation entrée/sortie à l'instant t est donnée par :

$$O_i^{(1)}(t) = I_i^{(1)}(t) = x_i(t) + w_i^{(1)} \cdot O_i^{(1)}(t-1) \qquad i = 1, 2, ..., m \tag{2.56}$$

Les paramètres $w_i^{(1)}$ représentent les poids synaptiques récurrents '*recurrent weights*'.

***Couche 2* :** Couche des prémisses des règles floues. Chaque neurone p est une fonction d'appartenance, qui est donnée dans le cas de la fonction gaussienne par :

$$\begin{cases} \mu_{A_{ij}}\left(O_i^{(1)}\right) = \exp\left\{-\dfrac{\left(O_i^{(1)} - m_{ij}\right)^2}{\left(\sigma_{ij}\right)^2}\right\} & i = 1, 2, ..., m, \quad j = 1, 2, ..., n \\ O_p^{(2)} = I_p^{(2)} = \mu_{A_{ij}}\left(O_i^{(1)}\right) & i = 1, 2, ..., m \cdot n \end{cases} \tag{2.57}$$

***Couche 3* :** La base des règles floues. Un neurone est une règle d'inférence floue. L'opérateur de conjonction ET est utilisé pour calculer la sortie de neurone q :

$$O_q^{(3)} = I_q^{(3)} = \prod_{i=1}^{M} \mu_{A_{ij}}\left(O_i^{(1)}\right) \qquad i = 1, 2, ..., m, \quad q = 1, 2, ..., n^m \tag{2.58}$$

***Couche 4* :** Couche de mémoire associative. Les neurones réalisent l'opération de défuzzification. Le modèle mathématique du nœud i est :

$$O_q^{(4)} = I_q^{(4)} = w_q^{(4)} \cdot O_q^{(3)} \qquad q = 1, 2, ..., n^m \tag{2.59}$$

$w_q^{(4)}$ est un poids qui représente l'action de la règle q (TS0).

***Couche 5* :** Couche de sortie. Elle est donnée par :

$$y = O^{(5)} = I^{(5)} = \sum_{q=1}^{n^m} O_q^{(4)} \tag{2.60}$$

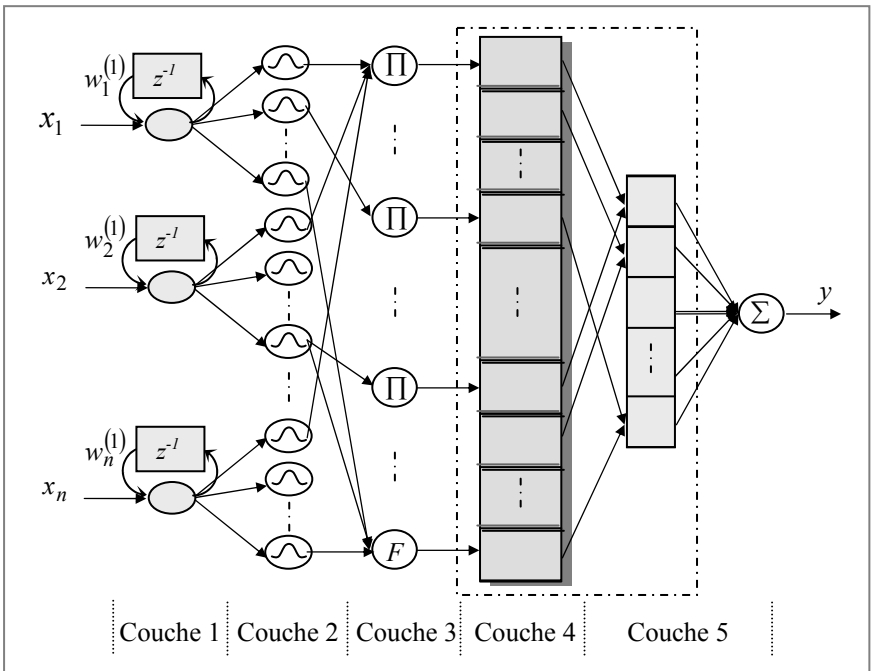


Fig. 2.21 : Architecture RFCMAC.

L'algorithme de la rétropropagation est proposé pour l'adaptation des paramètres ajustables du réseau RFCMAC $\left(w_{sq}^{(4)}, m_{ij}, \sigma_{ij}, w_i^{(1)}\right)$ [210]. Plusieurs versions de structures/algorithmes d'apprentissage du modèle RFCMAC sont proposées dans la littérature [211] - [212].

Cheng-Jian Lin et al. ont proposé un FCMAC paramétrique (P-FCMAC) qui exploite le raisonnement de TS d'ordre 1 [213]. Un apprentissage hybride qui utilise la technique de groupage automatique (self - clustering algorithm) et les AG est proposé pour la construction de P-FCMAC.

2.2.2.2. Modèles basés sur le raisonnement de Mamdani

a. Modèles statiques

Ce type de modèles regroupe les structures FLP, NEFCON, NEFCLASS, NEFPROX et SANFIS-I.

a.1. Modèle FLP

Cette forme est inspirée du raisonnement ou du modèle de Mamdani, appelé FLP ou EFuNN proposée par Jinwoo Kim et al. & Kasabov [158] - [159] & [214]. La structure d'un FLP à 2 variables d'entrées x_1, x_2, une sortie F, 3 sous ensembles flous pour chaque variable d'entrée/sortie et 9 règles est représentée par la figure 2.22. Cinq couches forment le réseau FLP.

Couche 1 **:** Chaque neurone de cette couche représente une variable d'entrée. Le transfert direct des signaux d'entrées vers la couche suivante est effectué au niveau de cette couche :

$$O_i^{(1)} = I_i^{(1)} = x_i \tag{2.61}$$

Couche 2 **:** Couche de fuzzification. Chaque noeud calcule le degré d'appartenance des variables d'entrées.

$$I_j^{(2)} = w_{ij}^{(2)} \cdot O_i^{(2)} \quad \& \quad O_j^{(2)} = f\left(I_j^{(2)}\right) \tag{2.62}$$

La fonction $f(\cdot)$ est la fonction d'appartenance correspondante. Les fonctions triangulaires et gaussiennes sont les plus utilisées.

Couche 3 **:** Couche d'inférence (**ET**). Elle utilise, généralement, un opérateur de conjonction tel que le *MIN* ou le *Produit*. L'entrée/fonction de chaque neurone est donnée par :

$$I_l^{(3)} = \prod_j w_{jl}^{(3)} \cdot O_j^{(2)} \quad \& \quad O_l^{(3)} = I_l^{(3)} \tag{2.63}$$

Couche 4 **:** Couche **OU** : Elle utilise une disjonction (habituellement, le *MAX*). Les relations des nœuds dans cette couche sont :

$$I_k^{(4)} = \sum_k w_{lk}^{(4)} \cdot O_l^{(3)} \quad \& \quad O_k^{(4)} = \min\left(1, I_k^{(4)}\right) \tag{2.64}$$

Couche 5 : Couche de défuzzification. La méthode du centre de gravité est bien adaptée. Elle est donnée par :

$$I_t^{(5)} = \sum_k w_{kt}^{(5)} \cdot O_k^{(4)} = \sum_k \left(m_k \cdot \sigma_k\right) \cdot O_k^{(4)} \quad \& \quad O_t^{(5)} = F = \frac{I_t^{(5)}}{\sum_k \sigma_k \cdot O_k^{(4)}} \tag{2.65}$$

$I_i^{(k)}$ et $O_i^{(k)}$ sont l'entrée et la sortie du neurone *i* de la couche k, respectivement. $w_{ij}^{(k)}$ est le poids synaptique entre le neurone *i* de la couche $k-1$ et le neurone *j* de la couche k.

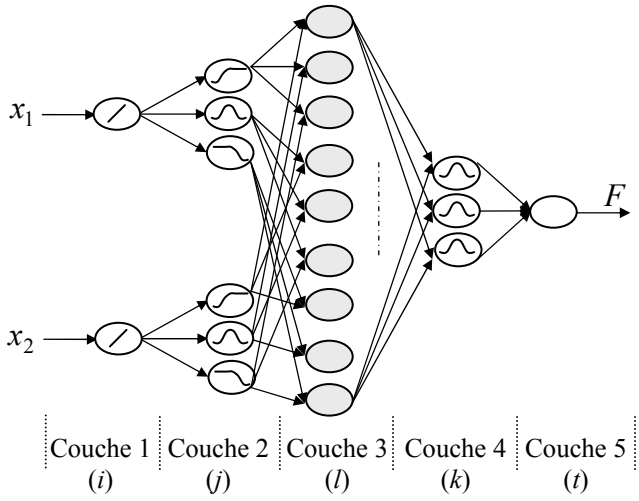


Fig. 2.22 : Architecture FLP.

L'apprentissage s'effectue par la méthode de la rétropropagation. Les paramètres du réseau sont modifiés pour diminuer l'erreur E en sortie sur l'ensemble d'apprentissage X :

$$E = \frac{1}{2} \sum_X \sum_{l=1}^{q} \left(O_l^{(d)} - O_l^{(5)}\right)^2 \tag{2.66}$$

q est le nombre des neurones dans la couche 5. $O_l^{(d)}$ et $O_l^{(5)}$ sont les sorties désirée et actuelle dans l'ensemble X, respectivement.

Les poids ajustables w_{kj} correspondent au neurone k dans la couche 4 et au neurone j dans la couche 3, c'est-à-dire, les fonctions d'appartenance de l'espace d'entrée/sortie. La règle d'ajustement utilise l'algorithme de descente du gradient, où la règle d'enchaînement s'effectue par :

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial O_k^{(4)}} \cdot \frac{\partial O_k^{(4)}}{\partial w_{kj}} = \frac{\partial E}{\partial O_l^{(5)}} \cdot \frac{\partial O_l^{(5)}}{\partial O_k^{(4)}} \cdot \frac{\partial O_k^{(4)}}{\partial w_{kj}} \tag{2.67}$$

Certains travaux utilisent les AG pour l'apprentissage des paramètres adaptatifs du réseau [158] - [159] & [215] - [216]. La diversité entre les différentes propositions réside dans le type du codage des paramètres fonctionnels dans l'AG. Liang Hsuan Chen et al. ont proposé un algorithme pour l'apprentissage du réseau FLP [217]. Deux types d'apprentissage sont utilisés :

- Apprentissage structurel pour la génération d'un ensemble initial de règles floues. Ce dernier est proposé par Wang & Mandel. Il est basé sur l'apprentissage par des exemples '*learning from examples*' [127].
- Apprentissage paramétrique plus fin des fonctions d'appartenance. La technique utilisée est celle de la descente du gradient.

a.2. Modèle NEFPROX

L'architecture NEFPROX proposée par Detlef Nauck & Rudolf Kruse est une extension des modèles NEFCON et NEFCLASS, utilisés dans la commande et la classification, respectivement [218] - [219]. Le NEFPROX est une généralisation du modèle RBF-flou avec le raisonnement approximatif de Mamdani.

Une structure de NEFPROX est représentée par la figure 2.23. Il s'agit d'un modèle à 2 entrées, 2 sorties et 5 règles floues. Comme son nom l'indique, le NEFPROX est dédié beaucoup plus à l'approximation des fonctions. Ce dernier possède les spécifications suivantes :

- Une couche d'entrée réalise le transfert direct des données d'entrées vers la couche cachée. Aucun traitement n'a eu lieu.
- Le neurone de la couche cachée calcule l'inférence de la règle correspondante.
- Les neurones de la couche de sortie ont pour objectif l'opération de défuzzification.

Il est à noter que la différence entre les modèles NEFPROX, NEFCON et NEFCLASS réside au niveau de la dernière couche ainsi qu'aux liaisons entre les couches cachée et sortie, respectivement.

Pour plus de détails sur l'évolution et l'apprentissage des architectures précédentes, on cite la référence [220].

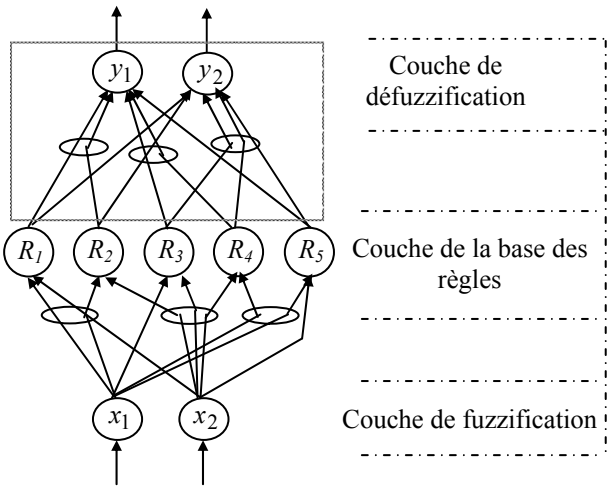


Fig. 2.23 : Système NEFPROX à 5 règles, 2 entrées et 2 sorties.

b. Modèles dynamiques

b.1. Modèle RSONFIN

La structure RSONFIN proposée par Chia-Feng Juang est un RNF à auto - apprentissage [221]. Les noeuds de contexte (context nodes) agissent comme la mémoire interne organisée dans la couche de rétroaction (feedback layer). Le modèle RSONFIN est représenté par la figure 2.24. La propagation des signaux et la fonction des nœuds dans chaque couche seront détaillées ci-dessus.

La forme de la i^{eme} règle floue dynamique, réalisée par le modèle RSONFIN à m entrées $(x_1(t) \sim x_m(t))$, 2 sorties $y_1(t)$ et $y_1(t)$ et M règles, respectivement, sera donnée par l'expression :

$$\begin{aligned} &\textbf{Règle } i: \ \textbf{SI } \ x_1(t) \ \textbf{est } \ A_{i1} \ \textbf{ET ... ET } \ x_m(t) \ \textbf{est } \ A_{im} \\ &\quad \textbf{ET } h_i(t) \ \textbf{est } \ G \\ &\quad \textbf{ALORS } \ y_1(t+1) \ \textbf{est } \ B_{i1} \ \textbf{ET } \ y_2(t+1) \ \textbf{est } \ B_{i2} \\ &\quad \textbf{ET } \ h_1(t+1) \ \textbf{est } \ w_{1i} \ \textbf{ET ... ET } \ h_M(t+1) \ \textbf{est } \ w_{Mi} \end{aligned} \tag{2.68}$$

$(A_{i1} \sim A_{im})$, $(B_{i1} \sim B_{i2})$ et G sont des sous ensembles flous.

***Couche 1* :** Couche de transfert des signaux d'entrée vers la prochaine couche directement :

$$O^{(1)} = I_i^{(1)} \tag{2.69}$$

***Couche 2* :** Chaque neurone dans cette couche est une partition floue de l'espace d'entrée (G). Sa sortie est le degré d'appartenance, donnée par la fonction gaussienne :

$$O^{(2)} = \exp\left\{ - \frac{\left(I_i^{(2)} - m_{ij}\right)^2}{\left(\sigma_{ij}\right)^2} \right\} \tag{2.70}$$

L'indice ij signifie le terme j de la i^{eme} entrée x_i.

***Couche 3* :** Chaque neurone correspond à une règle. La sortie de chaque neurone dans cette couche est calculée par l'opérateur de conjonction **ET**. La fonction de chaque règle est :

$$O^{(3)} = O^{(6)} \cdot \prod_i I_i^{(3)} = O^{(6)} \cdot \exp\left\{ - \left[D_i \cdot (x - m_i)\right]^T \left[D_i \cdot (x - m_i)\right] \right\} \tag{2.71}$$

avec $D_i = \text{diag}(1/\sigma_{i1}, 1/\sigma_{i2}, ..., 1/\sigma_{im})$, $m_i = (m_{i1}, m_{i2}, ..., m_{im})$ et $O^{(6)}$ est la sortie de chaque nœud dans la couche de rétroaction.

***Couche 4* :** Couche des conséquences des règles. Chaque neurone représente le sous ensemble flou de la sortie (gaussienne). Le modèle mathématique du nœud i est l'opérateur **OU** :

$$O_i^{(4)} = \sum_i I_i^{(4)} \tag{2.72}$$

Couche 5 : Couche de défuzzification. La méthode adaptée est celle du centre de gravité, donnée par :

$$y_i = O_i^{(5)} = \frac{\sum_i I_i^{(5)} \cdot \widetilde{m}_{ji}}{\sum_i I_i^{(5)}} \tag{2.73}$$

$\left\{I_i^{(5)} = O_i^{(4)}\right\}$ et $\widetilde{m}_{ji}$ sont des poids de connexion et le centre de la fonction d'appartenance du terme *i* de la variable *j* de la sortie, respectivement.

Couche de rétroaction : Cette couche permet de calculer la valeur de la variable interne h_i. La méthode des sommes pondérées est bien adaptée, donnée par

$$h_i = \sum_i O_i^{(4)} \cdot w_{ji} \tag{2.74}$$

w_{ij} est le poids synaptique du neurone *i* de la couche 4 et la j^{eme} variable interne. La méthode du centre de gravité peut, aussi, être utilisée pour calculer $h_i = \sum_i O_i^{(4)} \cdot w_{ji} \Big/ \sum_i O_i^{(4)}$. Le neurone de rétroaction calcule la sortie par la sigmoïde *S*, donnée par :

$$O^{(6)} = \frac{1}{1 + \exp(-h_i)} \tag{2.75}$$

Cette sortie sera connectée aux neurones des règles de la couche 3.

En conclusion, la fonction réalisée par le modèle RSONFIN est donnée par les relations suivantes :

$$y_i(t+1) = \frac{\sum_i I_i^{(5)} \cdot \widetilde{m}_{ji}}{\sum_i I_i^{(5)}} \tag{2.76}$$

$$I_i^{(5)} = \sum_k O_i^{(6)} \cdot I_{ik}^{(4)} = \sum_k O_i^{(6)} \cdot \prod_i I_{ki}^{(3)} = \sum_k O_i^{(6)} \cdot \exp\left\{-\sum_i \left(\frac{x_i(t) - m_{ik}}{\sigma_{ik}}\right)^2\right\} \tag{2.77}$$

$$a_k^{(6)} = \frac{1}{1 + \exp(-h_k)} \tag{2.78}$$

$$h_k(t) = \sum_l w_{kl} O_l^{(4)} \cdot (t-1) = \sum_l w_{kl} \cdot O_l^{(6)}(t-1) . \exp\left\{-\sum_j \left(\frac{x_j(t-1) - m_{jl}}{\sigma_{jl}}\right)^2\right\} \tag{2.79}$$

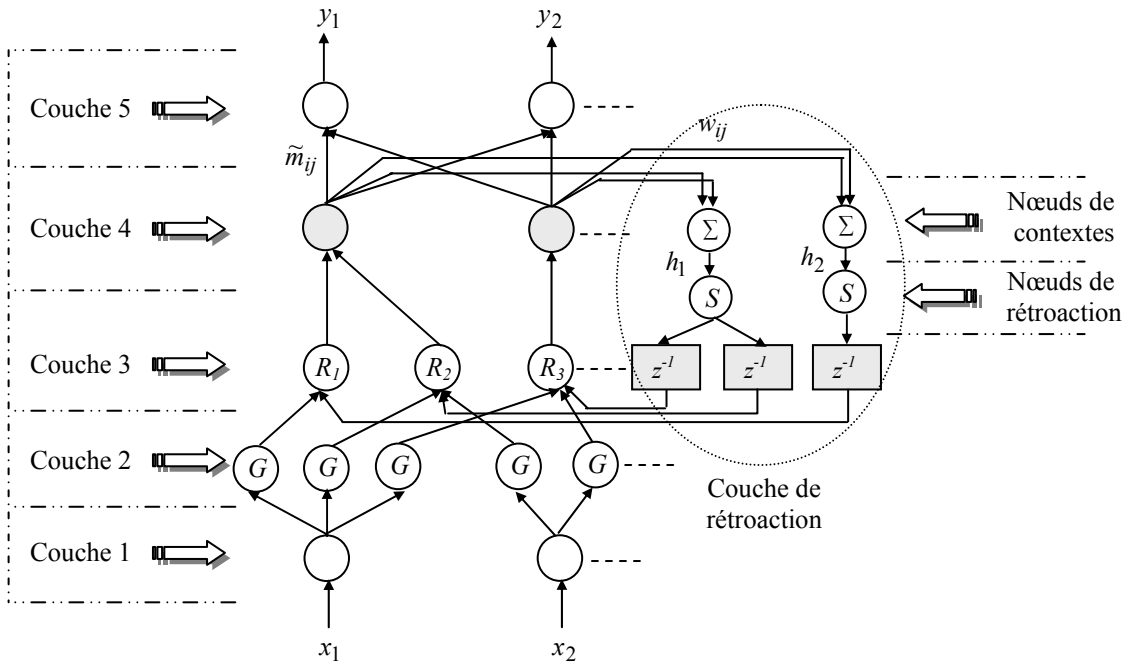


Fig. 2.24 : Structure du modèle RSONFIN.

L'apprentissage du modèle RSONFIN nécessite 2 phases : Structurelle et paramétrique. L'apprentissage consiste à :

- Partitionner l'univers de discours des variables d'entrées/sorties.
- Construction des règles floues par l'utilisation d'une approche dite *incrémentale* ou *constructive* qui consiste à commencer par le réseau le plus simple possible, puis à ajouter des couches (des règles) pendant l'apprentissage.
- Identification de la structure de rétroaction.
- Identification des paramètres du réseau.

Les paramètres ajustables sont :

- Les centres et les largueurs des fonctions d'appartenance des variables d'entrées/sorties.
- Les poids synaptiques dans la couche de rétroaction.

L'apprentissage de ces paramètres s'effectue par l'utilisation de la rétropropagation basée sur la méthode de la descente du gradient [221].

b.2. Modèle FALCON

FALCON a été proposé par Cheng-Jian Lin et al. Il peut être comparé avec un modèle flou de base par sa structure neuronale et sa capacité d'apprentissage [222]. La topologie de FALCON est donnée par la figure 2.25.

***Couche 1* :** Chaque nœud dans cette couche correspond à une contribution d'une variable d'entrée. Sa fonction est de transmettre, directement, la valeur du signal d'entrée vers la couche suivante. Ainsi, la fonction du nœud *i* est définie comme suit :

$$f = u_i^1 = x_i \quad \& \quad a = f \tag{2.80}$$

***Couche 2* :** Chaque nœud dans cette couche correspond à une variable linguistique de l'un des variables d'entrée dans la couche 1. En d'autres termes, le degré d'appartenance qui précise les mesures dans laquelle une valeur appartient à un ensemble flou est calculé dans la couche 2. Dans le cas de la fonction gaussienne, l'opération exécutée dans cette couche est :

$$f = -\frac{1}{2} \cdot \left(\frac{u_i^2 - m_{ij}}{\sigma_{ij}} \right)^2 \quad \& \quad a = f \tag{2.81}$$

***Couche 3* :** Les noeuds dans cette couche représentent les règles d'inférence et constituent les antécédents de la base de règles floues. Les inférences sont calculées par :

$$f = \prod_{i=1}^{n} u_i^3 \quad \& \quad a = f \tag{2.82}$$

***Couche 4* :** Les nœuds dans cette couche sont appelés 'noeuds des variables linguistiques de sortie'. Chacun possède 2 modes de fonctionnement : Mode de transmission bas-haut 'down-up' et haut-bas 'up-down'. Dans le mode bas-haut, les noeuds dans la couche 4 effectuent l'opération de disjonction '**OU'** sur les règles actives qui ont les mêmes conséquences :

$$f = \prod_{j=1}^{J} u_j^4 \quad \& \quad a = \min(1, f) \tag{2.83}$$

Dans le mode de transmission haut-bas, les noeuds de la couche 4 et les liens dans la couche 5 ont le même mode de fonctionnement que celle de la couche 2, sauf que seulement un seul nœud est utilisé pour jouer le rôle de la fonction d'appartenance des variables de sortie.

***Couche 5* :** Il existe 2 types de noeuds dans cette couche. Le premier type exécute le mode de transmission haut-bas pour activer les données l'apprentissage dans le réseau. Pour ce type de nœuds :

$$f = y_k \quad \& \quad a = f \tag{2.84}$$

Le second type des nœuds exécute le mode de transmission bas-haut du signal de décision de sortie. Ces nœuds avec les liens attachés à la couche 5 vont agir comme un défuzzificateur. Si m_{ij}^5 et σ_{ij}^5 sont les centres et les largeurs des fonctions d'appartenance de sortie, respectivement, la méthode du centre de gravité est utilisée pour le calcul du signal de sortie :

$$f = \sum w_{ij}^5 \cdot u_i^5 = \sum \left(m_{ij} \cdot \sigma_{ij} \right) \cdot u_i^5 \quad \& \quad a = \frac{f}{\sum \sigma_{ij} \cdot u_i^5} \tag{2.85}$$

I-Fang Chung_2000 et al. ont proposé un algorithme d'apprentissage hybride pour une identification structurelle/paramétrique du modèle FALCON [223]. L'algorithme de

groupage '*clustering algorithm*', les AG et la rétropropagation du gradient sont les outils de conception utilisés.

L'algorithme de groupage a été utilisé par Haisheng Lin pour la recherche d'une topologie optimale du réseau FALCON [224]. Une fois fixée la structure, l'apprentissage des paramètres du réseau s'effectue par le biais de la méthode de la rétropropagation.

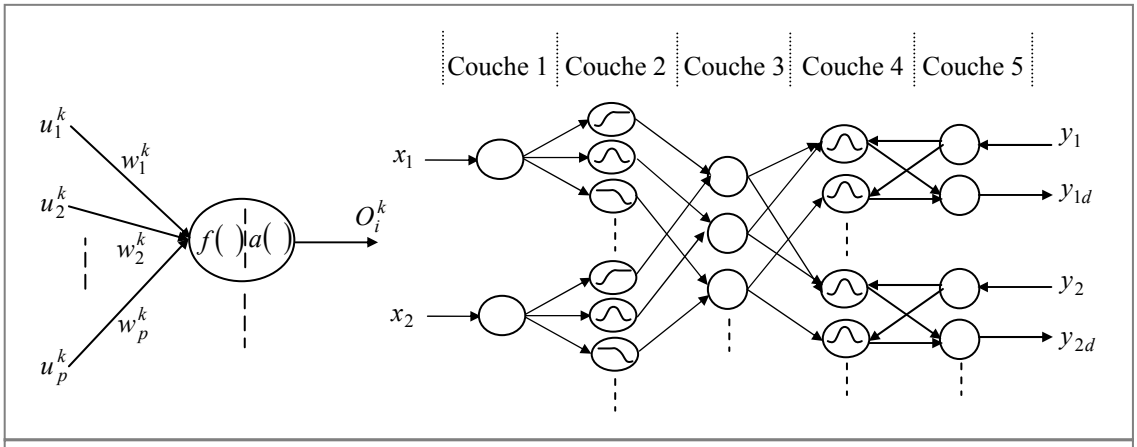


Fig. 2.25 : Topologie du modèle FALCON.

La structure FALCON est utilisée par plusieurs chercheurs dans le domaine de l'identification et de contrôle des systèmes non linéaires ainsi que dans le domaine de classification et reconnaissance des formes [223] & [225] - [226].

2.2.2.3. Modèles basés sur le raisonnement de Tsukamoto : TNFIN

Le modèle TNFIN a été proposé par Rahmat Shoureshi [227]. Il s'est inspiré du raisonnement approximatif de Tsukamoto. La forme générale d'un TNFIN à n_i variables d'entrées x_i, $(i = 1, ..., n_i)$, n_f variables linguistiques et une sortie y, est représentée par la figure 2.26. Dans cet exemple $n_i = 2$ et $n_f = 2$, respectivement. Le nombre des règles dénoté n_r. est égal à 4. Par la suite, une description de fonctionnement du modèle TNFIN est donnée.

***Couche 1* :** Couche de fuzzification. La fonction d'activation est de forme en cloche 'bell-shaped' caractérisée par les paramètres a_{ij} et b_{ij}. La sortie du neurone l est :

$$O_l^1 = O_{ij}^1(x_i) = \frac{1}{1 + \left(\dfrac{x_i - a_{ij}}{b_{ij}}\right)^2} \qquad (1 \le i \le n_i, \quad 1 \le j \le n_f) \qquad (2.86)$$

avec $l = (i-1) * n_i + j$.

***Couche 2* :** Couche d'inférence. L'opérateur de conjonction **ET** est utilisé pour calculer la sortie du neurone k :

$$O_k^2 = \prod_i O_{ij}^1(x_i) \qquad \left(1 \le i \le n_i, \quad 1 \le j \le n_f, \quad 1 \le k \le n_r\right) \tag{2.87}$$

***Couche 3* :** Couche de normalisation. La sortie normalisée du neurone k de cette couche est donnée par :

$$O_k^3 = \frac{O_k^2}{\sum_k O_k^2} \qquad \left(1 \le k \le n_r\right) \tag{2.88}$$

***Couche 4* :** Couche de défuzzification. Les fonctions d'appartenance sont aussi en forme de cloche :

$$\mu_B(y) = \frac{1}{1 + \left(\frac{y-c}{d}\right)^2} \tag{2.89}$$

Selon le nombre des règles défini dans la couche 2, la sortie est partitionnée en $n_r/2$ variables linguistiques $\{B_1, B_2, ..., B_{n_r/2}\}$. Le raisonnement flou de Tsukamoto est utilisé dans la phase de défuzzification comme indiqué par la figure 2.27. La sortie défuzzifiée de la k^{eme} règle est :

$$y_k = \begin{cases} c_k - d_k \sqrt{\frac{1}{O_k^3} - 1} & \text{si} \quad k = \text{impair} \\ c_k + d_k \sqrt{\frac{1}{O_k^3} - 1} & \text{si} \quad k = \text{pair} \end{cases} \tag{2.90}$$

Les paramètres $\{c_k, d_k\}$ sont utilisés pour ajuster les fonctions d'appartenance de la partie des conséquences des règles. La sortie de chaque neurone sera calculée par :

$$O_k^4 = O_k^3 \cdot y_k = \begin{cases} O_k^3 \cdot \left(c_k - d_k \cdot \sqrt{\frac{1}{O_k^3} - 1}\right) & \text{si} \quad k = \text{impair} \\ O_k^3 \cdot \left(c_k + d_k \cdot \sqrt{\frac{1}{O_k^3} - 1}\right) & \text{si} \quad k = \text{pair} \end{cases} \tag{2.91}$$

***Couche 5* :** Un seul neurone permet de calculer la somme de l'ensemble des sorties des règles. Elle est donnée par :

$$Y = O^5 = \sum_k O_k^4 \tag{2.92}$$

L'apprentissage hybride combinant la méthode de descente du gradient et la méthode des MCR est utilisé pour l'ajustement de la totalité des paramètres du réseau TNFIN.

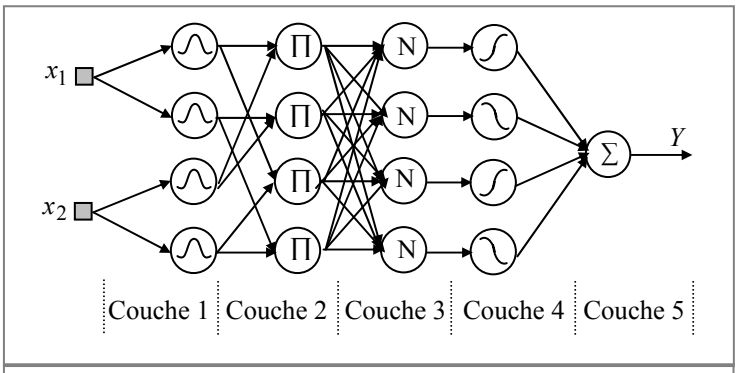


Fig.2. 26 : Architecture TNFIN.

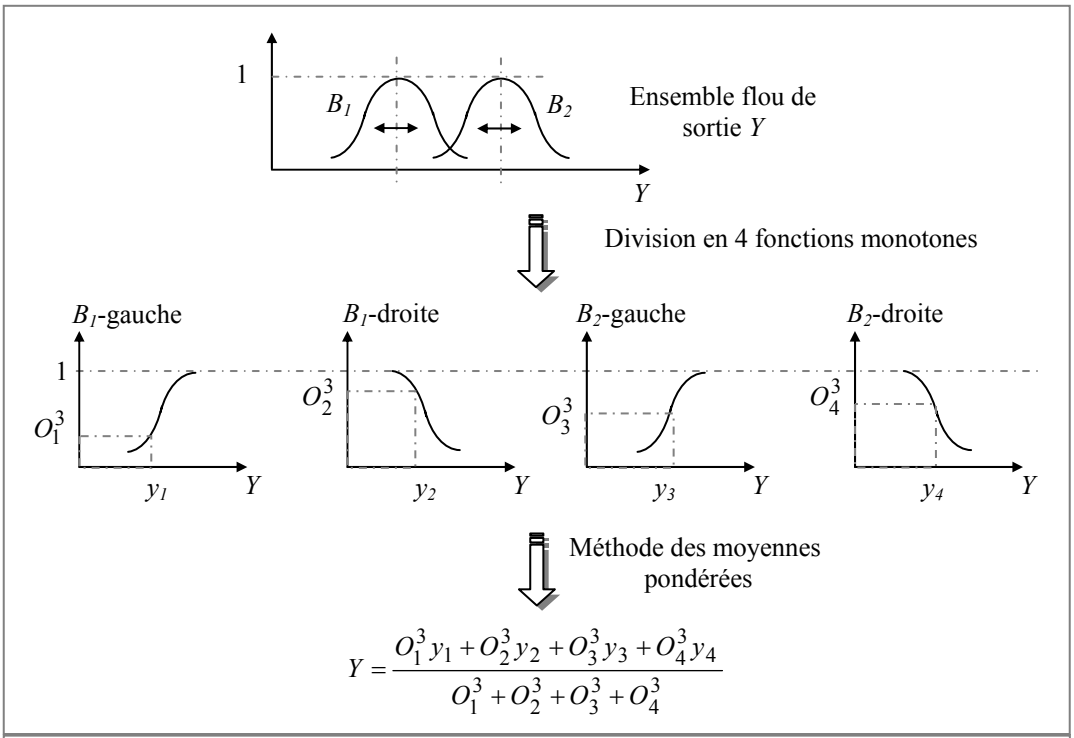


Fig. 2.27 : Méthode de défuzzification proposé par Tsukamoto.

2.3. Association série - parallèle

Un troisième grand type d'association correspond à l'utilisation des RNA et des SIF en série ou en parallèle [167].

2.3.1. Association série

Dans l'association série (Fig. 2.28), le RNA fonctionne en amont du SIF. Les variables d'entrées d'un système flou sont déterminées à partir des sorties du RNA (dans le cas où elles ne sont pas mesurables directement) ou encore un RNA effectue une tâche de classification ou de reconnaissance de formes, qui est suivit d'un système flou d'aide à la décision [169].

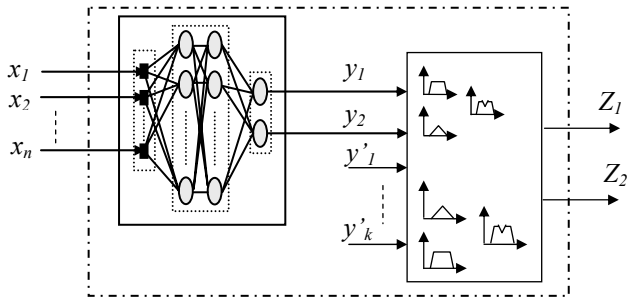


Fig. 2.28 : Exemple d'association en série d'un RNA avec un SIF.

2.3.2. Association parallèle

Dans l'association parallèle (Fig. 2.29), le RNA fonctionne en aval du SIF dans le but d'ajuster les sorties d'un système de commande floue à de nouvelles connaissances obtenues. Les variables d'entrées caractérisent l'ensemble de celles du SIF et des nouvelles obtenues. Les variables de sortie seront les erreurs sur les variables de sortie du système flou.

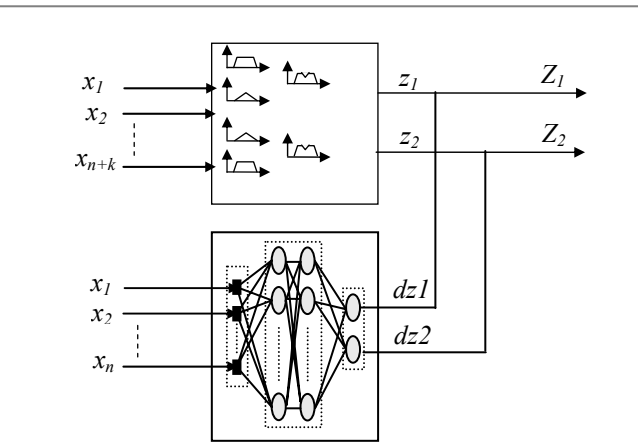


Fig. 2.29 : Exemple d'association en parallèle d'un RNA avec un SIF.

2.4. Conclusions

Suite à la proposition des ensembles flous par L. Zadeh, l'idée de réunir les SIF, les AG et les RNA a attirée l'attention des chercheurs dans le domaine des TIA. Les SIF ont la propriété d'approximer n'importe quelle fonction non linéaire. Mais, la modélisation de ces systèmes n'était pas toujours évidente. C'est pourquoi le flou adaptatif a fait son apparition. Les lois d'adaptation proposées dans la littérature (on-line & off-line) agissent sur le comportement global du SIF : Sa configuration structurelle & paramétrique.

Les TIA sont des approches visant à émuler, dans des systèmes artificiels, les capacités des espèces vivantes. L'identification de l'environnement et la prise des décisions adéquates en présence d'incertitudes et d'imprécisions sont l'objectif idéal des TIA. Les RNA et la LF sont 2 techniques qui disposent aujourd'hui d'une théorie bien fondée et qui ont été, efficacement, utilisées dans divers domaines.

Les AG sont des algorithmes d'optimisation stochastiques fondés sur les mécanismes de l'évolution génétique des espèces. Ils ont été utilisés avec succès pour obtenir des structures/paramètres optimaux pour les SIF. Cela exige une représentation des paramètres à rechercher dans une chaîne de caractères codés. Une fonction objective appropriée est nécessaire pour évaluer l'aptitude des solutions représentées par une population de chaînes.

Les RNF sont créés afin de synthétiser les avantages et de surmonter les inconvénients des RNA et des SIF. De cette manière, les algorithmes d'apprentissage peuvent être employés pour déterminer les paramètres des SIF. Ceci revient à créer ou améliorer un SIF de manière automatique, au moyen des méthodes spécifiques aux RNA. De nombreuses structures neuro - floues ont été appliquées avec succès dans de nombreux domaines en raison de leurs avantages. Cependant, l'implantation de telles structures dépend, fortement, des connaissances, à priori, sur le système et des données empiriques. Ces approches présentent l'intérêt de résoudre 2 problèmes importants des SIF :

- La détermination des univers de discours ou les partitions flous pour chaque sous ensemble flou.
- Les règles et leurs adaptations à l'environnement.

L'objectif des approches floue - génétique, neuro - floue et neuro – floue - génétique est de développer des systèmes hybrides qui réunissent les capacités d'apprentissage des RNA, d'optimisation des AG et la lisibilité et la souplesse des éléments manipulés par les SIF. Ceci s'effectue dans un sens coopératif. Une alternative d'hybridation des TIA destinée à la commande des systèmes non linéaires, hautement complexes est présentée dans le chapitre 3.

Chapitre 3

Synthèse d'une loi de commande floue à configuration minimale

Résumé :

Dans ce chapitre, nous allons développer une méthodologie de synthèse d'une loi de commande floue. L'hybridation NF est exploitée dans le développement de l'algorithme de contrôle. L'une des caractéristiques principales est la réduction de la base de règles floues. L'efficacité, la simplicité, la stabilité et la robustesse sont des facteurs significatifs intervenant dans l'indexe de performances désirées. L'optimisation de la configuration du système de contrôle s'effectue par une approche hybride. L'AGM suivi par la descente du gradient est utilisée pour réaliser les objectifs mentionnés dans la phase de conception. Dans le but de tester et d'évaluer les performances de l'algorithme proposé pour la synthèse d'une loi de commande à objectifs multiples (simple, précis, stable et robuste), nous effectuerons des simulations sur quelques systèmes non linéaires. A chaque type de système non linéaire est appliquée une forme spécifique de la loi de commande développée. Cette différence est due, principalement, à la nature et à la complexité du procédé à commander. Les résultats de simulations montrent bien l'efficacité de l'approche proposée.

Les sciences sont susceptibles d'atteindre
un état d'achèvement, de perfection,
au-delà du quel elles n'évoluent
plus guère que dans
les détails ».
Averroès.

Sommaire

3. Synthèse d'une loi de commande floue à configuration minimale

3.1. Introduction

L'application du contrôle flou à grande échelle des systèmes complexes n'est pas une tâche évidente. Pour ces systèmes, le nombre des règles de contrôle **SI - ALORS** s'explose exponentiellement. Dans beaucoup de cas, la structure est déterminée, empiriquement, en choisissant, à priori, le type du raisonnement floue de Mamdani ou de TS, le nombre des sous ensembles flous pour chaque variable et en prenant toutes les combinaisons possibles pour construire la base des règles. Le nombre des sous ensembles flous sur chacune des entrées influera, directement, sur le nombre de règles. Si nous avons l sous ensembles flous pour chacune des n variables linguistiques, nous aurons ainsi, l^n combinaisons possibles de règles. Dans les applications réelles, les systèmes sont souvent complexes et ont besoin d'une approche spécifique de modélisation et de contrôle.

Pour extraire, à partir d'une base de données, des règles floues qui complètent le savoir faire des experts, une optimisation structurelle/paramétrique du système est nécessaire. Elle consiste en :

- Sélection des variables.
- Réduction de la base des règles.
- Optimisation des paramètres correspondants.

Notre travail a été motivé par les problèmes rencontrés dans la phase de conception des SIF. Il s'agit, essentiellement, de :

- Problème de dimensionnement de la base des règles floues du SIF, c'est-à-dire, sa structure.
- Problème de configuration des paramètres de la base de connaissances.
- Problèmes de stabilité/robustesse de l'ensemble contrôleur/procédé à commander.

C'est ainsi qu'on a envisagé l'utilisation d'un algorithme hybride qui combine les méthodes d'optimisation d'ordre zéro et un pour la conception d'un contrôleur flou. Un algorithme hybride pour la construction d'un contrôleur NF est proposé dans ce travail. La simplicité de la loi de commande, l'efficacité, la stabilité et la robustesse sont des facteurs intervenant dans le critère des performances à optimiser. Le processus de mise en œuvre ainsi que les résultats de simulation sont décrits dans le reste de ce chapitre.

3.2. Méthodes de réduction de la base des règles floues : Etat de l'art

L'une des applications les plus importantes de la théorie des ensembles flous est les systèmes à base des règles floues ou les SIF [2]. Parmi les problèmes majeurs des SIF, on cite la complexité de la conception et le temps important de calcul. Ceci est dû, principalement, à la haute dimensionnalité de l'espace d'entrée, c'est-à-dire, plusieurs variables d'entrées avec plusieurs variables linguistiques pour chacune d'elles.

La réduction de la complexité de la base des règles (base de connaissances) est d'une importance capitale dans la conception des SIF, en particulier pour des applications en temps réel des contrôleurs flous. La taille de la base des règles peut être contrôlée par l'emploi des techniques de modélisation et d'identification floues.

Une des premières tentatives de réduction de la base des règles des SIF est la méthode essais - erreurs '*trial & errors*'. Elle consiste à construire une matrice d'inférence complète et d'éliminer à chaque essai une règle (resp. plusieurs règles). Garder ou éliminer une règle dépend du comportement global du système à étudier [11] & [228].

Cependant, pour les applications de contrôle des procédés, généralement, il n'y a pas assez de données pour extraire la base des règles du contrôleur. Le concepteur doit concevoir une base des règles générique, c'est-à-dire, une base qui prend en considération toutes les combinaisons possibles de valeurs des variables d'entrées. L'inconvénient majeur est que la taille de la base des règles floues augmente, exponentiellement, en fonction du nombre des entrées du contrôleur. Comme la complexité d'un système à commander augmente, il devient difficile et parfois impossible de prendre une décision précise à propos de son comportement. Deux principales familles de techniques sont proposées dans la littérature. La première consiste à fusionner des éléments de même type : Groupes, sous ensembles flous ou variables. La seconde famille est basée sur des techniques stochastiques et mathématiques [119] - [120].

Par exemple, le groupage flou '*fuzzy clustering*' est considéré comme l'une des importantes techniques de génération automatique des règles floues à partir des données numériques [229]. Dans cet algorithme, le partitionnement de l'espace est dérivé de la partition des données et conduit à une règle par groupe dans le cas général. Par conséquent, les ensembles flous résultants sont propres à une règle. Ils ne sont pas partagés par l'ensemble des règles. La taille de la base des règles peut être contrôlée via le contrôle du nombre de centres de groupage.

Une méthode d'extraction automatique et de simplification de la base des règles floues, pour établir des modèles flous interprétables à partir des données numériques, a été proposée par Min-Yu Chen et al. [230]. La technique de groupage flou et l'analyse des mesures de similarité '*similarity analysis*' entre règles sont prises en considération.

L'algorithme OLS est un processus d'apprentissage et de réduction de la base de connaissances floues qui, à partir d'une base de données d'apprentissage, construit un ensemble de règles pour ensuite conserver que les plus significatives [231] - [232]. L'OLS s'opère par le biais d'une régression linéaire utilisant les fonctions de base floues de Wang & Mendel [10].

Une technique pour la génération et la minimisation des règles floues dans le cas où les données disponibles sont insuffisantes pour l'élaboration du modèle flou a été proposée par Zaheeruddin [233]. Des règles initiales pour chaque paire de données sont générées. Les conflits de règles sont fusionnés en fonction de leur degré de pertinence (degree of soundness).

Une recherche arborescente, dont la base des règles est construite d'une manière incrémentale, a été développée par Pierre Yves Glorennec [119]. La méthode dite aussi des arbres de décisions flous consiste à concevoir des bases de règles incomplètes, partant d'une configuration plus simple et en ajoutant des fonctions d'appartenance à chaque étape. Les variables d'entrées sont, parfois, sélectionnées en fonction de leurs effets sur la sortie. Les moins influentes peuvent être retirées du modèle. Ceci peut être appliqué dans divers formalismes comme les RNA (Chapitre 2).

Un moyen simple et probablement plus efficace pour réduire la taille de la base des règles est d'utiliser le mode glissant de contrôle. L'inconvénient de cette approche est que les paramètres de la fonction de commutation doivent être sélectionnés par un expert ou conçus à travers la théorie du contrôle classique [234].

Les travaux de Kazuo Tanaka & Tadannri Taniguchi représentent une autre voie pour la réduction de la base des règles des SIF. L'analyse de stabilité au sens de Lyapunov est l'outil de base. Les conditions pour réduire le nombre de règles ont été représentées dans le

contexte LMI [235] - [236]. La méthode de réduction SVD est aussi utilisée pour la réduction automatique de la base des règles des SIF [237].

Comme mentionné auparavant, le dimensionnement des SIF et des RNF est un problème non résolu d'une manière définitive. Le concept de simplification et de réduction des bases de connaissances des SIF est une voie ouverte pour la proposition des techniques de modélisation et de contrôle simples, efficaces et robustes.

Construire des SIF employant aussi moins que possible des règles avec la garantie des performances désirées est un problème significatif qui a attiré l'attention des chercheurs dans la communauté de la logique floue [238]. Les TIA représentent une base algorithmique importante pour répondre aux besoins des concepteurs. Le choix du type de raisonnement, les contraintes conceptuelles (interprétabilité des règles, stabilité en BF, ...etc.), la méthode d'optimisation, le mode de représentation des informations et le critère des performances désirées sont des facteurs qui font la différence entre les différentes propositions [125], [155], [161] & [239] - [244].

J. Zhao et al. ont proposé d'utiliser les AG pour l'élimination des règles redondantes. Le raisonnement de TS est exploité [239]. K. Belarbi et al. ont introduit un coefficient de stabilité comme base pour minimisation des règles du type Mamdani. Le codage binaire est utilisé pour la représentation de paramètres/structure de la base des règles floues [161].

Rafael Alcalá et al. ont présenté une étude sur la façon dont les méthodes d'amélioration dites à base des approches raffinées '*basic refinement approaches*' peuvent être combinées 'hybridation'. Les approches hybrides présentent un bon compromis entre l'interprétabilité et la précision [240].

L'inclusion des termes structurels (nombre des entrées, nombre des fonctions d'appartenance, nombre des règles, ...etc.) dans le critère des performances à minimiser en plus des termes fonctionnels (erreurs de poursuite, temps de réponses, ...etc.) est aussi abordée par certains chercheurs [239] - [240], [243] & [245] - [248].

A. Soukkou et al. ont proposé un AG avec une représentation mixte : Binaire-réel-entier comme méthode d'optimisation d'un RNF. La précision 'efficacité', la réduction de la base des règles et la minimisation de l'énergie de commande sont des facteurs significatifs dans l'indexe de performances [246]. La figure 3.1 représente la matrice ainsi que le codage approprié d'une partie des conséquences des règles d'inférence. Cette forme a été proposée par V. Matellan et al. [249].

Feng Wan et al. ont présenté une approche de calcul géométrique pour la détermination d'un nombre minimal de règles tout en gardant un niveau acceptable de précision du SIF résultant [250].

Francesco Cupertino et al. & Vincenzo Giordano et al. ont développé une stratégie d'apprentissage hybride pour la conception d'un contrôleur flou adaptatif. L'AG est utilisé pour l'optimisation des paramètres du contrôleur, y compris le nombre des fonctions d'appartenance et les règles d'inférence. Un ajustement plus fin des paramètres des conséquences 'singleton' est effectué par l'approche énergétique de Lyapunov [251] - [252].

Chi-Ho Lee et al. ont présenté 2 phases pour l'optimisation d'un SIF. A chaque règle floue est associé un poids déterminant le degré de contribution de la règle à l'ensemble de la base des règles [253] :

- Dans la première phase, un algorithme évolutionnaire est appliqué pour trouver la forme réduite de la base des règles ainsi que les facteurs d'échelles pour les variables d'entrées.
- Le principe de l'entropie maximum est utilisé pour déterminer les poids des règles dans la deuxième phase de l'algorithme.

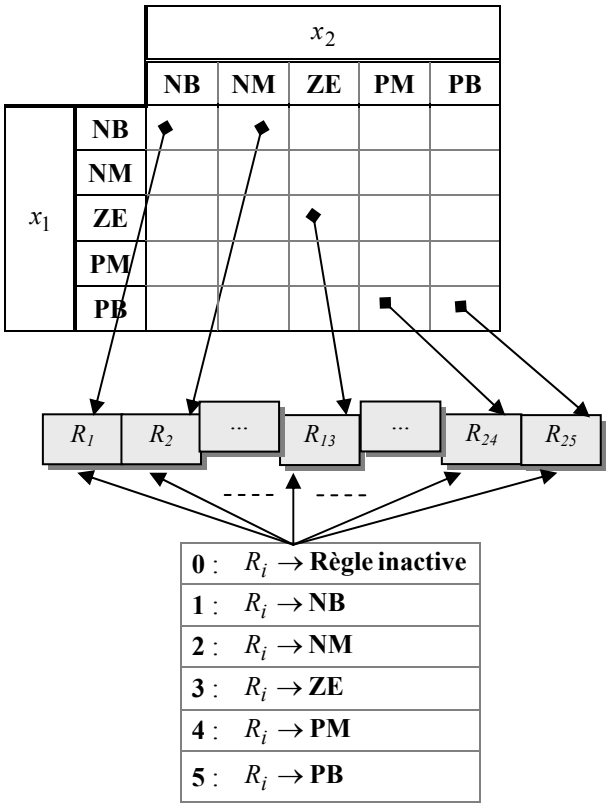


Fig. 3.1 : Codage entier des règles floues.

Des structures NF avec différents algorithmes d'apprentissage sont, habituellement, représentées sous forme de PMC (Chapitre 2). L'inconvénient de cette approche est le manque de moyens systématiques pour l'établissement des configurations (des topologies) optimales des réseaux. Le nombre de couches et les interconnexions entre les nœuds de ces couches sont encore un art.

Une structure hiérarchique (Fig. 3.2), suggérée par Raju, permet la réduction d'une manière considérable du nombre des règles [254]. Jamshidi a proposé l'idée de fusionner plusieurs entrées en une pour réduire la taille de la base des règles (méthode de fusion sensorielle) [255]. Dans [256], l'ensemble des paramètres de la technique de fusion sensorielle est optimisé par les AG. L'utilisation de la structure hiérarchique pour la conception des SIF a été reportée dans [238] & [257] - [260]. La topologie hiérarchique du SIF permet de réduire la taille de la base des règles. Le nombre de règles floues utilisées dans un SIF hiérarchique est proportionnel au nombre des variables d'entrées.

Chaque étage a une base de connaissances élémentaires. Sa sortie sera obtenue par la fonction, généralement, non linéaire suivante :

$$u_i(t) = f_i(e_i(t), e_{i+1}(t)) \qquad i = 1,\ 2,\ ...,\ m \tag{3.1}$$

Cette sortie est l'entrée de l'étage $i+1$, ainsi de suite. Par conséquent, on peut déduire la sortie de la structure hiérarchique du SIF. Elle est donnée par :

$$u(t) = \underbrace{f_m}_{\text{Etage } m}\left(\underbrace{f_{m-1}}_{\text{Etage } m-1}\left(\underbrace{f_{m-2}}_{\text{Etage } m-2}\left(\underbrace{f_{m-3}}_{\text{Etage } m-3}(\ldots), e_{m-2}(t)\right), e_{m-1}(t)\right), e_m(t)\right) \tag{3.2}$$

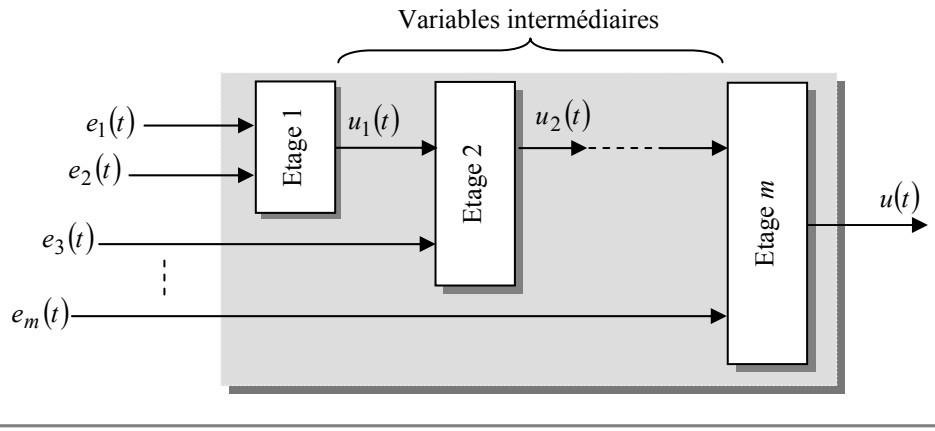


Fig. 3.2 : Structure hiérarchique d'un SIF.

Notre travail vise à l'élaboration d'une stratégie de commande efficace, robuste et stable pour des processus hautement non linéaires, intégrant une réduction de la complexité de cette stratégie. L'objectif est, donc, de développer une approche systématique par l'hybridation des TIA.

3.3. Synthèse de la loi de commande

La modélisation floue, basée sur la théorie des ensembles flous, peut être considérée comme une approche qui sert à modéliser les processus complexes mal définis, fortement non linéaires ou difficiles à analyser, mathématiquement [2].

Les formalismes les plus utilisés pour les modèles flous sont ceux de Mamdani, basé sur l'expérience et de TS, plus approprié pour une approche basée sur les données numériques [5] & [22]. D'autres formalismes existent dans la littérature, mais, leur utilisation est beaucoup plus restreinte [27]. La différence, entre ces modes de raisonnement, réside dans la forme d'évaluation des conséquences des règles floues.

Dans le cadre de ce travail, reprenons une règle linguistique R^i pour un système à n entrées et une sortie, de la forme générale suivante :

$$\underbrace{R^i}_{i=1,\ldots,N} : \textbf{SI} \quad \overbrace{\underbrace{\left(\tilde{x}_1 \textbf{ est } A_1^i\right)}_{\text{Proposition floue}} \ldots \textbf{ET} \left(\tilde{x}_n \textbf{ est } A_n^i\right)}^{\text{Prémisse}} \quad \textbf{ALORS} \quad \overbrace{u \textbf{ est } \underbrace{F^i(\tilde{x}_1, \ldots, \tilde{x}_n)}_{\text{Conséquence}}}^{\text{Conclusion}} \tag{3.3}$$

où les différents symbôles sont définis comme suit :

N	:	Dénote le nombre des règles du modèle.
$[\tilde{x}_1, \ldots, \tilde{x}_n]^T \in \tilde{X} \in \Re^n$	:	Est le vecteur des variables d'entrées dans le domaine normalisé $[-1, +1]$, c'est-à-dire, $\tilde{x}_i = x_i / \tilde{K}_{x_i}$ où $\tilde{K}_{x_i}$ est le facteur d'échelle associé à la variable x_i.
$u \in U \in \Re$	:	Est la variable de sortie.

$\widetilde{X}$ & U : Correspondent, respectivement, aux univers de discours des variables d'entrée et de sortie.

$A_1^i \sim A_n^i$: Sont les termes linguistiques caractérisant les sous ensembles flous des entrées.

$F^i(\widetilde{x}_1, ..., \widetilde{x}_n)$: Représente la partie conséquence de la i^{eme} règle.

Le tableau 3.1 résume les différentes configurations possibles du modèle donné par l'expression (3.3) en fonction des valeurs '*formes*' des conséquences.

Tab. 3.1 : Différentes combinaisons des modèles flous.

Conséquence	Valeur & forme	Type
$\underbrace{F^i}_{i=1,...,N}(x_1, ..., x_n)$	Ensemble flou B^i	• **Type A**.
	Ensemble flou $B^i \times \widetilde{c}_F^i$	• **Type B**.
	q_0^i	• **Type C**.
	$\sum_{l=0}^{l=n}\left(q_l^i \cdot \widetilde{x}_l\right)$	• **Type D**.
	Fonction non linéaire	• **Type E**.
	$\left(B^i\right)\times\left(\sum_{l=1}^{l=n}\left(q_l \cdot \widetilde{x}_l\right)\right)$	• **Type F**.
	$\left(B^i\right)\widetilde{\wedge}\left(\sum_{l=1}^{l=n}\left(q_l \cdot \widetilde{x}_l\right)\right)\widetilde{\wedge}\left[\widetilde{c}_F^i\right]$	• **Type G**.
	$\sum_{l=1}^{l=n}\left(q_l^i \cdot x_l(t)\right)\widetilde{\wedge}\left\{\left[\widetilde{c}_F^i\right]\widetilde{\wedge}\left[\widetilde{w}_i\right]\right\}$	• **Type H**.

Les paramètres $\widetilde{c}_F^i \in \{0,1\}$ et $\widetilde{w}_i \in \Re$ représentent, respectivement, le facteur d'incertitude '*certainty factor*' et le poids (facteur de contribution) de la i^{eme} règle. $\widetilde{w}_i$ sont des gains statiques assurant (ou participant à) la stabilité et la robustesse du système en BF. Si $\widetilde{c}_F^i = 1$, la i^{eme} règle est active sinon si $\widetilde{c}_F^i = 0$, la règle est inactive. $\widetilde{\wedge}$ est un opérateur de conjonction modélisant les facteurs d'incertitude/poids de la règle floue. Dans le reste de cette étude, l'opérateur $\widetilde{\wedge}$ peut être implémenté par le produit arithmétique.

Le concept d'associer à chaque règle floue un poids '*rule weight*' est utilisé, généralement, dans le domaine de classification [241] & [261] - [264]. Dans ce travail, le poids de la règle représente le facteur de participation 'de contribution' de la règle à la *stabilité* globale du système en BF. Les types mentionnés dans le tableau 3.1 seront détaillés ci-dessus.

- **Type A :** Modèle de Mamdani avec une base des règles complète.
- **Type B :** Modèle de Mamdani avec un facteur d'incertitude $\widetilde{c}_F^i$, c'est-à-dire à base des règles réduite [161] & [253].
- **Type C :** TS d'ordre zéro (TS0).
- **Type D :** TS linéaire. Si $q_0^i = 0$, le modèle correspond à un TS d'ordre un (TS1).

- **Type E :** SIF d'ordre supérieur [265] - [266]. Les auteurs ont proposé une collection des contrôleurs non linéaires du type :
  - Biquadratique - 1 : $TS0 + c_3 \cdot x_1^2 + c_4 \cdot x_2^2 + c_5 \cdot x_1 \cdot x_2$.
  - Biquadratique - 2 : $TS0 + c_3 \cdot x_1 \cdot x_2$.
  - Trilinéaire : $c_0 + c_1 \cdot x_1 + c_2 \cdot x_2 + c_3 \cdot x_3$.
  - Triquadratique - 1 : $\text{Trilinéaire} + c_4 \cdot x_1^2 + c_5 \cdot x_2^2 + c_6 \cdot x_3^2$.
  - Triquadratique - 2 : $\text{Trilinéaire} + c_4 \cdot x_1 \cdot x_2 + c_5 \cdot x_1 \cdot x_3 + c_6 \cdot x_2 \cdot x_3$.
- **Type F :** Ce travail est inspiré des travaux de Hao Ying dans [267] - [268]. B^i est le centre de gravité de l'ensemble flou de sortie (singleton) associé à la i^{eme} règle [269].
- **Type G :** Ce modèle est développé par A. Soukkou et al. [246]. Il s'agit d'une hybridation entre le modèle de Mamdani et le modèle de TS1 avec une base de règles réduite. Ce type est une amélioration du travail précédent. Un AG multiobjectif (MGA) avec un codage mixte : Binaire-réel-entier est développé pour l'optimisation de la base de connaissances floues. Les résultats de simulation montrent bien l'efficacité de l'approche proposée.
- **Type H :** La forme des règles proposée dans ce travail. Un contrôleur flou développé avec cette forme de règles est utilisé pour le contrôle de système de la température d'un bain d'eau et le pendule inversé, respectivement, est présenté par A. Soukkou et al. dans [270]. L'AG avec un codage mixte binaire - réel est utilisé pour l'optimisation de l'ensemble des paramètres figurant le contrôleur. A la fin de l'exécution de l'algorithme, nous avons constaté que seulement 8% des règles sont suffisantes pour assurer les performances désirées. Dans un autre travail dans le même cadre, une architecture ANFIS modifiée adaptée à cette forme de règles a été étudiée en détail avec différentes formes d'hybridation des méthodes d'optimisation : AG, MCR et de la rétropropagation du gradient [271] - [272]. Dans [272], 4% de la base des règles, c'est-à-dire 1/25, permet de donner de bons résultats (efficacité et robustesse) pour l'exemple de régulation du système du pendule inversé. Dans l'exemple de poursuite d'une trajectoire sinusoïdale, 8% de la base des règles, c'est-à-dire, 2/25 sont suffisantes pour réaliser l'objectif de la commande. Dans [271], les auteurs ont présenté une comparaison entre 2 alternatives de conception des SIF. La première est le cas où on fait associé à la base des règles globale un gain unique. La deuxième est celle présentée par la forme de règles proposée. Une règle présente un modèle local. L'effet collectif de l'ensemble des modèles locaux présente le comportement global du SIF. La dernière forme présente de meilleures performances. Le point commun entre ces 2 formes est le nombre minimal de règles floues. Les comparaisons ont été effectuées sur l'exemple du pendule inversé.

Dans cette partie de la thèse, nous intéressons à un type particulier des RNA, abordé au chapitre précédent, le RBF. S. R. Jang et al. ont montré qu'un RBF est, strictement, équivalent à un SIF du type TS0 sous certaines conditions [186] :

- **Condition 1 :** Le nombre des couches dans le réseau RBF correspond au nombre des règles floues.
- **Condition 2 :** Les fonctions d'appartenance sont choisies du type gaussiennes.

- **Condition 3 :** L'opérateur *t-norme* utilisé pour le calcul des inférences est effectué par le produit arithmétique.
- **Condition 4 :** Les conséquences des règles sont de type singleton. La sortie du réseau sera défuzzifiée par l'usage d'une méthode de défuzzification.

Kwang Bo Cho et al. ont proposé des extensions pour manipuler avec des SIF du type TS à conclusions polynomiales [188].

Définitions

Le modèle RBF a comme particularité que ses fonctions - noyaux sont locales, c'est-à-dire, qu'elles ne donnent des réponses utiles que pour un domaine de valeurs restreint. Ce domaine est défini autour d'un point, le centre (ou noyau). En général, chaque fonction - noyau est décrite par 2 paramètres : La position de son centre et la taille du domaine. Pour approximer un comportement donné, les fonctions - noyaux sont assemblées pour couvrir de leurs domaines l'ensemble des données d'entrée. Ces fonctions sont, ensuite, pondérées et leurs valeurs sont sommées pour produire une valeur de sortie. Les fonctions - noyaux peuvent être définies, de manière générale, par :

$$\Phi(X,c)=\Phi(\|X-c\|,\Sigma) \tag{3.4}$$

$\Phi(\cdot)$ *est une fonction-noyau.* $\|\cdot\|$ *est une norme (en général, la norme Euclidienne ou la distance de Mahalanobis).* c *et* $\sum$ *sont le centre et l'écart type de la fonction noyau, respectivement. Dans le cas de la fonction gaussienne, elle est décrite par la relation* :

$$\Phi(X,c)=\exp\left(-\frac{\lambda}{d_m^2}\|X-c_i\|^2\right) \tag{3.5}$$

λ *est le nombre des centres* c_i *(le nombre des fonctions de base) et* d_m *est la distance maximale entre les centres sélectionnés. De manière générale, la largeur des fonctions radiales doit être comprise entre le minimum et le maximum de la distance entre les points de la base de données d'apprentissage (entraînement). Une règle empirique est utilisée dans le cas où une même valeur d'écart type est choisie pour toutes les fonctions. La variance de la RBF est donnée par* [273] *:*

$$\sigma=\frac{d_m}{\sqrt{2.\lambda}} \tag{3.6}$$

Dans certains travaux, le paramètre σ est choisi de tel sorte que [73] :

$$\sigma_i=\beta\cdot|c_i| \tag{3.7}$$

β est un paramètre d'adaptation de la fonction gaussienne, déterminé empiriquement ou par des méthodes d'optimisation [73].

L'architecture RBF est bien adaptée à l'analyse multimodèle dont chacun des neurones de la couche cachée correspond à un modèle local. La forme des règles du contrôle flou développée dans ce travail est représentée par un modèle TS1 avec des facteurs d'incertitude et de contribution associés à chaque règle d'inférence. Ainsi, le contrôleur résultant est une collection de sous modèles flous où chaque sous modèle m_{odi}, $(i = 1, ..., N)$ est une règle floue (Fig. 3.3).

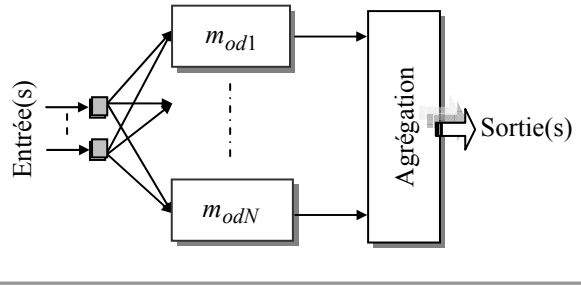


Fig. 3.3 : SIF multimodèle.

La structure d'un PMC (multimodèle) du type RBF-flou proposée dans ce travail est schématisée par la figure 3.4. L'effet combiné des 5 couches de la figure 3.4 du réseau constitue le mode fonctionnel du système de contrôle.

***Couche 1* :** Couche d'entrée ou de *distribution.* Les neurones de distribution transmettent les signaux d'entrées vers les neurones de la couche cachée.

***Couche 2* :** Couche d'inférence. Chaque unité de la couche cachée correspond à une règle floue. Les neurones de la deuxième couche sont dits unités du champ récepteur local '*local receptive field units*' avec des fonctions d'activation gaussiennes :

$$\Phi_j = \Phi_j\left(\left\|\tilde{X} - c_j\right\|,\ \sigma_j\right) = \exp\left(-\frac{\left\|\tilde{X} - c_j\right\|^2}{\sigma_j^2}\right) = \exp\left(-\sum_{i=1}^{n}\frac{\left(\tilde{x}_i - c_j^i\right)^2}{{\sigma_j^i}^2}\right) \qquad j = 1,\ 2,\ ...,\ N \tag{3.8}$$

Φ_j est la sortie du neurone j de la deuxième couche cachée. $c_j = \left[c_j^1 \quad c_j^2 \quad ... \quad c_j^n\right]^T \in \Re^n$ et $\sigma_j = \left[\sigma_j^1 \quad \sigma_j^2 \quad ... \quad \sigma_j^n\right]^T \in \Re^n$ sont les vecteurs centre et écart-type « dilatation» de la gaussienne du nœud j, respectivement. L'expression (3.8) sera reformulée par :

$$\Phi_j = \prod_{i=1}^{n} \underbrace{\exp\left(-\frac{\left(\tilde{x}_i - c_j^i\right)^2}{{\sigma_j^i}^2}\right)}_{\mu_{A_i^j}^i(\tilde{x}_i)} = \prod_{i=1}^{n} \mu_{A_i^j}^i\left(\tilde{x}_i\right) \tag{3.9}$$

L'expression (3.9) représente l'inférence de la j^{eme} règle calculée par le produit arithmétique des degrés d'appartenance $\mu^j_{A_1^j}(\tilde{x}_1), \mu^j_{A_2^j}(\tilde{x}_2), ..., \mu^j_{A_n^j}(\tilde{x}_n)$.

Remarque :

Le modèle du neurone flou n'a pu être formulé que grâce à la propriété de la fonction exponentielle : $\exp\left(\sum_{i=1}^{n}(\alpha_i)\right) = \prod_{i=1}^{n}\exp(\alpha_i)$.

***Couche 3* :** Chaque nœud représente un switch **Tout Ou Rien** (**TOR**). Le TOR permet d'activer ou de désactiver la règle correspondante. Sa formule mathématique est décrite par :

$$TOR_j = \begin{cases} \Phi_j & \text{si} \quad \tilde{c}_F^j = 1 \\ 0 & \text{si} \quad \tilde{c}_F^j = 0 \end{cases} \tag{3.10}$$

***Couche 4* :** Neurones de TS. Cette couche s'appelle la couche des conséquences des règles. La sortie du neurone *j* de la couche 4 est calculée par la relation suivante :

$$\begin{aligned} \Psi_j &= TOR_j \cdot \tilde{F}^j(\tilde{x}_1, ..., \tilde{x}_n) \\ &= TOR_j \cdot \left(q_1^j \cdot \tilde{x}_1 + q_2^j \cdot \tilde{x}_2 + ... + q_n^j \cdot \tilde{x}_n\right) \end{aligned} \tag{3.11}$$

***Couche 5* :** Couche de sortie. La sortie du réseau est définie telle que :

$$u^*(t) = \sum_{l=1}^{N} \tilde{w}_l \cdot \Psi_l \tag{3.12}$$

Par substitution des expressions (3.9) & (3.11) dans (3.12), la sortie du réseau contrôleur sera calculée par :

$$u^*(t) = \sum_{l=1}^{N} \tilde{w}_l \cdot TOR_l \cdot \tilde{F}^l(\tilde{x}_1, ..., \tilde{x}_n) \tag{3.13}$$

Le tableau 3.2 représente des cas particuliers du contrôleur (3.13) et ceci suivant la valeur du polynôme du TS $\tilde{F}^l(\tilde{x}_1, ..., \tilde{x}_n)$ et le vecteur poids des règles.

En résumé, la forme générale de la loi de commande est donnée par la relation (3.14). Par la suite, un contrôleur PID et ses dérivés P, PI et PD représentent des cas particuliers de cette loi. Le choix des variables d'entrée est un facteur significatif dans cette dernière conséquence.

$$u^*(t) = \sum_{i=1}^{n} \Theta_i(\tilde{c}_F, \tilde{x}, q, \tilde{w}) \cdot \tilde{x}_i \tag{3.14}$$

$$\Theta_i(\tilde{c}_F, \tilde{x}, q, \tilde{w}) = \sum_{l=1}^{N} \left[\left\{ \tilde{c}_F^l \times \tilde{w}_l \right\} \cdot \left\{ \prod_{r=1}^{n} \exp\left(-\frac{\left(\tilde{x}_r - c_l^r\right)^2}{{\sigma_l^r}^2} \right) \right\} \cdot q_i^l \right] \tag{3.15}$$

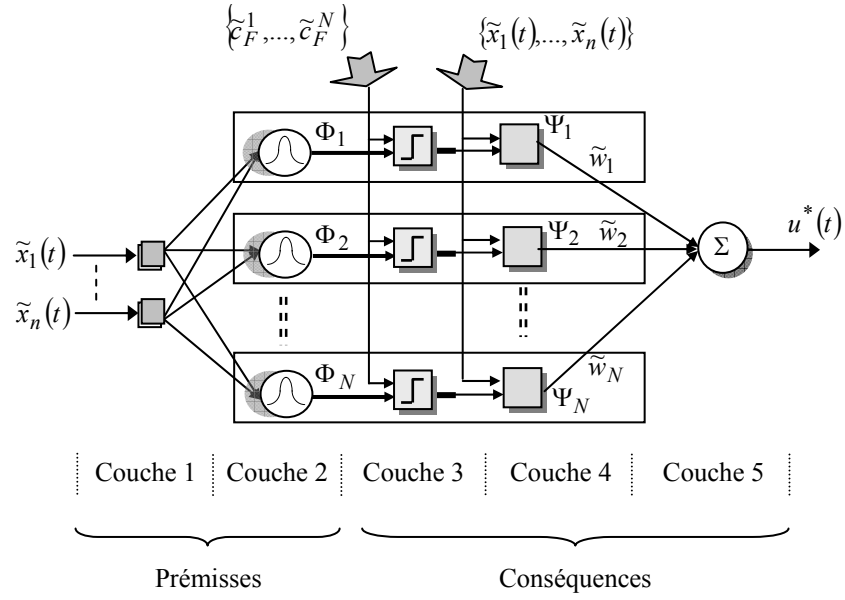


Fig. 3.4 : Structure du réseau contrôleur.

Tab. 3.2 : Caractéristiques particulières du contrôleur proposé.

$\tilde{F}^l(\tilde{x}_1,...,\tilde{x}_n)$	$\tilde{w}_l$	Type de contrôleur
1	$\tilde{w}_l$	TS0 à base réduite.
C^{te}	1	TS0 à base réduite.
C^{te}	$\tilde{w}_l$	TS0 à base réduite et poids associés aux règles.
$\sum_{i=1}^{n} q_i^l \cdot \tilde{x}_i$	1	TS1 à base réduite.
$\sum_{i=1}^{n} q_i^l \cdot \tilde{x}_i$	$\tilde{w}_l$	TS1 à base réduite et poids associés aux règles.

L'ajout d'un *processeur de normalisation* [**PN**] dans le réseau de la figure 3.4 nous permet d'avoir un réseau RBF-flou normalisé (Fig. 3.5). Le facteur de normalisation $\gamma_l(\tilde{x})$ de la l^{eme} règle est calculé par :

$$\gamma_l(\tilde{x}) = \frac{1}{\prod_{r=1}^{n} \exp\left(-\frac{(\tilde{x}_r - c_l^r)^2}{{\sigma_l^r}^2}\right)} \tag{3.16}$$

La valeur de la sortie du réseau contrôleur représentée par l'expression (3.14) sera multipliée par le facteur $\gamma(\tilde{x})$. L'expression (3.15) devient dans ce cas :

$$\Theta_i(\tilde{c}_F, \tilde{x}, q, \tilde{w}) = \frac{\sum_{l=1}^{N}\left[\left\{c_F^l \times \tilde{w}_l\right\}\cdot\left\{\prod_{r=1}^{n}\exp\left(-\frac{\left(\tilde{x}_r - c_l^r\right)^2}{{\sigma_l^r}^2}\right)\right\}\cdot q_i^l\right]}{\sum_{l=1}^{N}\left\{\underbrace{\prod_{r=1}^{n}\exp\left(-\frac{\left(\tilde{x}_r - c_l^r\right)^2}{{\sigma_l^r}^2}\right)}_{\rho_i(\tilde{x})}\right\}} \tag{3.17}$$

La sortie globale du réseau sera donnée par la formule suivante :

$$u^* = \frac{\sum_{l=1}^{N}\rho_l(\tilde{x})\cdot\overbrace{\left\{c_F^l \times \tilde{w}_l\right\}}^{f_l}\cdot\sum_{i=1}^{n}q_i^l\cdot\tilde{x}_i}{\sum_{l=1}^{N}\rho_i(\tilde{x})} \tag{3.18}$$

Le terme $[\mathrm{PN}] = 1\Big/\sum_{l=1}^{N}\left\{\prod_{r=1}^{n}\exp\left(-\left(\tilde{x}_r - c_l^r\right)^2\Big/{\sigma_l^r}^2\right)\right\}$ est le facteur de normalisation. Dans ce cas, on parle de réseau normalisé.

En introduisant la notion de fonctions floues de base [8], la sortie (3.18), peut être réécrite sous la forme compacte suivante :

$$u^* = \rho^T(\tilde{x})\cdot\theta \tag{3.19}$$

$\theta = [f_1, ..., f_N]^T$ est le vecteur des paramètres à ajuster. $\beta(\tilde{x}) = [\beta_1(\tilde{x}), ..., \beta_N(\tilde{x})]^T$ est le vecteur des fonctions floues de base dont chacune des composantes est donnée par :

$$\beta_k(\tilde{x}) = \frac{\rho_k(\tilde{x})}{\sum_{l=1}^{N}\rho_i(\tilde{x})} \qquad k = 1, ..., N \tag{3.20}$$

Pour ne pas avoir de pertes des données d'entrées, les fonctions floues de base sont construites de sorte que :

$$\sum_{l=1}^{N}\rho_i(\tilde{x}) \neq 0 \qquad \forall\, \tilde{x} \in \tilde{X} \tag{3.21}$$

Le modèle non normalisé de la figure 3.4 n'est pas concerné par la condition (3.21), il suffit de trouver les règles actives avec des paramètres adéquats.

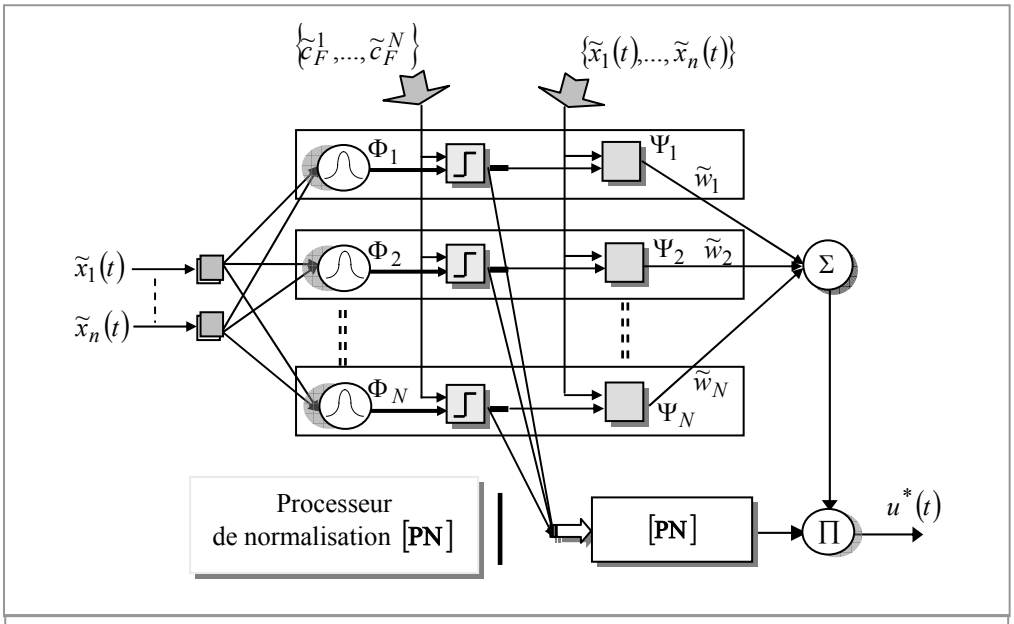


Fig. 3.5 : Structure normalisée du réseau contrôleur.

L'expression (3.15) (resp. (3.17)) est la partie non linéaire que présente le SIF. Les coefficients de TS q_i^j dans (3.15) (resp. (3.17)) et le vecteur poids des règles $\tilde{w} = [\tilde{w}_1, ..., \tilde{w}_N]^T$ sont des paramètres à identifier par un processus d'apprentissage adéquat.

La similarité du contrôleur proposé par la loi de commande conventionnelle PID sera mesurée par le type des entrées du contrôleur. L'erreur et la variation de l'erreur prises comme variables d'entrées sont des cas particuliers qui sont, couramment, utilisées pour la génération de la loi de commande.

L'expression $\tilde{F}^j(\tilde{x}_1, ..., \tilde{x}_n) = \sum_{i=0}^{n} q_i^j \cdot \tilde{x}_i$ présente une association d'un ensemble de contrôleurs linéaires, y compris le PID comme cas particulier (avec $q_0^j = 0$, $(j = 1, ..., N)$) [267] - [268].

Le tableau 3.3 illustre les différents contrôleurs conventionnels : P, PI, PD et PID, ainsi que les gains discrets des actions de proportionnalité K_P^d, de dérivation K_D^d et d'intégration K_I^d. $y^d(kT)$ et $y(kT)$ caractérisent la trajectoire de référence (désirée) et la sortie du processus à commander à l'instant d'échantillonnage kT, respectivement.

Selon les données du problème à traiter, la variable de contrôle peut être l'action directe $u^*(t)$ ou sa variation $\Delta u^*(t)$.

Malgré sa popularité dans l'industrie et sa facilité d'implantation, l'inconvénient du contrôleur PID est le choix de ces gains K_P^d, K_D^d et K_I^d. En conséquence, beaucoup de techniques de réglage des paramètres basées sur différentes approches, tels que la méthode de Ziegler-Nichols et le lieu géométrique des racines sont encore sujet de plusieurs travaux de recherche et d'implantation [274] - [276].

Des méthodes d'optimisation, telles que celles de Cohen & Cone ou les méthodes de Lopez sont, aussi, proposées [277]. Yougho Lee a présenté un bref historique sur les différentes méthodes de calcul de ces coefficients. Il a aussi proposé une méthode de calcul basée sur le développement en série de Mc Laurin [278].

Ces techniques permettent d'obtenir les coefficients de différentes actions en se basant sur la minimisation d'un critère de performances désirées (IAE, ITAE, ...etc.). M. A. Rodrigo et al. ont présenté des stratégies de conception des contrôleurs PID basées sur les approches : ITAE, IMC et Cohen & Cone. Les résultats obtenus ont été comparés à ceux obtenus en employant un contrôleur PID flou. La stratégie traditionnelle de conception essais - erreurs a été utilisée pour la conception du PID flou [115].

Tab. 3.3 : Caractéristiques des contrôleurs conventionnels : P, PI, PD & PID.

Type	Variables d'entrées	Loi de commande
P	$x_1(kT)=y^d(kT)-y(kT)$	$u_P^*(kT)=K_P^d\cdot x_1(kT)$
PI	$\begin{cases}x_1(kT)=y^d(kT)-y(kT)\\ x_2(kT)=x_1(kT)-x_1((k-1)\cdot T)\end{cases}$	$\Delta u_{PI}^*(kT)=\frac{K_P^d}{T}\cdot x_2(kT)+K_I^d\cdot x_1(kT)$ $u_{PI}^*(kT)=u_{PI}^*((k-1)\cdot T)+\Delta u_{PI}^*(kT)$
PD	$\begin{cases}x_1(kT)=y^d(kT)-y(kT)\\ x_2(kT)=x_1(kT)-x_1((k-1)\cdot T)\end{cases}$	$u_{PD}^*(kT)=K_p^I\cdot x_1(kT)+K_D^d\cdot x_2(kT)$
PID	$\begin{cases}x_1(kT)=y^d(kT)-y(kT)\\ x_2(kT)=x_1(kT)-x_1((k-1)\cdot T)\\ x_3(kT)=x_2(kT)-x_2((k-1)\cdot T)\end{cases}$	$\Delta u_{PID}^*(kT)=\frac{K_P^d}{T}\cdot x_2(kT)+K_I^d\cdot x_1(kT)+\frac{K_D^d}{T}\cdot x_3(kT)$ $u_{PID}^*(kT)=u_{PID}^*((k-1)\cdot T)+\Delta u_{PID}^*(kT)$

L'utilisation des TIA est la tendance actuelle dans le développement des systèmes de contrôle efficaces et robustes. Wei-Der Chang et al. ont présenté une méthode adaptative de réglage d'un contrôleur PID multivariable [279]. Les éléments de base sont des neurones adaptatifs. Le réglage s'effectue par une procédure temps réel (on-line).

Pour améliorer les performances de la commande des procédés complexes, un contrôleur non linéaire de type PID a été proposé par TU Diep Cong Thanh et al. [280]. Sylvie Galichet & Laurent Foulloy ont montré qu'un contrôleur flou basé sur le raisonnement approximatif de Mamadani ou de TS est, strictement, équivalent à un contrôleur PI avec quelques conditions restrictives [281].

Dans [70], un contrôleur PID robuste à caractère NF a été développé. Sa capacité de compensation de variations des paramètres et sa robustesse sont prouvées à travers un exemple d'application. Les capacités d'apprentissage des RNA et les non linéarités des lois d'adaptation des paramètres du PID sont les éléments de base de la méthode proposée.

Les AG sont aussi des concurrents puissants dans le domaine de conception des contrôleurs PID robustes [113] & [282]. A. Soukkou et al. ont proposé un contrôleur robuste du type PI pour la commande de la température d'un bain. Un chromosome de l'AG codifie l'ensemble des gains des actions proportionnelle et intégrale, respectivement. Le codage réel avec l'opérateur de mutation non uniforme a été développé [282].

Les résultats obtenus sont appréciables. La structure du chromosome est illustrée par le schéma de la figure 3.6. Le mode de représentation et l'index de performances représentent les 2 facteurs qui font la différence entre les travaux proposés. Wei-Der Chang a utilisé une représentation réelle des chromosomes codant les gains du PID [114].

Chang-Kuo Chen et al. ont proposé l'utilisation du codage binaire [99]. Les opérateurs génétiques seront choisis selon le mode de représentation. Les méthodes de PSO sont aussi utilisées pour la conception des contrôleurs PID [283].

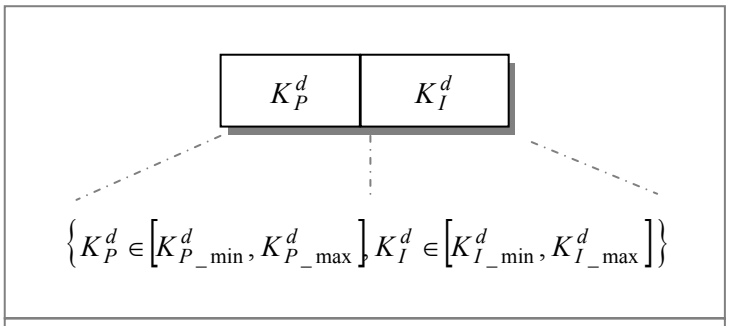


Fig. 3.6 : Structure du chromosome.

A. Soukkou et al. ont proposé un SIF du type TS pour la supervision des paramètres d'un PID [284]. La structure de supervision proposée pour un contrôleur PD est schématisée par la figure 3.7. Deux bases de règles sont construites pour la génération des fonctions non linéaires : $K_{P\_PD} = \tilde{H}_1(x_1, x_2)$ et $K_{D\_PD} = \tilde{H}_2(x_1, x_2)$, respectivement. Les résultats obtenus sur le système du pendule inversé sont encourageants et montrent le potentiel de la démarche.

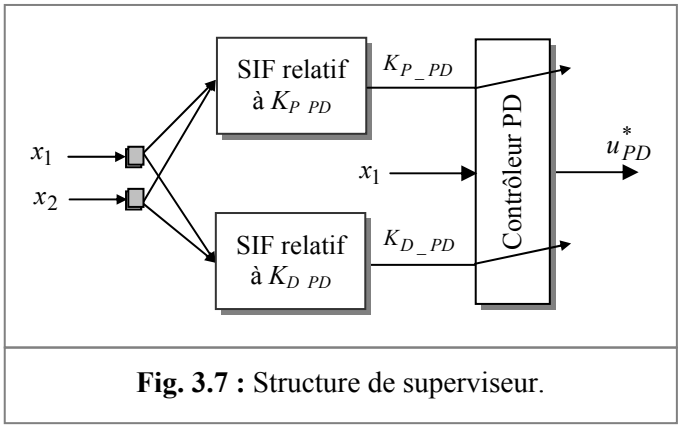


Fig. 3.7 : Structure de superviseur.

Récemment, plusieurs chercheurs ont essayé de combiner les contrôleurs PID conventionnels avec la commande floue. Ils ont proposé différents algorithmes de conception des contrôleurs PID-flous.

Les PID-flous sont des contrôleurs non linéaires dues aux non linéarités que présentent les systèmes flous [75], [91], [267] - [268], [281] & [285] - [292].

Dans la même tendance, l'expression (3.13) donne naissance à une série de contrôleurs non linéaires NL-P, NL-PI, NL-PD et NL-PID. Le tableau 3.4 est une version améliorée de celui du tableau 3.3 où les coefficients de différentes actions sont des fonctions non linéaires. Bien que la plupart des contrôleurs flous proposés ou utilisés sont de type PI ou PD. Dans certains cas, il est nécessaire d'utiliser des contrôleurs de type PID-flous [293].

Tab. 3.4 : Caractéristiques des contrôleurs non linéaires : NL-P, NL-PI, NL-PD & NL-PID.

Type	Variables d'entrées	Loi de commande
NL-P	$\tilde{x}_1(kT) = \left(y^d(kT) - y(kT)\right) \Big/ \tilde{K}_{x_1}$	$u_P^*(kT) = \underbrace{\dfrac{\Theta_1(\tilde{c}_F, \tilde{x}, q, \tilde{w})}{\tilde{K}_{x_1}}}_{\tilde{K}_P^d} \cdot x_1(kT) \Big\vert_{\tilde{c}_{F=1}^1}$
NL-PI	$\begin{cases} \tilde{x}_1(kT) = \left(y^d(kT) - y(kT)\right) \Big/ \tilde{K}_{x_1} \\ \tilde{x}_2(kT) = \left(x_1(kT) - x_1((k-1)\cdot T)\right) \Big/ \tilde{K}_{x_2} \end{cases}$	$\Delta u_{PI}^*(kT) = \tilde{K}_P^d \cdot x_2(kT) + \tilde{K}_I^d \cdot x_1(kT)$ avec $\begin{cases} \tilde{K}_P^d = \dfrac{\Theta_2(\tilde{c}_F, \tilde{x}, q, \tilde{w})}{\tilde{K}_{x_2}} \\ \tilde{K}_I^d = \dfrac{\Theta_1(\tilde{c}_F, \tilde{x}, q, \tilde{w})}{\tilde{K}_{x_1}} \end{cases}$ $u_{PI}^*(kT) = u_{PI}^*((k-1)\cdot T) + \Delta u_{PI}^*(kT)$
NL-PD	$\begin{cases} \tilde{x}_1(kT) = \left(y^d(kT) - y(kT)\right) \Big/ \tilde{K}_{x_1} \\ \tilde{x}_2(kT) = \left(x_1(kT) - x_1((k-1)\cdot T)\right) \Big/ \tilde{K}_{x_2} \end{cases}$	$u_{PD}^*(kT) = \tilde{K}_P^d \cdot x_1(kT) + \tilde{K}_D^d \cdot x_2(kT)$ avec $\begin{cases} \tilde{K}_P^d = \dfrac{\Theta_1(\tilde{c}_F, \tilde{x}, q, \tilde{w})}{\tilde{K}_{x_1}} \\ \tilde{K}_D^d = \dfrac{\Theta_2(\tilde{c}_F, \tilde{x}, q, \tilde{w})}{\tilde{K}_{x_2}} \end{cases}$
NL-PID	$\begin{cases} \tilde{x}_1(kT) = \left(y^d(kT) - y(kT)\right) \Big/ \tilde{K}_{x_1} \\ \tilde{x}_2(kT) = \left(x_1(kT) - x_1((k-1)\cdot T)\right) \Big/ \tilde{K}_{x_2} \\ \tilde{x}_3(kT) = \left(x_2(kT) - x_2((k-1)\cdot T)\right) \Big/ \tilde{K}_{x_3} \end{cases}$	$\Delta u_{PID}^*(kT) = \tilde{K}_P^d \cdot x_2(kT) + \tilde{K}_I^d \cdot x_1(kT) + \tilde{K}_D^d \cdot x_3(kT)$ avec $\begin{cases} \tilde{K}_P^d = \dfrac{\Theta_2(\tilde{c}_F, \tilde{x}, q, \tilde{w})}{\tilde{K}_{x_2}} \\ \tilde{K}_I^d = \dfrac{\Theta_1(\tilde{c}_F, \tilde{x}, q, \tilde{w})}{\tilde{K}_{x_1}} \\ \tilde{K}_D^d = \dfrac{\Theta_3(\tilde{c}_F, \tilde{x}, q, \tilde{w})}{\tilde{K}_{x_3}} \end{cases}$ $u_{PID}^*(kT) = u_{PID}^*((k-1)\cdot T) + \Delta u_{PID}^*(kT)$

Parmi les problèmes rencontrés dans la phase de conception des contrôleurs PID-flous est le nombre important des règles floues. Li & Gatland ont proposé une commande hybride qui consiste à utiliser 2 contrôleurs flous en parallèle du type PI et PD, comme le montre la figure 3.8 [294].

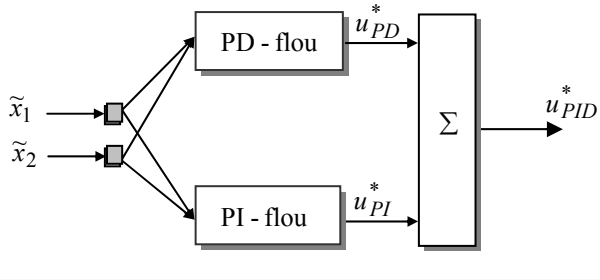


Fig. 3.8 : Contrôleur flou hybride de type PID.

Pour un réseau comportant n entrées et N unités cachées, l'activation des neurones cachés est donnée par une fonction de type gaussienne :

$$\Psi_i = \exp\left(-\sum_{k=1}^{n}\left(\frac{\tilde{x}_k - c_i^k}{\sigma_i^k}\right)^2\right) = \prod_{k=1}^{n}\exp\left(-\left(\frac{\tilde{x}_k - c_i^k}{\sigma_i^k}\right)^2\right) \tag{3.22}$$

i désigne l'indice du neurone k (k^{eme} règle). c_i^k et $\left(\sigma_i^k\right)^2$ sont les centres et les variances des gaussiennes, respectivement. La figure 3.9 présente la forme de cette fonction d'activation pour un neurone possédant une seule entrée et 2 entrées, respectivement.

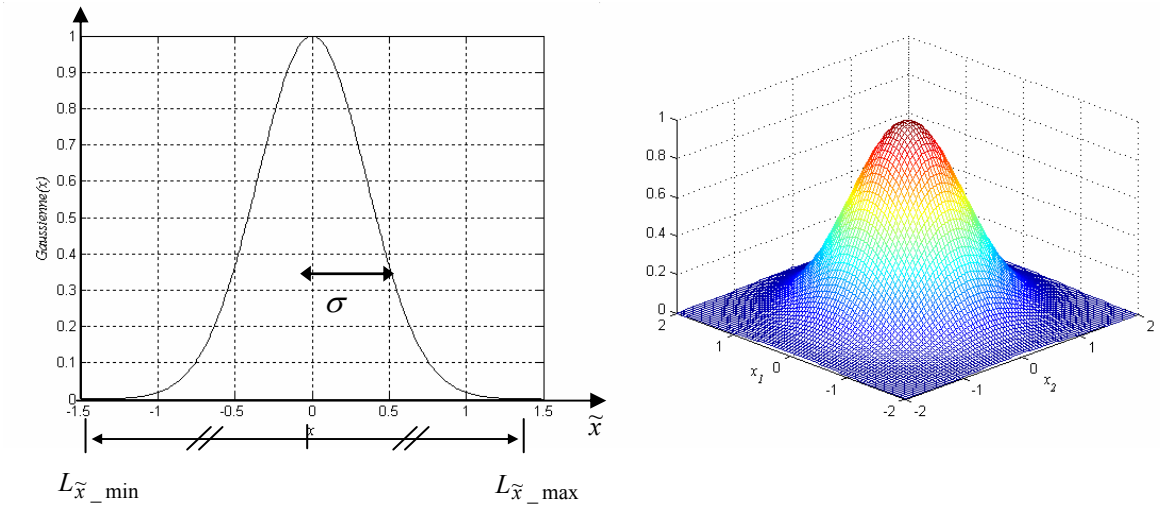


Fig. 3.9 : Sous ensemble flou gaussien à une ou 2 dimensions.

3.4. Processus de conception

3.4.1. Position du problème

Considérons le cas général d'un système dynamique, décrit par les équations différentielles suivantes :

$$\begin{aligned} \dot{x}(t) &= g(x(t), u(t), d_{ist}(t)) \\ y(t) &= C \cdot x(t) \\ \dot{d}_{ist}(t) &= s(d_{ist}(t)) \\ x(o) &= x_0 \qquad\qquad t \in [0, t_f] \end{aligned} \tag{3.23}$$

$x \in \Re^n, u \in \Re^m$ et $y \in \Re^q$ sont les vecteurs d'état, de commande et de sortie, respectivement. $d_{ist}(t) \in R^q$ est le vecteur des signaux additifs (bruits externes). x_0 est le vecteur d'état initial. $g : \Re^n \times \Re^m \to \Re^n$ est une relation non linéaire reliant les variables d'état du système à commander, les signaux de commande générées et les signaux de perturbations, respectivement. $C \in \Re^{q \times n}$ est la matrice de sortie, généralement, constante.

Les signaux de perturbations sont, couramment, de types stochastiques ou sinusoïdaux [295]. Une version des perturbations est générée par des oscillateurs chaotiques $s(\cdot)$ est aussi utilisée dans le domaine de contrôle des procédés [296].

Les signaux de perturbations sont introduits afin de créer un milieu de simulation proche de celui dans la pratique '*pratique virtuelle*'. Généralement, 2 modes de perturbations sont utilisés : Perturbations sur les paramètres fonctionnels du procédé ou sur l'environnement de fonctionnement. Dans ces conditions, la robustesse des systèmes de contrôle peut être vue comme un objectif prémordial. La forme, couramment, rencontrée dans la littérature et utilisée dans la modélisation et la commande des procédés est donnée par :

$$\begin{aligned} \dot{x}_i &= x_{i+1} \qquad\qquad i = 1, ..., n-1 \\ \dot{x}_n &= f(x) + g(x) \cdot u + d_{ist} \\ y &= x_1 \end{aligned} \tag{3.24}$$

ou encore

$$\begin{aligned} y^{(n)} &= f(x) + g(x) \cdot u + d_{ist} \\ &= f\left(y, \dot{y}, ..., y^{(n-1)}\right) + g\left(y, \dot{y}, ..., y^{(n-1)}\right) \cdot u + d_{ist} \end{aligned} \tag{3.25}$$

$f(x)$ et $g(x)$ sont des fonctions (resp. vecteur de fonctions) non linéaires. $g(x)$ représente le gain de commande. L'objectif de la commande consiste à synthétiser une loi de commande u assurant la bornitude de tous les signaux du système asservis et permettant la poursuite d'une trajectoire de référence y^d par la sortie du système.

Les incertitudes sont dues à la variation des paramètres en fonction du mode opératoire, aux conditions et à l'environnement de fonctionnement, ...etc. La robustesse de la commande est dès lors impérative, de sorte que les propriétés du processus commandé puissent être garanties en dépit des différentes sources d'incertitudes.

La capacité du système à rejeter des perturbations, à atténuer leur effet sur les variables commandées ou l'adaptation avec le changement des conditions de fonctionnement est un facteur significatif dans la synthèse des lois de commande simples, efficaces et robustes. L'objectif de poursuite est modélisé par :

$$\lim_{t \to \infty} \left| y(t) - y^d(t) \right| \leq \varepsilon \tag{3.26}$$

où ε est une constante, convenablement, choisie. Afin de pouvoir résoudre le problème de poursuite de trajectoire, le système non linéaire (3.25) doit assurer la condition de commandabilité de sortie, c'est-à-dire :

$$\frac{\partial y^{(n)}}{\partial u} = g(x) \neq 0 \tag{3.27}$$

tout au long de la trajectoire de référence y^d. Le diagramme de la commande en BF est schématisé par la figure 3.10. Il consiste en 4 principaux blocs :

- Bloc d'optimisation caractérisé par un algorithme hybride.
- Bloc structurel représenté par le réseau contrôleur proposé.
- Bloc décisionnel caractérisé par le critère de performances désirées.
- Bloc du système à commander.

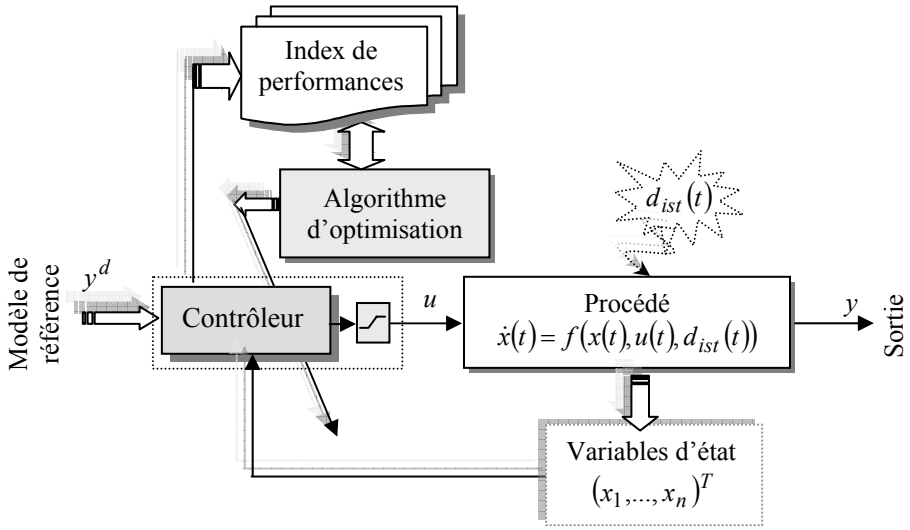


Fig. 3.10 : Structure d'optimisation et de contrôle.

L'interaction mutuelle entre les différents blocs de la structure de la figure 3.10 est illustrée par la procédure suivante :

E1. Génération d'une population initiale des chromosomes caractérisant les paramètres du réseau contrôleur.

E2. Projection de chaque chromosome sur la structure RBF-flou proposée.

E3. Pour tous les chromosomes et tous $(x^T, y^d) \in \Omega$:

- √ Evaluer la fonction objective.
- √ Classifier les chromosomes selon leur aptitude.
- √ Construction d'une nouvelle population par application des opérateurs génétiques adaptés au codage des chromosomes.

Les étapes E2 et E3 sont répétées jusqu'à ce qu'un nombre maximum de générations soit effectué. Après le processus d'évolution, la génération finale de l'algorithme se compose des individus bien adaptés et qui fournissent des solutions 'optimales' ou proches. Le reste de l'algorithme hybride est d'appliquer la partie d'apprentissage plus fin des paramètres de la structure RBF-floue obtenue à la dernière génération de l'AG. L'objectif de cette dernière phase est d'améliorer la précision de la boucle de commande.

3.4.2. Méthode d'optimisation

La majorité des méthodes d'apprentissage des RBF-flous rencontrées dans la littérature ne concerne que la modification d'une partie, seulement, des paramètres du système. La rétropropagation du gradient et la méthode des MCR sont les approches, couramment, utilisées [297] - [298]. L'apprentissage structurel qui consiste à essayer de construire une base de règles floues est aussi abordé. Le groupage flou, la méthode constructive ou destructive sont les approches les plus utilisées. L'utilisation des algorithmes évolutionnaires a, également, été proposée pour permettre un réglage de l'ensemble des paramètres du réseau RBF-flou [101]. L'approche hybride 'multi étage' permet de combiner les méthodes d'optimisation de différents ordres est aussi utilisée pour la conception des RBF-flous. Il est important de noter que le choix d'une méthode d'optimisation doit prendre en considération les points significatifs suivants :

- La convergence de l'algorithme.
- La robustesse numérique de l'algorithme.

Dans ce travail, un algorithme hybride est utilisé pour une optimisation structurelle et paramétrique du contrôleur proposé. Deux étapes d'apprentissage sont présentées (Fig. 3.11) :

- Une optimisation globale de l'ensemble structure & paramètres du réseau contrôleur proposé. L'algorithme utilisé est un AG à objectifs multiples '*multiobjectif*' : AGM. Dans cette phase, une procédure de conception comprenant la réduction de base des règles est présentée. Les règles floues sont modélisées avec des poids binaires. L'AGM est utilisé pour trouver une topologie optimale avec des paramètres adéquats.
- Après l'identification initiale de la topologie et les paramètres du réseau contrôleur, la deuxième étape de l'algorithme d'apprentissage est l'utilisation de la technique de la descente du gradient pour un réglage plus fin des paramètres des conséquences des règles. L'objectif de cette phase est d'améliorer la précision de la réponse du système en BF.

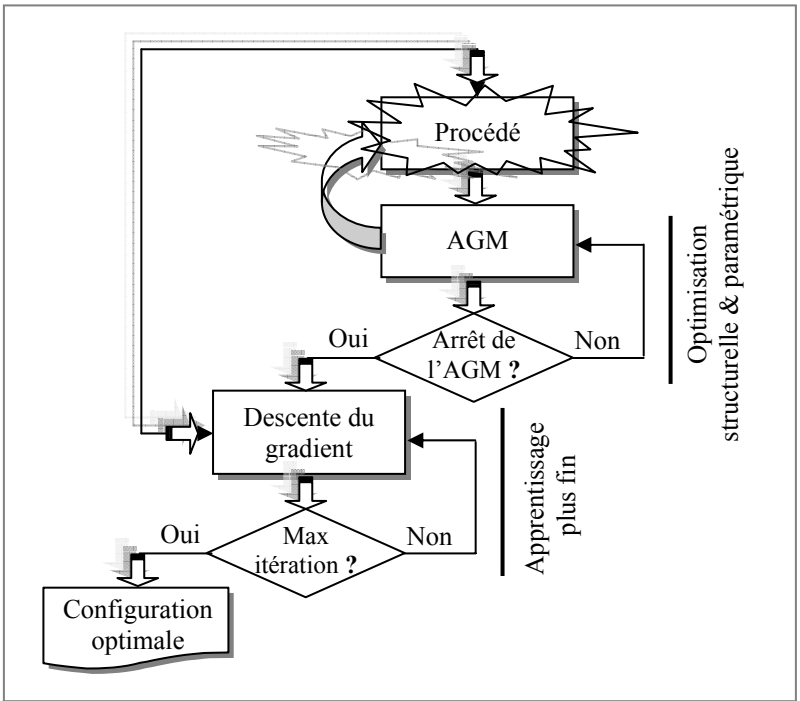


Fig. 3.11 : Algorithme hybride d'apprentissage.

L'inclusion des contraintes de conception dans le processus d'optimisation est présentée dans cette partie de la thèse. Des contraintes sur les limites du vecteur des paramètres à identifier, des limites sur les grandeurs de commande et des variables d'état du procédé à commander et des équations contraignantes assurant un bon compromis entre les objectifs de conception mentionnés par le cahier des charges. La fonction objective doit être, explicitement ou implicitement, dépendante de l'ensemble des paramètres de conception. D'une manière générale, le processus de l'optimisation multiobjectif peut se dérouler de la façon suivante [299] :

Minimiser $F(\alpha)$

Sous les contraintes : (3.28)

$$\begin{cases} G_j(\alpha) \le 0 & j = 1, ..., m_c \\ \& & \\ \alpha_{\inf(i)} \le \alpha_{(i)} \le \alpha_{\sup(i)} & i = 1, ..., n_b \end{cases}$$

$F(\alpha)$ est la fonction objective. Une fois $F(\alpha)$ réduite au minimum, on aura les performances optimales voulues par la conception. Le vecteur α contient n_b paramètres du système à identifier par le processus itératif d'optimisation. $G_j(\alpha)$ est la j^{eme} contrainte sur les paramètres de conception. Il y a m_c contraintes. Chaque paramètre de conception $\alpha_{(i)}$ est limité par ces bornes inférieure et supérieure $\alpha_{\inf(i)}$ et $\alpha_{\sup(i)}$, respectivement. On parle d'une violation de contraintes si $G_j(\alpha) > 0$. La procédure itérative d'optimisation (resp. de réglage) est :

- Déterminer la base de connaissances floues à optimiser.
- Déterminer les objectifs d'optimisation.
- Équations des contraintes.
- Limites pour les éléments du vecteur de conception. Généralement, les fonctions d'appartenances, les variables d'état du procédé et le signal de sortie (saturation).

3.4.2.1. Optimisation génétique

L'AGM proposé dans cette partie de la thèse possède une configuration réduite (micro AG). Un nombre minimal de chromosomes avec des tailles réduites est utilisé dans la population de l'AGM. Les opérateurs génétiques sont adaptés au mode de représentation des chromosomes. Ces opérateurs permettent à la fois une diversité dans l'évolution des populations et une accélération du processus de convergence de l'algorithme. Les μAG sont une modification importante des AG. L'objectif de ces algorithmes est d'accélérer la recherche tout en réduisant la taille de la population. Le but principal de l'optimisation consiste à :

- Trouver une représentation structurelle/paramétrique adéquate pour le réseau RBF-flou.
- Etablir un jeu de contraintes permettant de préserver un compromis des objectifs de conception tout au long du processus d'optimisation.

Nous avons décomposé la construction de la base de connaissances floues en 2 étapes : Une étape d'optimisation simultanée de la structure des règles et des paramètres correspondants. La deuxième étape est un apprentissage plus fin des paramètres, initialement, optimisés dans la première phase.

a. Caractéristiques fondamentales

L'apprentissage génétique consiste à déterminer le nombre minimal de règles floues, soit le nombre des neurones de la couche cachée du RBF-flou proposé, les paramètres correspondants $\{c_i, \sigma_i\}$, les coefficients de TS1 ainsi que les poids des règles actives.

Afin de construire un algorithme d'optimisation à base des AG, il faut bien préparer la plateforme suivante :

- Définir la configuration initiale (taille de la population, critère d'arrêt, probabilités de croisement et de mutation, ...etc.).
- Mode du codage approprié (binaire, réel, ...etc.).
- Méthode de génération de la population initiale.
- Définir la fonction objective.
- Adapter les opérateurs génétiques au mode de codage choisi.

Les caractéristiques du réseau proposé sont :

- A l'entrée du réseau sont associés des facteurs d'échelles permettant de convertir le domaine physique des variables d'entrées au domaine normalisé.
- Les partitions floues des prémisses sont, uniformément, réparties (distribuées) sur l'univers de discours de chaque variable d'entrée, comme l'indique le schéma de la figure 3.12. Les sous ensembles flous utilisés peuvent être définis par des fonctions gaussiennes symétriques équidistantes avec un chevauchement de 50%. Un deuxième type de sous ensemble flou, souvent, utilisé dans les extrémités de l'univers de discours (lignes en pointillés dans la figure 3.12) peut être obtenu à l'aide de la famille de fonctions suivantes :

$$\mu_i(\tilde{x}) = \begin{cases} 1 & \text{si} \quad \tilde{x} < c_i \\ \exp\left(-\dfrac{(\tilde{x} - c_i)}{\sigma_i^2}\right) & \text{autrement} \end{cases} \tag{3.29}$$

$$\mu_i(\tilde{x}) = \begin{cases} \exp\left(-\dfrac{(\tilde{x} - c_i)}{\sigma_i^2}\right) & \text{si} \quad \tilde{x} < c_i \\ 1 & \text{autrement} \end{cases} \tag{3.30}$$

Si χ_{mf}^k est le nombre des fonctions d'appartenance associées à la variable d'entrée $\tilde{x}_k$, $(k = 1,...,n)$. En se basant sur les travaux de Z. J. Zhou et al. & Mu-Song Chen et al. [215] & [300], les paramètres de la i^{eme} gaussienne $\{c_{mf_k}^i, \sigma_{mf_k}^i\}$ sont calculés par la formule suivante :

$$\begin{aligned} c_{mfk}^1 &= \tilde{x}_{\max}^k \\ c_{mf_k}^i &= c_{mf_k}^{i-1} + \frac{\tilde{x}_{\max}^k - \tilde{x}_{\min}^k}{\chi_{mf}^k - 1} \qquad i > 1 \\ \sigma_{mf_k}^i &= \frac{\tilde{x}_{\max}^k - \tilde{x}_{\min}^k}{2 \cdot (\chi_{mf}^k - 1)} \end{aligned} \tag{3.31}$$

$\tilde{x}^k_{\max}$ et $\tilde{x}^k_{\min}$ sont les bornes supérieure et inférieure de l'univers de discours associé à la k^{eme} variable d'entrée, respectivement. Dans l'exemple de la figure 3.12, la k^{eme} variable d'entrée est échantillonnée en 5 valeurs linguistiques modélisées par des fonctions d'appartenance gaussiennes symétriques : **NB**, **NM**, **ZE**, **PM** et **PB** avec leurs numéros **1**, **2**, **3**, **4** et **5**, respectivement. Cette méthode permet de réduire le nombre des paramètres à optimiser.

Il suffit de définir la plage de variation de chaque variable d'entrée et le nombre des sous ensembles flous associés. Le reste est d'appliquer la formule itérative (3.31).

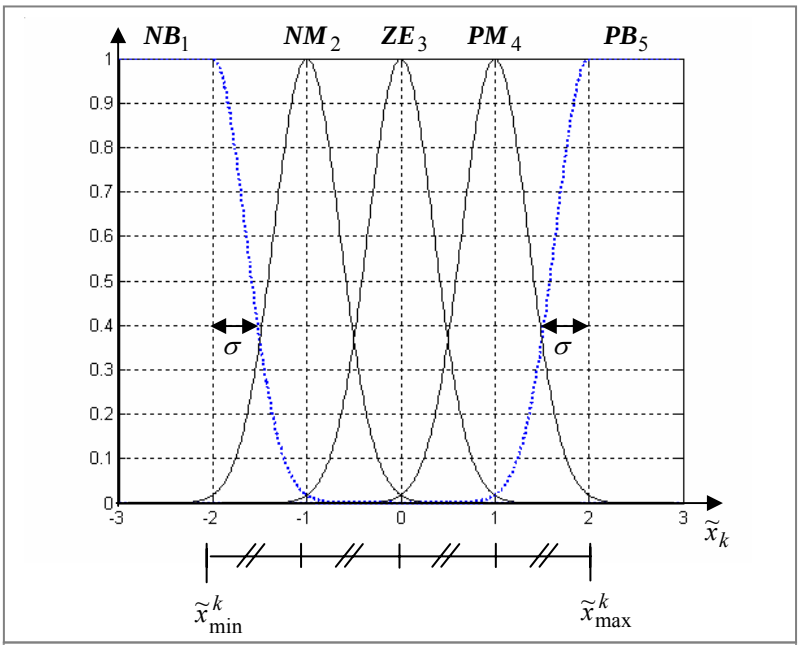


Fig. 3.12 : Partitions floues, uniformément, distribuées.

- A chaque règle est associé un facteur d'incertitude. Selon sa valeur, la règle correspondante doit être active ou inactive.
- Chaque règle active est affectée par un poids caractérisant sa contribution (importance) à la stabilité du système en BF par rapport à l'ensemble de la base des règles réduite.
- Les liens entre les différentes couches du réseau de la figure 3.4 (resp. 3.5) sont donnés par le tableau 3.5.

La matrice des inférences et la structure chromosomiale correspondante pour un contrôleur à 2 entrées $\{\tilde{x}_1, \tilde{x}_2\}$, une sortie u et 25 règles d'inférence est représentée par le schéma de la figure 3.13. Chacune des entrées est échantillonnée en 5 valeurs linguistiques modélisées par des fonctions d'appartenance gaussiennes symétriques équidistantes avec un chevauchement de 50 %. Dans cette représentation et comme exemple, la règle 9 s'écrit de la façon suivante :

$$R_9 : \textbf{SI } (\tilde{x}_1 \text{ est } \textit{NM}) \textbf{ ET } (\tilde{x}_2 \text{ est } \textit{PM}) \textbf{ ALORS } u \textit{ est } F^9(\tilde{x}_1, \tilde{x}_2)$$

Tab. 3.5 : Liens entre les couches du réseau contrôleur.

Lien entre couche	Valeur
Entrées & couche 1	$\left(\widetilde{K}_{x_1}^{-1}, ..., \widetilde{K}_{x_n}^{-1}\right)$
Couche 1 & 2	$c_j = \left[c_j^1 \quad c_j^2 \quad ... \quad c_j^n\right]^T \quad j = 1, ..., N$
Couche 2 & 3	$\left(\underbrace{1, ..., 1}_{N \text{ fois}}\right)$
Couche 3 & 4	$\left(\underbrace{1, ..., 1}_{N \text{ fois}}\right)$
Couche 4 & 5	$\left(\widetilde{w}_1, ..., \widetilde{w}_N\right)$
Couche 5 & sortie	1

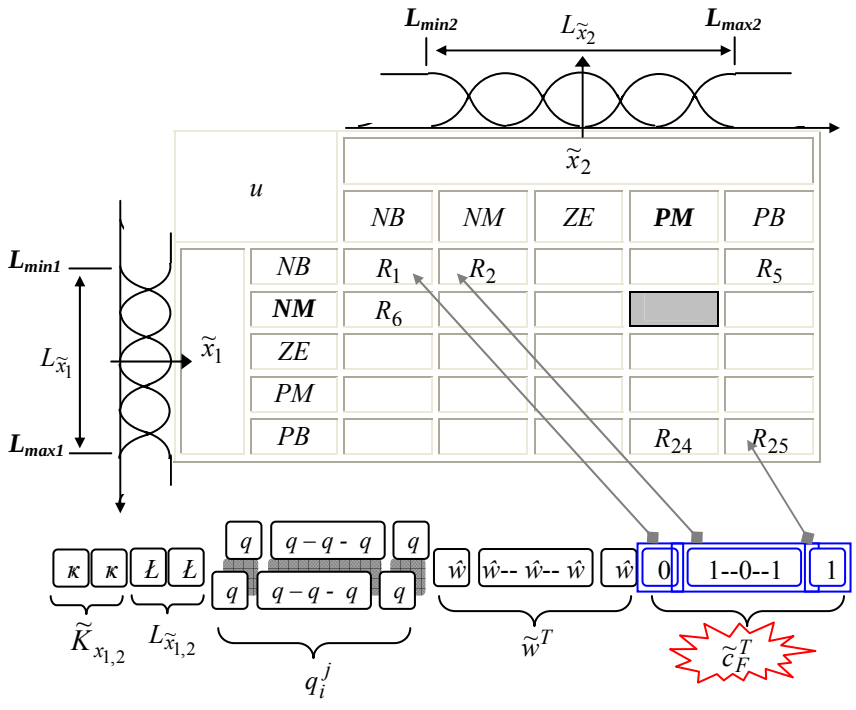


Fig. 3.13 : Correspondance entre la matrice d'inférence et le chromosome.

Les variations des AG peuvent être distingués par le type de codification utilisé pour les chromosomes et les opérateurs génétiques utilisés. Le codage multiparamétré mixte binaire - réel est utilisé comme outil de représentation chromosomiale dans ce travail. La signification ainsi que le mode du codage approprié est illustré par le tableau 3.6.

Le but principal de l'application des AG pour la conception d'un SIF est non seulement de profiter des avantages des AG, mais de développer, également, une approche systématique de conception des SIF. Les principales caractéristiques de l'AGM (Fig. 3.15) adapté à la forme de représentation précédente sont données par le tableau 3.7.

Tab. 3.6 : Modes du codage des chromosomes.

Paramètre	Mode de codage	Signification
$\widetilde{K}_{x_{1,2}}$	Réel	Facteurs d'échelles des variables d'entrées
$L_{\widetilde{x}_{1,2}}$	Réel	Largeurs des univers de discours des variables d'entrées $\widetilde{x}_1$ et $\widetilde{x}_2$, respectivement.
$q_i^j \Big\vert_{i=1,\ldots,n}^{j=1,\ldots,N}$	Réel	Paramètres des conséquences (TS1)
$\widetilde{w}^T = \left[\widetilde{w}_1, \ldots, \widetilde{w}_N\right]$	Réel	Poids des règles
$\widetilde{c}_F^T = \left[\widetilde{c}_F^1, \ldots, \widetilde{c}_F^N\right]$	Binaire	Facteurs d'incertitudes
n		Nombre des variables d'entrée
N		Nombre des règles d'inférence

Tab. 3.7 : Caractéristiques de l'AGM.

Caractéristiques	valeurs
Taille de la population	$Max\_Pop \in [5,10]$
Nombre de génération	$Max\_gen \in [10,500]$
Représentation	Mixte binaire-réelle
Initialisation	Aléatoire
Facteurs d'échelle	$\widetilde{K}_{x_1} \in \left[\widetilde{K}_{x_1\_\min}, \widetilde{K}_{x_1\_\max}\right]$ & $\widetilde{K}_{x_2} \in \left[\widetilde{K}_{x_2\_\min}, \widetilde{K}_{x_2\_\max}\right]$
Univers de discours	$L_{\widetilde{x}_1} \in \left\lfloor L_{\widetilde{x}_1\_\min}, L_{\widetilde{x}_1\_\max}\right\rfloor$ & $L_{\widetilde{x}_2} \in \left\lfloor L_{\widetilde{x}_2\_\min}, L_{\widetilde{x}_2\_\max}\right\rfloor$
Poids des règles	Voir critère de stabilité
Paramètres de TS1	$q_i^j \Big\vert_{i=1,\ldots,n}^{j=1,\ldots,N} \in [-1,+1]$
Processus de sélection	Roue de loterie
Opérateur de croisement	Multipoints (2 points) (Fig. 3.14)
Probabilité de croisement	$P_c = 0.85$
Opérateur de mutation	Uniforme dans la partie binaire & non uniforme autrement
Probabilité de mutation	$P_m \cong 1/\left(Max\_Pop \cdot \sqrt{L\_chrom}\right)$ où L_chrom est la taille du chromosome [189].
Mode de remplacement	Elitisme

L'opérateur de mutation permet d'avoir une large diversité des solutions dans la population afin d'améliorer la chance pour que l'AGM s'échappe des optimums locaux. Le processus de mutation dépend du mode de codage appliqué. Dans la partie binaire du chromosome, l'opération consiste, tout simplement, en une inversion de gènes 'bit'.

Si la mutation a lieu dans la partie du chromosome codée en réel, nous avons appliqué la mutation non linéaire donnée par l'expression (1.72) avec $\tau = 3$. Enfin et pour mettre une limite à la divergence génétique, la stratégie d'élitisme a été introduite.

L'élitisme est un mécanisme introduit pour conserver les bonnes solutions lors du passage de la génération courante à la prochaine génération. Conserver ces solutions pour

les générations futures permet d'améliorer les performances des algorithmes sur certains problèmes.

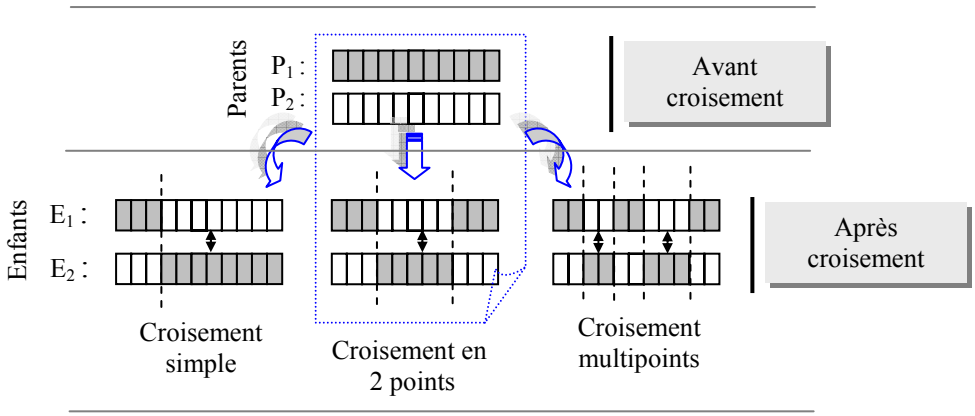


Fig. 3.14 : Croisement.

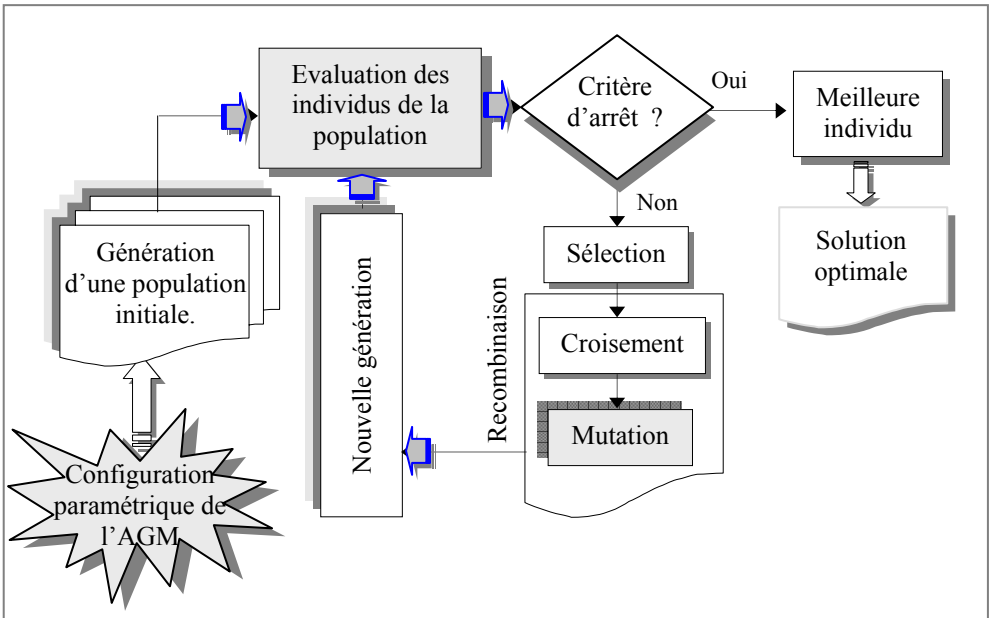


Fig. 3.15 : Processus d'exécution de l'AGM.

b. Critère de performances

La notion de performances désirées 'coût' est centrale dans les procédures d'optimisation et d'apprentissage. L'obtention d'un contrôle performant requiert une bonne formulation de cette notion. L'efficacité du contrôleur synthétisé dépend de la fonction coût choisie ainsi que de l'algorithme d'adaptation. Généralement, lors de la synthèse d'une loi de commande, les performances recherchées s'articulent souvent autour de 3 types de spécifications, à savoir :

- La simplicité d'implantation et de mise en œuvre.
- La précision et la rapidité.
- La robustesse par rapport à la stabilité.

Dans cette étude, la fonction à optimiser tient compte des performances du système à évaluer, de la complexité de la structure de contrôle et de l'effort énergétique à appliquer. Mathématiquement, la fonction à optimiser sera donnée par :

$$J = \underbrace{\Phi_1\left(J_1\left(S_{SBF}\right)\right)}_{\text{Critère 1}} \widetilde{\oplus} \underbrace{\Phi_2\left(J_2\left(S_C\right)\right)}_{\text{Critère 2}} \widetilde{\oplus} \underbrace{\Phi_3\left(J_3\left(S_p\right)\right)}_{\text{Critère 3}} \tag{3.32}$$

Le symbole $\widetilde{\oplus}$ désigne une composition arithmétique implémentée par la somme algébrique. $J_1\left(S_{SBF}\right)$ mesure la performance de l'ensemble : Système de commande-système à commander (Système en BF SBF). $J_2\left(S_C\right)$ représente la structure du système de commande S_C spécifiée par le nombre des règles floues. $J_3\left(S_p\right)$ est l'effort de contrôle à appliquer sur le procédé S_p. Les fonctions Φ_i, $\left(i = 1, 2, 3\right)$ sont choisies en fonction des objectifs visés. En plus les objectifs visés, la prise en considération des contraintes présente une alternative de l'optimisation des fonctions (Fig. 3.16).

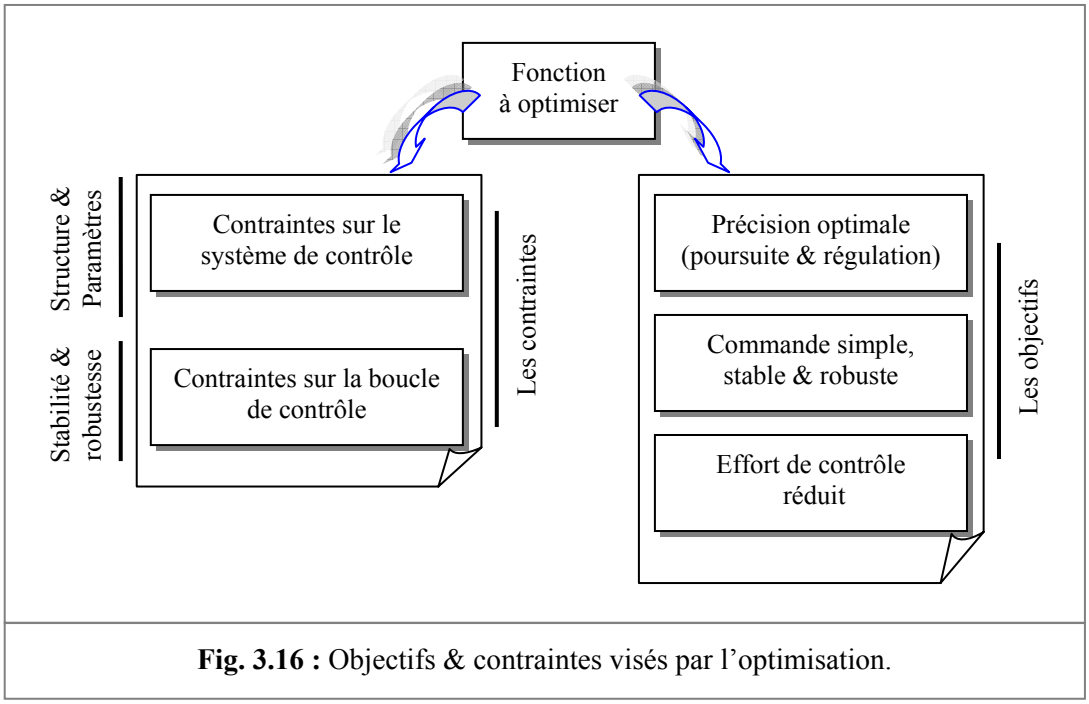


Fig. 3.16 : Objectifs & contraintes visés par l'optimisation.

Le tableau 3.8 résume la stratégie d'optimisation adaptée aux objectifs visés dans ce travail. Le lien entre l'environnement (procédé) et l'AGM est représenté par le critère de performances à optimiser (voir Annexe A).

Tab. 3.8 : Critères élémentaires d'optimisation.

Fonction	Forme	Ordre de priorité
$\Phi_1\left(J_1\left(S_{SBF}\right)\right)$	$\frac{\beta_1}{\max\_k} \cdot \sum_{k=0}^{\max\_k - 1} \left\|e(k)\right\|$	1
$\Phi_2\left(J_2\left(S_C\right)\right)$	$\frac{\beta_2}{\max\_règle} \cdot \sum_{i=1}^{\max\_règle} \left\{\widetilde{c}_F^{\,i}\right\}$	2
$\Phi_3\left(J_3\left(S_p\right)\right)$	$\frac{\beta_3}{\max\_k} \cdot \sum_{k=0}^{\max\_k - 1} \left\|u(k)\right\|$	3

$e(k) = y(k) - y^d(k)$ est l'erreur de poursuite à l'instant d'échantillonnage k. max_k et max_*règle* sont le nombre des échantillons des données d'apprentissage et la taille maximale de la base des règles, respectivement. max_*règle* dépend du partitionnement de l'espace des variables d'entrées.

β_i, $(i = 1, 2, 3)$ sont des facteurs dynamiques de pondérations caractérisant la précision, l'énergie et la complexité du contrôleur. Ils représentent l'importance relative de chaque objectif. Supposant que :

$$\begin{cases} \beta_1, \beta_2, \beta_3 \geq 0 \\ \quad \& \\ \beta_1 + \beta_2 + \beta_3 \leq 1 \end{cases} \tag{3.33}$$

Un cas particulier est le cas où $\beta_1 + \beta_2 + \beta_3 = 1$. On aura une combinaison convexe des facteurs conceptuels dans cette situation.

$$\beta_1 > \beta_2 > \beta_3 \tag{3.34}$$

Dans le cas où $\beta_1 + \beta_2 + \beta_3 \cong 1$, un compromis entre meilleure précision, un nombre optimal de règles et d'une consommation réduite de l'énergie de commande (plus de sécurité). En plus de l'efficacité, de la simplicité et de la sécurité de fonctionnement du procédé, la stabilité de la boucle de contrôle et la robustesse du contrôleur vis-à-vis des perturbations sont aussi des facteurs importants dans la modélisation et le contrôle des systèmes. L'ordre de priorité est mesuré par la capacité du système de contrôle de réaliser 'efficacement' les objectifs visés par le concepteur. Dans notre cas, les objectifs sont classés en fonction du degré d'importance des facteurs conceptuels.

- La précision & la stabilité.
- La réduction de complexité de la loi de commande.
- La réduction de l'effort de contrôle (énergie) à appliquer sur le procédé.

En conclusion, on peut formuler le critère des performances désirées d'une manière mathématique selon la relation :

Minimiser J

$$\text{avec} \quad J = \frac{\beta_1}{\max\_k} \cdot \sum_{k=0}^{\max\_k - 1} \left| e(k) \right| + \frac{\beta_2}{\max\_règle} \cdot \sum_{i=1}^{\max\_règle} \left\{ \tilde{c}_F^i \right\} + \frac{\overbrace{\left(1 - (\beta_1 + \beta_2)\right)}^{\beta_3}}{\max\_k} \cdot \sum_{k=0}^{\max\_k - 1} \left| u(k) \right|$$

Sous les contraintes :

$$\begin{cases} \beta_1 + \beta_2 + \beta_3 = 1 & \text{Contrainte 1} \\ \begin{cases} \sum_{i=1}^{\max\_règle} \left\{ \tilde{c}_F^i \right\} \geq 1 \\ \sum_{l=1}^{N} \rho_i(\tilde{x}) \neq 0 \end{cases} & \text{Contrainte 2 (3.17)} \\ \sum_{i=1}^{n} \left| \Theta_i(\cdot) \right| \cdot \xi_i \leq C_u & \text{Contrainte 3 (3.45)} \end{cases} \tag{3.35}$$

En plus, les critères de bornitude des variables d'état, de contrôle et les paramètres conceptuels du contrôleur sont introduits tout en assurant la stabilité de la boucle de contrôle, l'évitement des régions de saturation de la loi de commande est une formulation adéquate de la structure du contrôle.

Afin de tenir compte des contraintes sur les variables manipulées ϑ_i, la fonction suivante de saturation a été définie comme suit :

$$\vartheta_i = \begin{cases} \vartheta_{i,\max} & \text{si} \quad \vartheta_{i,\max} < \vartheta_i^* \\ \vartheta_i^* & \text{si} \quad \vartheta_{i,\min} \le \vartheta_i^* \le \vartheta_{i,\max} \\ \vartheta_{i,\min} & \text{si} \quad \vartheta_i^* < \vartheta_{i,\min} \end{cases} \tag{3.36}$$

Dans la situation (3.35), l'objectif est exprimé sous forme de minimisation de la fonction coût $J(\cdot)$. Pour passer du problème de minimisation au problème de maximisation bien adapté à la nature de l'AG, nous avons adopté à utiliser la fonction objective donnée par l'expression (1.62) où C^{te1} et C^{te2} sont égales à l'unité.

Les contraintes 1, 2 et 3 dans l'expression (3.35) sont examinées d'une manière séquentielle de sorte qu'on aura aucune violation d'une de ces contraintes durant l'évolution de l'algorithme d'optimisation.

- Le choix des β_i, $(i = 1, 2, 3)$ n'est pas une tâche évidente. La connaissance des dépendances entre les objectifs simplifie le choix final. Une solution est d'utiliser une combinaison linéaire des objectifs et de faire varier les β_i de manière à constater l'influence de tel ou tel objectif sur le résultat. Ils sont choisis empiriquement, avec bien sûr le respect des ordres des objectifs d'optimisation mentionnés dans les paragraphes précédents. Les valeurs des facteurs β_i, $(i = 1, 2, 3)$ adaptées à la stratégie de commande ainsi développée seront données par le tableau 3.9. Cette proposition est une sorte d'amélioration de fonctionnement des AG : Le mécanisme de changement d'échelle de la fonction d'évaluation. Ce mécanisme permet une compétition entre les critères et d'assurer un niveau de compétition entre les objectifs d'un coté et une diversité des solutions dans la population de l'autre côté.

Tab. 3.9 : Evolution des facteurs de pondérations.

Paramètre	Valeur	Condition
β_1	$0.50\,(50\%)$	$gen \le (2 \cdot Max\_gen)/3$
β_2	$0.35\,(35\%)$	
β_3	$0.15\,(15\%)$	
β_1	$0.34\,(34\%)$	$gen > (2 \cdot Max\_gen)/3$
β_2	$0.33\,(33\%)$	
β_3	$0.33\,(33\%)$	

- La contrainte 2 permet au contrôleur d'être opérationnel tout au long de l'espace entier des données d'entrées. Elles assurent le fonctionnement nominal de la boucle de contrôle. Dans le cas des fonctions d'appartenance triangulaire, la

contrainte $\sum_{l=1}^{N} \rho_i(\tilde{x}) \neq 0$ doit être vérifiée avec un bon choix des règles qui couvrent l'espace des données d'entrées. Dans le cas des fonctions gaussiennes, comme dans notre cas, les degrés d'appartenance ne seront pas zéro en un point quelconque, et le système (3.18) est bien défini, ainsi aucune modification est nécessaire [301]. Durant le processus d'optimisation, le problème de violation de la contrainte 2 survient de temps en temps. La solution proposée est d'agir dans la partie des facteurs d'incertitudes du chromosome correspondant (Fig. 3.17). Une méthode proposée par A. Soukkou et al. est la destruction du chromosome responsable de la violation des contraintes et de générer un remplaçant qui vérifie les conditions imposées par le concepteur [270]. Les résultats obtenus sont encourageants. Le temps de simulation est l'inconvénient majeur de cette méthode. Pour réduire le temps d'exécution de l'AGM et d'éviter des boucles infinies de l'exécution, nous avons introduit la notion de *mutation forcée* dans la partie binaire des chromosomes. A chaque fois qu'il y a une violation des contraintes par un chromosome, un appel à la procédure de mutation forcée aura lieu. L'ancien chromosome sera remplacé par le nouveau né qui vérifie toutes les contraintes. Ce processus se répète pour tous les chromosomes de la population en cours. L'organigramme de la figure 3.17 résume la stratégie adoptée pour éviter la violation des contraintes 2 & 3. La contrainte 2 permet au contrôleur d'être opérationnel tout au long de l'espace entier des données d'entrées. La contrainte 3 sera examinée par la suite du théorème ci-dessous.

- Les valeurs des poids $\tilde{w}_i$ ont une influence directe sur la stabilité et la robustesse du système en BF. L'analyse de la stabilité par l'approche *Entrée Bornée – Sortie Bornée* 'BIBO' est adoptée dans ce travail. Cette approche permet d'affirmer la stabilité d'un système si la sortie de celui-ci, en réponse à une entrée bornée, est elle-même bornée. L'analyse de la stabilité d'une boucle de contrôle en présence d'un contrôleur flou est un sujet difficile puisque les modèles considérés sont de nature non linéaires [302]. Dans le contexte de la stabilité générale, 2 approches peuvent être utilisées :
  - La théorie de la stabilité entrée/sortie traite des relations entre les sorties du système et ses entrées.
  - La théorie de Lyapunov fait référence aux propriétés de la représentation interne du système en terme de variable d'état et d'effet des perturbations résultantes du changement des conditions initiales.

  D'un point de vue mathématique, l'approche entrée/sortie est basée sur les opérateurs et l'analyse fonctionnelle. Par contre, la méthode de Lyapunov utilise une formulation par les équations différentielles.

Théorème :

Le système est dit BIBO (entrée bornée - sortie bornée) stable si la sortie reste d'énergie finie tant que l'entrée est d'énergie finie. La stabilité entrée-sortie signifie qu'un signal e de norme finie $\|e\|$ en entrée du système engendre un signal s de norme finie $\|s\|$ en sortie et qu'il existe un scalaire γ tel que [303] *:*

$$\|s\| < \gamma \|e\| \tag{3.37}$$

La stabilité BIBO est dite faible s'il existe 2 scalaires β et γ tels que :

$$\|s\| < \gamma\|e\| + \beta \tag{3.38}$$

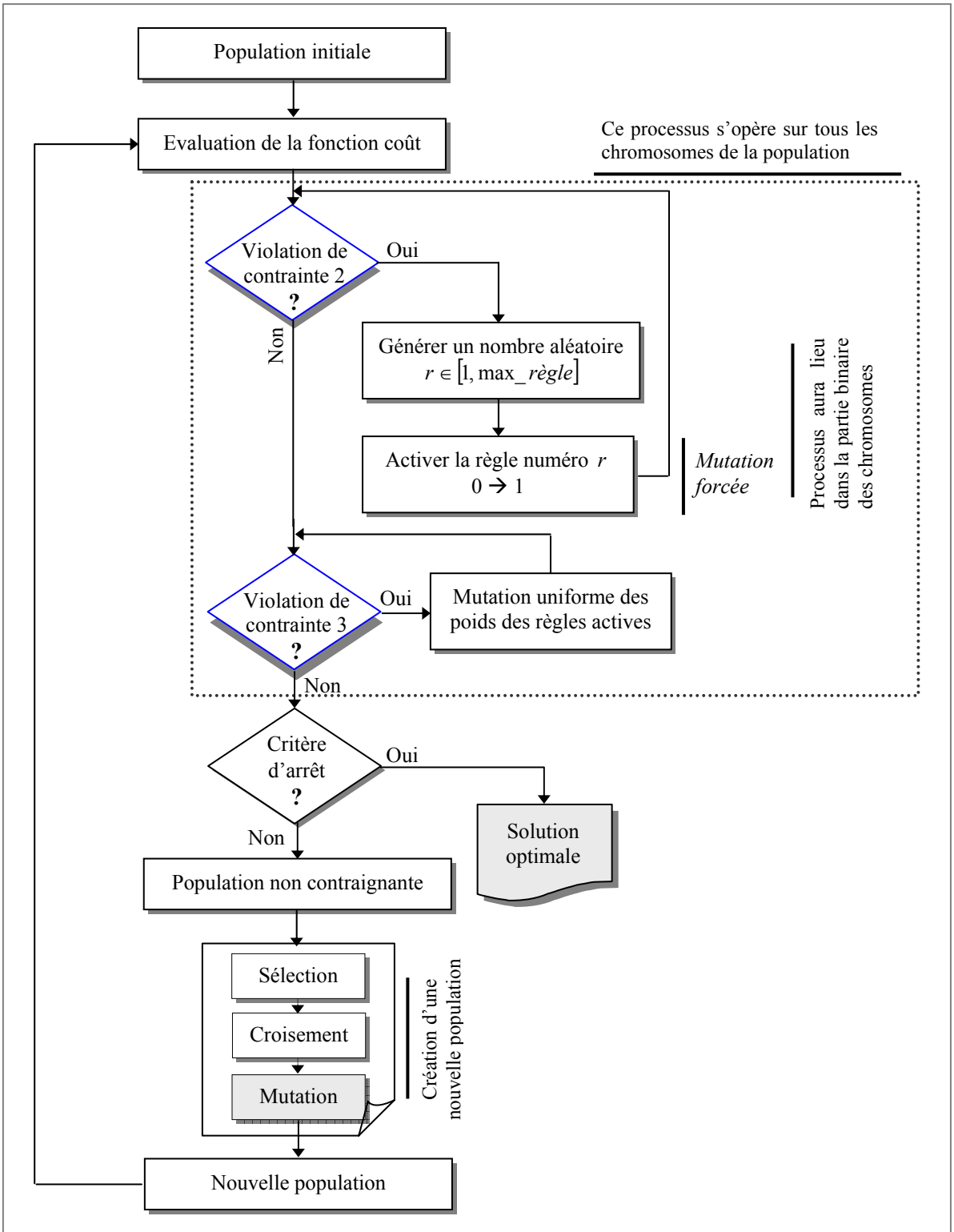


Fig. 3.17 : Processus de traitement des contraintes par l'AGM.

Pratiquement, un processus industriel fonctionne dans un voisinage de ses points de fonctionnement. Sa dynamique est bornée auteur de ces points de fonctionnement à cause des contraintes de bornitude sur le processus. Habituellement, les contraintes sont connues

à priori [304]. Dans n'importe quelle application pratique, une analyse de la stabilité du système est nécessaire pour assurer son bon fonctionnement [302].

Prenons le cas d'un système à une entrée de commande u et n variables d'état $x=[x_1,...,x_n]^T$ décrivant son comportement. Les conditions aux limites sont données par les inégalités suivantes :

$$|x| \le C_x < \infty \tag{3.39}$$

&

$$|u| \le C_u < \infty \quad \& \quad C_u = \max\left(|u_{\min}|, |u_{\max}|\right) \tag{3.40}$$

Les vecteurs C_x et C_u sont des scalaires prédéfinis dépendant du fonctionnement du procédé à commander. De l'expression (3.18), la sortie du contrôleur flou à n entrées sera réécrite sous la forme :

$$u = \underbrace{\frac{\sum_{l=1}^{N} \rho_l(\tilde{x}) \cdot \left\{c_F^l \cdot q_1^l\right\} \cdot \tilde{w}_l}{\sum_{l=1}^{N} \rho_i(\tilde{x})}}_{\Theta_1(\cdot)} \cdot \tilde{x}_1 + ... + \underbrace{\frac{\sum_{l=1}^{N} \rho_l(\tilde{x}) \cdot \left\{c_F^l \cdot q_n^l\right\} \cdot \tilde{w}_l}{\sum_{l=1}^{N} \rho_i(\tilde{x})}}_{\Theta_n(\cdot)} \cdot \tilde{x}_n \tag{3.41}$$

Si $x^d = [x_1^d,...,x_n^d]^T$ est le vecteur des états désirés, l'erreur de poursuite sera donnée par la limite :

$$|x^d - x| \le \max|x^d - C_x| \quad \& \; C_x = \max\left(|x_{\min}|, |x_{\max}|\right) \tag{3.42}$$

Après normalisation du vecteur d'entrée, la condition de bornitude de chaque entrée sera reformulée par la relation :

$$|\tilde{x}_i| \le \underbrace{\frac{|x_i^d - C_{x_i}|}{\tilde{K}_{x_i}}}_{\xi_i} \tag{3.43}$$

De la condition (3.40), l'expression (3.41) peut être remplacée par l'inégalité suivante :

$$|u| \le |\Theta_1(\cdot) \cdot \tilde{x}_1| + ... + |\Theta_n(\cdot) \cdot \tilde{x}_n| \le C_u \tag{3.44}$$

En remplaçant l'inégalité (3.43) dans (3.44), nous obtenons la condition de stabilité du système en BF :

$$\sum_{i=1}^{n} |\Theta_i(\cdot)| \cdot \xi_i \le C_u \tag{3.45}$$

Cette expression nous permet de trouver des poids des règles actives $\tilde{w}_i$ assurant la stabilité en BF. On peut étendre ce théorème au cas discret. Pour aider à guider la recherche génétique des poids $\tilde{w}_i$, nous proposons de choisir $\tilde{w} \in \left]0, \alpha \times \max\left(|u_{\min}|, |u_{\max}|\right)\right]$ avec $\alpha > 0$ comme condition nécessaire, mais pas suffisante. Pour assurer la stabilité globale, il faut que la condition (3.45) soit satisfaite. S'il y a une violation de la contrainte 3 dans (3.35), nous avons appliqué la même procédure qu'avec la contrainte 2. Une succession des changements aléatoires dans la partie des poids dans le chromosome en cours d'évaluation aura lieu par application de l'opérateur de mutation. Les poids des règles actives vont subir à ses changements par application de l'opérateur de mutation uniforme mentionné dans le chapitre 1.

Une fois les conditions de fonctionnement nominales du système en BF sont établies, un réglage plus fin de certains paramètres de la boucle de contrôle est adopté. Cette opération agit directement sur la précision, c'est-à-dire, l'efficacité du système en BF. La procédure de la descente du gradient est suffisante pour accomplir cette tâche.

3.4.2.2. Apprentissage fin

Les paramètres du réseau sont adaptés par application de la méthode de la rétropropagation du gradient de la fonction coût donnée par :

$$E(k) \equiv \frac{1}{2} \sum_{k=1}^{P} \left(y^d(k) - y(k)\right)^2 \tag{3.46}$$

P est le nombre des échantillons de données d'entrées (données d'apprentissage). $y^d(k)$ et $y(k)$ sont les échantillons de la sortie désirée et de la sortie actuelle du processus à l'itération *k*, respectivement. Le vecteur des paramètres à adapter est représenté par :

$$\vartheta_i \equiv \left[q_i^{jT}, \tilde{w}^T \right]^T \tag{3.47}$$

La loi d'adaptation est [188] :

$$\vartheta_i^{k+1} \equiv \vartheta_i^k - \eta_\vartheta \cdot \frac{\partial E(k)}{\partial \vartheta_i} \tag{3.48}$$

Une fois la structure du modèle du contrôleur est construite par l'AGM, un processus itératif d'apprentissage plus fin de certains paramètres est, successivement, appliqué pour améliorer la précision du modèle ainsi obtenu. L'algorithme de la rétropropagation du gradient est appliqué dans ce travail. Le modèle dans ce cas, peut être vu comme réseau fonctionnel avec une topologie minimale. Une 'bonne' initialisation structurelle & paramétrique est établie par l'AGM.

La procédure d'adaptation des paramètres des conséquences ainsi que les poids des règles peut être récapitulé comme suit :

E1. Initialisation des paramètres. Les valeurs initiales des paramètres sont égales aux valeurs obtenues à la dernière génération de l'AGM.

E2. Appliquer le vecteur d'entrée du contrôleur $X = (x_1, ..., x_n)^T$.

E3. Spécifier le vecteur des sorties désirées $Y^d = \left(y_1^d, ..., y_k^d\right)^T$.

E4. Calcul des sorties $Y = (y_1,..., y_k)^T$ du système.

E5. Adaptation des paramètres q_j^i et $\tilde{w}_i$.

E6. Evaluation de la fonction coût donnée par l'expression (3.46).

Répéter les étapes E2 à E6 jusqu'à ce que la valeur de E donnée par l'équation (3.46) soit plus petite qu'une valeur de seuil prédéfinie ou le critère d'arrêt soit satisfaite. Un nombre maximal d'itérations est le critère d'arrêt utilisé couramment dans ce type d'algorithmes. La procédure de la technique de descente du gradient est décrite par l'organigramme de la figure 3.18.

Procédure Descente_gradient (*Max_iteration*)
{
For $(k = 1; k <= Max\_iteration; k++)$
{
For all $(X^T, y_k^d) \in \Omega$ **do**
{
For $(j = 1; j <= n; j++)$
For $(i = 1; i <= N; i++)$

$$q_j^i(k+1) = q_j^i(k) - \eta_{q_j^i}(y_k^d - y_k)\frac{\rho_i}{\sum_{l=1}^{N}\rho_l} \cdot \tilde{w}_i(k) \cdot x_j(k);$$

$$\tilde{w}_i(k+1) = \tilde{w}_i(k) - \eta_{\tilde{w}}(y_k^d - y_k)\frac{\rho_i}{\sum_{l=1}^{N}\rho_l} \sum_{j=1}^{n} q_j^i(k) \cdot x_j(k);$$

}
}
}

Fig. 3.18 : Organigramme de la descente du gradient.

Les coefficients $(\eta_{\tilde{w}}, \eta_{q_i^j}) \in]0,1]$ représentent le taux d'apprentissage des poids des règles et les paramètres des conséquences, respectivement. Les valeurs de $\eta_{\tilde{w}}$ et $\eta_{q_i^j}$ sont choisis, empiriquement, égales à 0.001.

Il est important de souligner la nécessité pour la méthode de synthèse développée d'être peu gourmande en temps de calcul. Les procédures de test des contraintes et la recherche des corrections 'solutions' en cas de violation de ces contraintes représentent l'inconvénient majeur de l'algorithme, en particulier la contrainte 2. Pour réduire la complexité de calcul, il est important de bien choisir les conditions initiales de l'exécution de l'algorithme. Dans ce cas, le savoir faire du concepteur 'l'expert' peut améliorer le temps de convergence de l'algorithme vers l'objectif visé. De plus l'AGM est utilisé avec une forme réduite, c'est-à-dire, un nombre minimal de chromosomes.

La validité de la méthodologie de synthèse proposée, est testée, sur différents systèmes non linéaires : Contrôle de la température d'un bain d'eau, contrôle du pendule inversé et la commande d'un robot manipulateur à 2 degrés de liberté.

3.5. Simulation

Dans le but de tester et évaluer les performances de l'algorithme proposé pour la synthèse d'une loi de commande à objectifs multiples (simple, précise, stable et robuste), nous avons effectué des simulations sur quelques systèmes non linéaires. A chaque type de système non linéaire est appliquée une forme spécifique de la loi de commande développée. Cette différence est due, principalement, à la nature et à la complexité du procédé à commander.

La configuration initiale utilisée dans les simulations est la suivante :

- Pour chaque variable d'entrée sont associés 5 sous ensembles flous, comme mentionnée dans la figure 3.13.
- La taille maximale de la base des règles pour un SIF à 2 entrées et 1 sortie est de 25 règles $(\max\_ régle = 25)$.

L'objectif de commande est de permettre aux états du système de suivre une trajectoire de référence prédéterminée. Ainsi, les erreurs de poursuite doivent être aussi petites que possible (précision) et le système en BF doit être globalement stable et robuste, c'est-à-dire, tous ses paramètres sont, uniformément, bornés et l'effet des perturbations externes est atténué à un niveau prescrit.

3.5.1. Commande de la température d'un bain d'eau (contrôleur NL-P)

Le but de cette section est de commander la température d'un bain d'eau. La structure du procédé est donnée par le schéma de la figure 3.19.

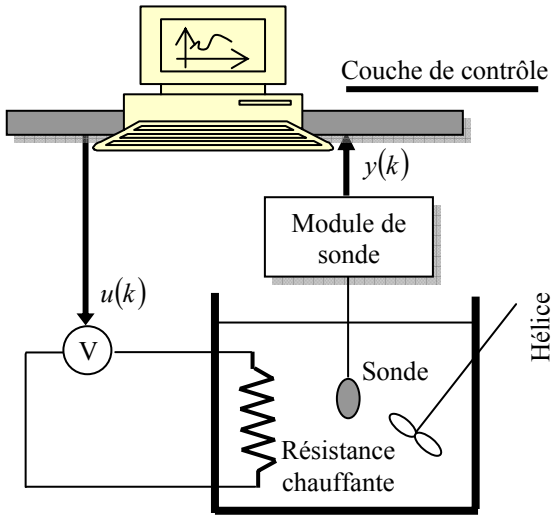


Fig. 3.19 : Structure du procédé.

Le modèle récurrent (discret) du système est donné par [163] & [305] - [306] :

$$y(k+1) = Q_1(T_s)\cdot y(k) + \frac{Q_2(T_s)}{1+\exp(0.5y(k)-\gamma)}u(k) + (1-Q_1(T_s))\cdot Y_0 + d_{ist}(k) \qquad (3.49)$$

avec :

$$Q_1(T_s) = \exp(-\alpha T_s)$$
$$Q_2(T_s) = \frac{\beta}{\alpha}(1 - \exp(-\alpha T_s)) \tag{3.50}$$

$y(k)$ et $u(k)$ sont la température du système et le signal de commande à appliquer, respectivement. Y_0 est la température ambiante. T_s est la période d'échantillonnage. α et β sont des valeurs constantes décrivant le système. $d_{ist}(k)$ est une perturbation (Fig. 3.20) définie par [281] & [307] :

$$d_{ist}(k) = p_1 \cdot \sin(p_2 \cdot k) \cdot \exp(-p_3 \cdot k) \tag{3.51}$$

Les coefficients p_i, $(i = 1, 2, 3)$ caractérisant le signal bruit sont donnés empiriquement $(p_1, p_2, p_3) = (0.5, 2, 0.01)$. Les paramètres du système utilisé dans cet exemple sont celui d'un modèle réel développé par Chia-Feng Juang [305]. Ils sont donnés par le tableau 3.10.

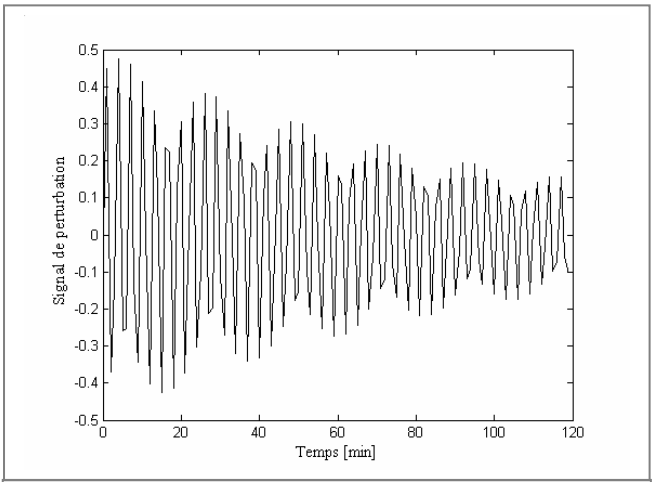


Fig. 3.20 : Forme de perturbation.

Tab. 3.10 : Paramètres du système.

Coefficient	Valeur
α	$1.0015.10^{-4}$
β	$8.6793.10^{-3}$
γ	40
T_s [s]	30
Y_0 [°C]	25.0

Les valeurs spécifiques de la loi de commande ainsi que les contraintes imposées sont données par le tableau 3.11.

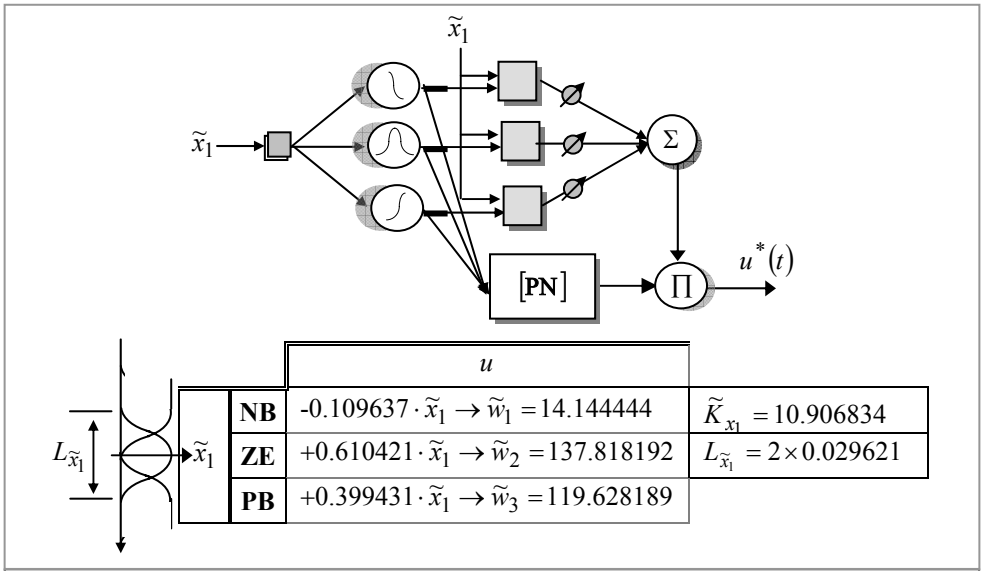
Tab. 3.11 : Valeurs spécifiques de l'algorithme de contrôle.

Caractéristique	Valeur
Taille de la population	10
Nombre de génération	100
Facteurs d'échelle	$\widetilde{K}_{x_1} \in [10, 50]$
Univers de discours	$L_{\widetilde{x}_1} \in [-1, +1]$
Poids des règles	$\widetilde{w} \in \left]0, 40 \times \max\left(\left\|u_{\min}\right|, \left|u_{\max}\right|\right)\right]$
L_chrom	8 gènes
$[u_{\min}, u_{\max}]$	$[0[\text{v}], 5[\text{v}]]$
max_k	120
$Max\_itération$	100

L'objectif de contrôle est de générer une séquence d'actions $0[\text{v}] \le u(k) \le 5[\text{v}]$ permettant au système de suivre la trajectoire de référence donnée par [305] :

$$y^d(k) = \begin{cases} 35[°C] & \text{si} \quad k \le 40 \\ 55[°C] & \text{si} \quad 40 < k \le 80 \\ 75[°C] & \text{si} \quad 80 < k \le 120 \end{cases} \tag{3.52}$$

Trois règles d'inférence sont proposées pour la construction d'une loi de commande du type NL-P. La base de connaissances obtenue après l'exécution de l'algorithme d'optimisation est donnée par la figure 3.21.



	u	
NB	$-0.109637 \cdot \widetilde{x}_1 \rightarrow \widetilde{w}_1 = 14.144444$	$\widetilde{K}_{x_1} = 10.906834$
ZE	$+0.610421 \cdot \widetilde{x}_1 \rightarrow \widetilde{w}_2 = 137.818192$	$L_{\widetilde{x}_1} = 2 \times 0.029621$
PB	$+0.399431 \cdot \widetilde{x}_1 \rightarrow \widetilde{w}_3 = 119.628189$	

Fig. 3.21 : Base de connaissances floues.

La figure 3.22 résume l'évolution de la fonction objective durant l'exécution de l'AGM.

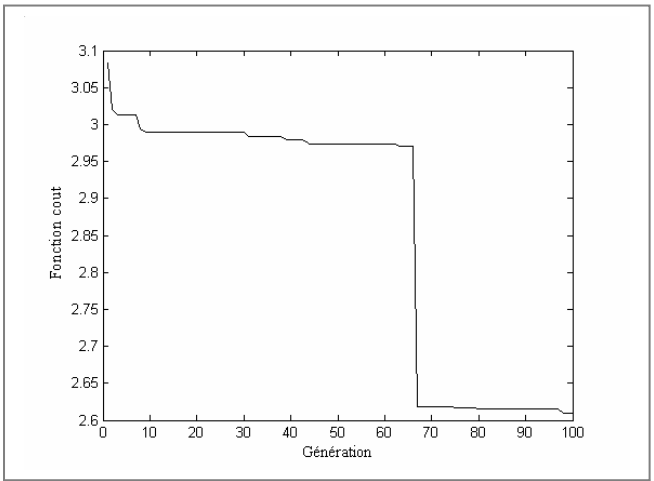


Fig. 3.22 : Evolution de la fonction coût.

Les résultats obtenus sont encourageants et justifient bien la validité de l'approche proposée. Le schéma de la figure 3.23 montre la réponse du système ainsi que la commande générée. On remarque une convergence de la réponse vers le modèle de référence.

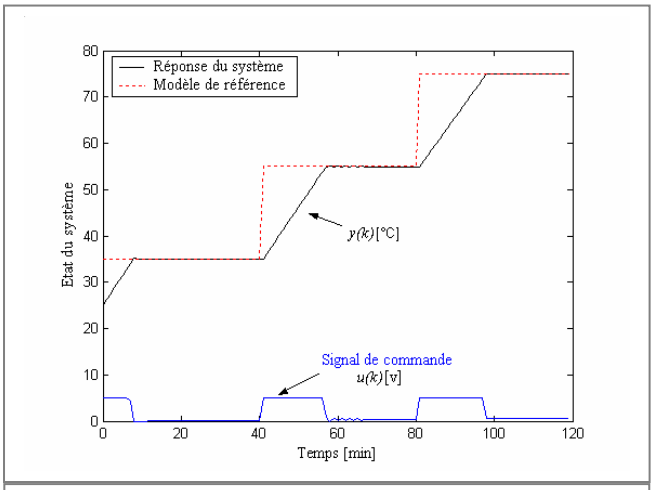


Fig. 3.23 : Performances du système.

La robustesse du contrôleur est justifiée par sa capacité de bien réagir lorsqu'on s'écarte des conditions de fonctionnement nominales (Fig. 3.24). Dans ce cas, la robustesse est testée par le changement de la trajectoire de référence avec une perturbation spontanée. Le nouveau milieu de fonctionnement sera défini par :

Modèle de référence :

$$y^d(k)=\begin{cases}34\left[^0C\right] & k\le 30\\ \left(34+0.5\cdot(k-30)\right)\left[^0C\right] & 30<k\le 50\\ \left(44+0.8\cdot(k-50)\right)\left[^0C\right] & 50<k\le 70\\ \left(60+0.5\cdot(k-70)\right)\left[^0C\right] & 70<k\le 90\\ 70\left[^0C\right] & 90<k\le 120\end{cases} \qquad (3.53)$$

Perturbation brusque :

$$d_{\text{ist}}^{\text{b}} = -5\left[^0C\right] \quad \text{à} \quad k=60\ \text{min}$$

Sur la figure 3.24, on s'aperçoit que les performances de suivi sont satisfaisantes. Une déformation du signal de sortie autour de t = 60 [min] est due à une perturbation brusque d'amplitude de 5[°C] ($Y_0 = 25\left[{}^0C\right] / *$ Condition initiale $* /$). L'évolution de la loi de commande est donnée dans la même figure.

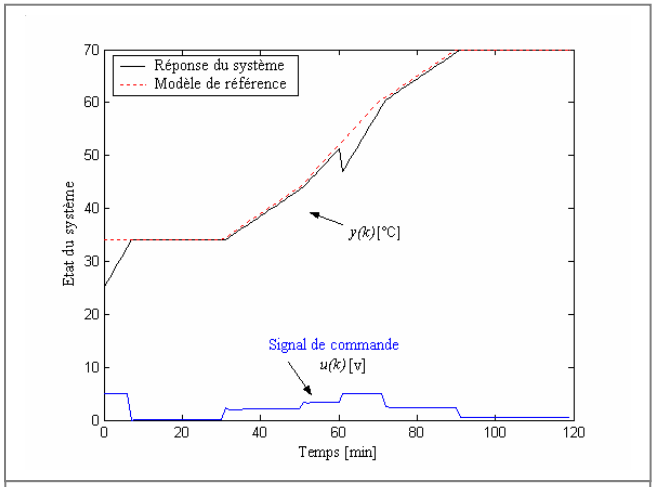


Fig. 3.24 : Test de la robustesse.

3.5.2. Commande du pendule inversé (contrôleur NL-PD)

La figure 3.25 montre le diagramme schématique du système du pendule inversé, ainsi que sa structure. La dynamique du système est caractérisée par l'expression (3.54) [170] & [308] - [309]. Les caractéristiques du pendule sont résumées dans le tableau 3.12.

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{g \cdot \sin(x_1) + \dfrac{\left[-u - m_P \cdot l \cdot x_2^2 \cdot \sin(x_1)\right]}{m_c + m_P} \cdot \cos(x_1)}{l\left(\dfrac{4}{3} - \dfrac{m_P \cdot \cos^2(x_1)}{m_c + m_P}\right)} + d_{ist} \end{aligned} \qquad (3.54)$$

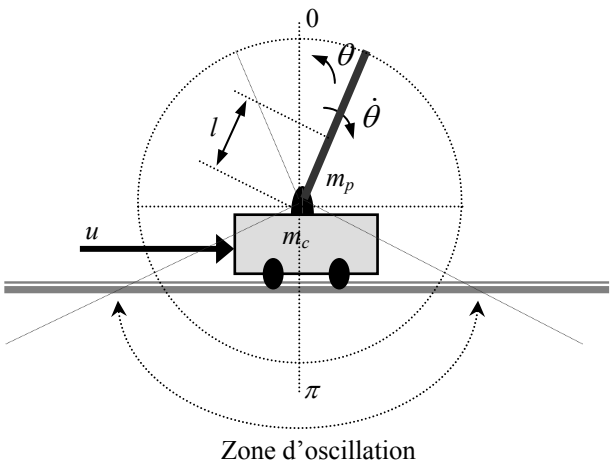


Fig. 3.25 : Structure du pendule inversé.

Tab. 3.12 : Variables fonctionnelles du pendule inversé.

Notation	Description
m_c[kg]	Masse du chariot
m_p[kg]	Masse du pendule
l[m]	Demi longueur du pendule
g[m/s^2]	Accélération gravitationnelle
$x_1 = \theta$ [rad]	L'angle du segment par rapport à l'axe vertical
$x_2 = \dot{\theta}$ [rad/s]	La vitesse angulaire du segment
u [N]	Force à appliquer au chariot

Les valeurs numériques utilisées dans la simulation sont données par le tableau suivant :

Tab. 3.13 : Valeurs des paramètres du pendule.

Paramètre	Valeur
m_c	1.0
m_p	0.1
l	0.5

Les valeurs spécifiques de l'algorithme de synthèse de la loi de commande NL-PD ainsi que les contraintes imposées sont données par le tableau 3.14.

Tab. 3.14 : Valeurs spécifiques de l'algorithme de contrôle.

Caractéristique	Valeur
Taille de la population	5
Nombre de génération	200
Facteurs d'échelle	$\widetilde{K}_{x_1}$ & $\widetilde{K}_{x_2} \in [2\cdot\pi, 6\cdot\pi]$
Univers de discours	$L_{\widetilde{x}_1}$ & $L_{\widetilde{x}_2} \in [-1, +1]$
Poids des règles	$\widetilde{w} \in]0, 20\times\max(\lvert u_{\min}\rvert, \lvert u_{\max}\rvert)]$
L_chrom	$(2+2+25\times2+25+25)$ gènes
$[u_{\min}, u_{\max}]$	$[-50[\mathrm{N}], 50[\mathrm{N}]]$
max_k	400
$Max\_itération$	150

Une perturbation d'état $d_{ist}(t)$ est ajoutée au modèle du système non linéaire (3.54). Elle est caractérisée par $(p_1, p_2, p_3) = (\pi/60, 50, 0.3)$ dans (3.51). Sa forme est donnée par la figure 3.26.

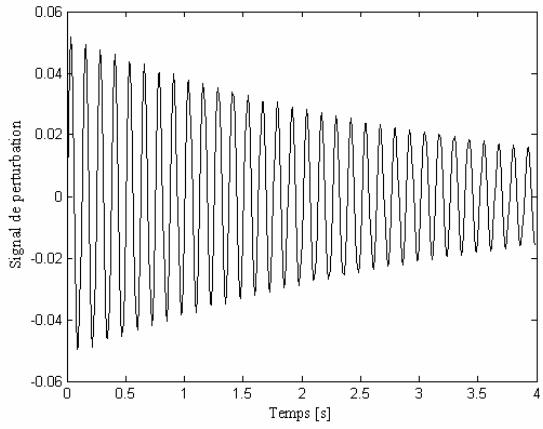


Fig. 3.26 : Perturbation d'état.

Pour illustrer les performances de la méthode de la commande proposée, nous considérons la commande en régulation et en poursuite du système de pendule inversé dont la dynamique est donnée par les équations différentielles (3.54).

3.5.2.1. Commande en régulation

Le contrôleur adapté à ce système se caractérise par 2 entrées $\tilde{x}_1 = x_1^d - x_1$, $\tilde{x}_2 = x_2^d - x_2$ et une sortie de commande u. La commande synthétisée doit accomplir 2 tâches : Maintenir le pendule dans sa position d'équilibre correspondante à un angle $\theta = 0\,[\text{rad}]$ et une vitesse angulaire $\dot{\theta} = 0\,[\text{rad/s}]$ en présence de perturbations d'état et assurer la stabilité et la robustesse du système en BF vis-à-vis aux changements des conditions de fonctionnement.

La stratégie de commande adaptée à cette situation sera représentée par la loi de commande NL-PD. Le chariot se déplace à droite ou à gauche, sans prise en compte de sa position. Les conditions initiales du système sont générées, aléatoirement.

Dans le cas illustré par la suite, on a $(x_1(0), x_2(0)) = (\pi/18\,[\text{rad}], 0\,[\text{rad/s}])$. L'évolution de la fonction objective ainsi que le nombre des règles actives au cours de l'exécution de l'algorithme de synthèse sont illustrées par les figures 3.27 et 3.28, respectivement.

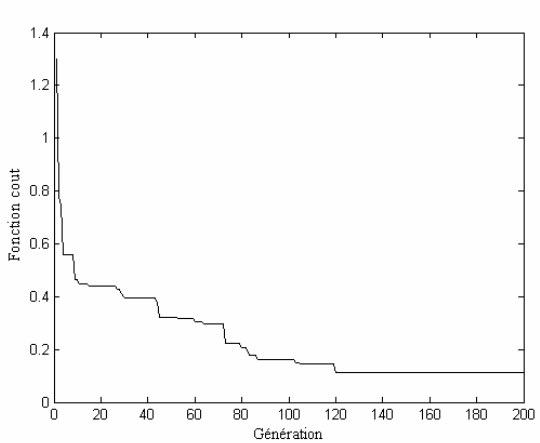


Fig. 3.27 : Evolution de la fonction coût.

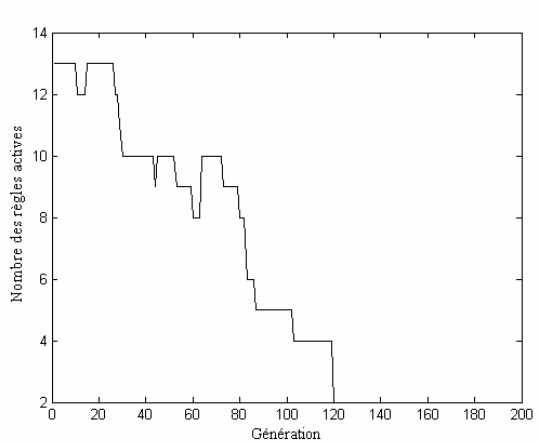


Fig. 3.28 : Evolution de la base de règles.

La base de connaissances floues obtenue à la fin de l'exécution de l'algorithme d'optimisation sera donnée par le tableau 3.15.

Tab. 3.15 : Base de connaissances floues.

Prémisse	Conséquence
Règles actives	
R_{19} : SI $\tilde{x}_1$ est PM ET $\tilde{x}_2$ est PM ALORS $u = (-0.484 \cdot \tilde{x}_1 - 0.208 \cdot \tilde{x}_2)$ R_{22} : SI $\tilde{x}_1$ est PB ET $\tilde{x}_2$ est NM ALORS $u = (+0.076 \tilde{x}_1 - 0.506 \tilde{x}_2)$	
$\left(\begin{cases} \tilde{K}_{x_1} = 10.552 \\ \tilde{K}_{x_2} = 16.503 \end{cases}, \begin{cases} L_{\tilde{x}_1} = 0.500 \\ L_{\tilde{x}_2} = 0.140 \end{cases} \right)$	$\{\tilde{w}_{19} = 725.811\}$ $\{\tilde{w}_{22} = 473.135\}$

La forme de la commande générée par le contrôleur en fonction des entrées est donnée par la figure 3.29. La figure 3.30 montre bien la convergence de la trajectoire vers l'état d'équilibre $(0\,[\text{rad}], 0\,[\text{rad/s}])$ à partir de l'état initial $(\pi/18\,[\text{rad}], 0\,[\text{rad/s}])$ (lignes en pointillées). La robustesse du contrôleur est justifiée par sa capacité de bien réagir devant le changement des conditions de fonctionnement. Le changement des conditions initiales de fonctionnement ainsi que le changement des paramètres du pendule sont 2 tests de robustesse effectués. Les conditions initiales testées sont les suivantes : $\left(\pm \underset{n=1,\ldots,7}{\underbrace{n}} \; \pi/18\,[\text{rad}], \; 0\,[\text{rad/s}] \right)$. La figure 3.30 illustre cette dernière situation. Le contrôleur devient instable à partir des conditions initiales $(x_1^d(0), x_2^d(0)) = (7{,}7\pi/18\,[\text{rad}], 0\,[\text{rad/s}])$. Le contrôleur obtenu à la fin de l'exécution de l'algorithme d'optimisation est capable d'assurer la stabilité robuste du système en BF dans la marge des angle $(-7{,}7\pi/18\,[\text{rad}] < x_1 < 7{,}7\pi/18\,[\text{rad}])$.

Au delà de la région de stabilité, un régime oscillatoire est établi. La figure 3.31 représente la réaction du contrôleur devant les changements des paramètres caractérisant le pendule inversé. On fait des tests sur les changements de la longueur du segment (0.25, 1.0, 1.5, 2.0, 2.5 et 3.0 [m]).

En analysant les figures 3.30 et 3.31, nous pouvons conclure que la stratégie proposée de conception accomplit, efficacement, les performances désirées.

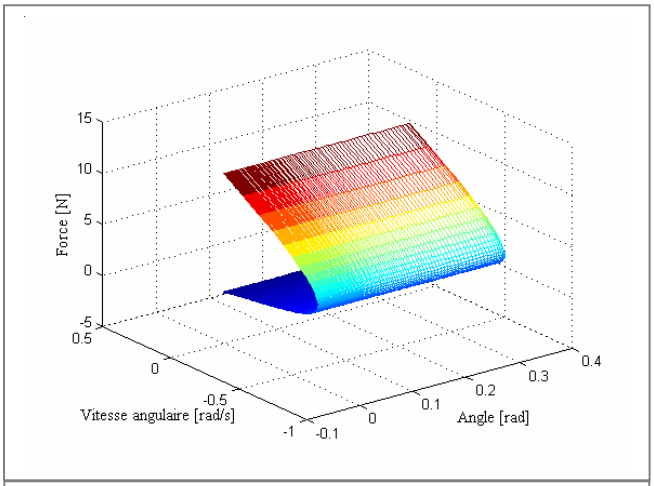


Fig. 3.29 : Commande générée en fonction des entrées du contrôleur.

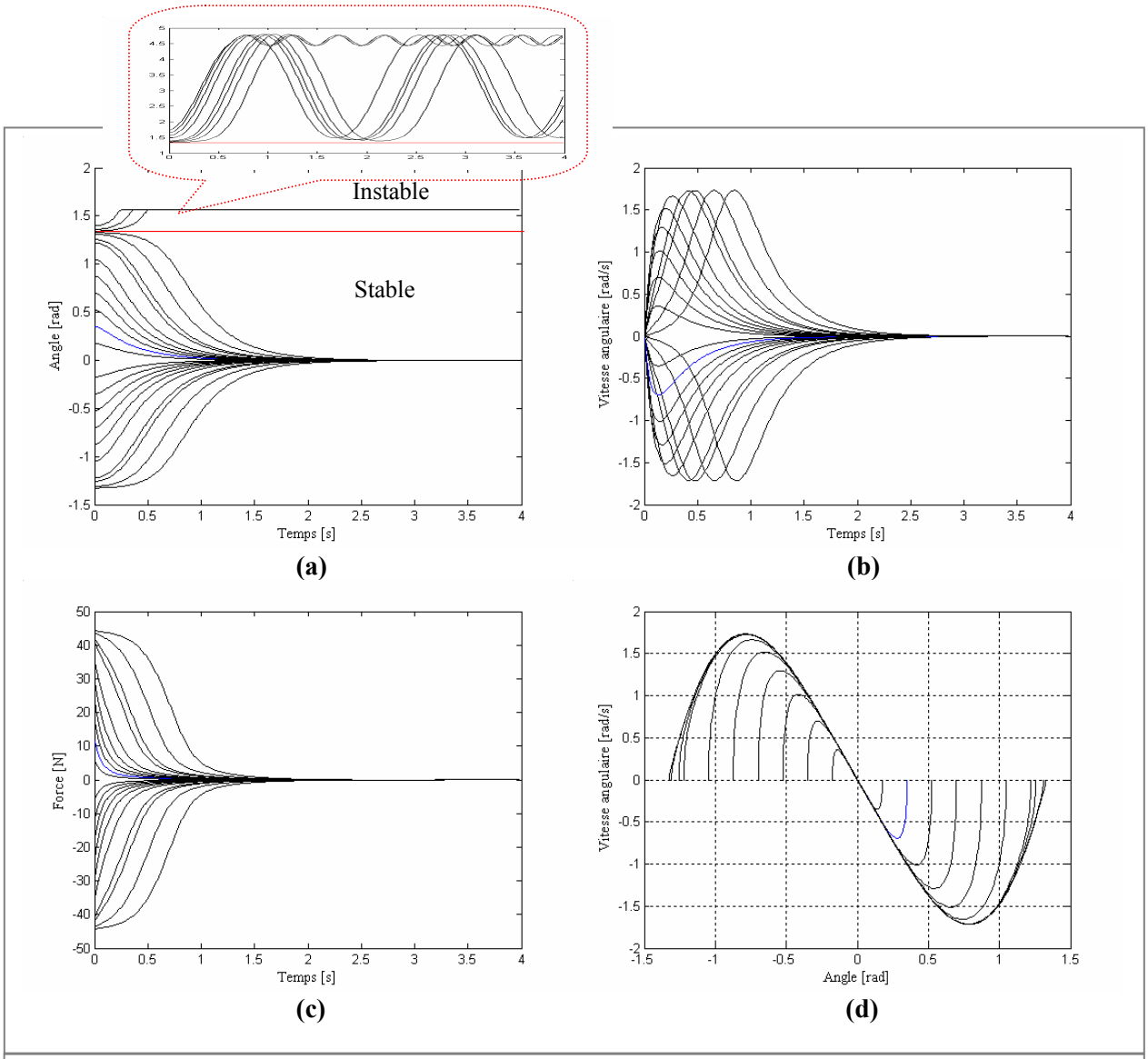


Fig. 3.30 : Réaction du système pour différentes conditions initiales :
(a) & (b) : Etat du système. **(c) :** Commandes générées. **(d) :** Plan de phase.

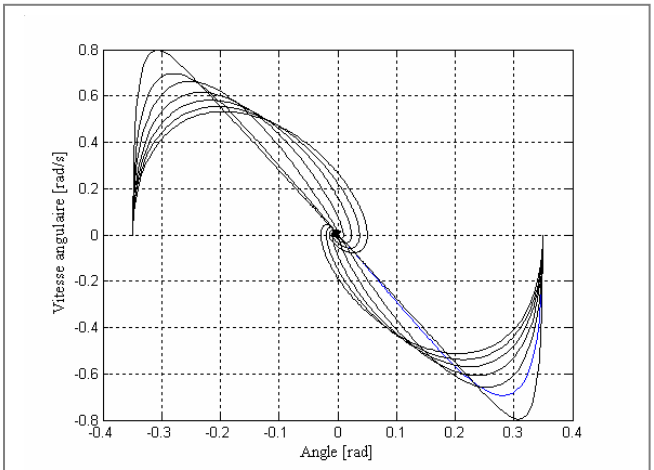


Fig. 3.31 : Réponses aux changements de poids du segment : Plan de phase.

3.5.2.2. Commande en poursuite

Dans cette partie de la simulation, le modèle de référence sera du type sinusoïdal donné par [308] :

$$\left[y^d(t), \dot{y}^d(t)\right]^T = \left[\left(\frac{\pi}{30}\right)\cdot\sin(t), \left(\frac{\pi}{30}\right)\cdot\cos(t)\right]^T \tag{3.56}$$

Dans le plan de phase $(x_1(t), x_2(t))$ de la figure 3.34f, la trajectoire de référence peut être représentée par le cercle :

$$\left(y^d(t)\right)^2 + \left(\dot{y}^d(t)\right)^2 = \left(\frac{\pi}{30}\right)^2 \tag{3.57}$$

Les conditions initiales sont choisies de sorte que $x(0) = [-\pi/18\,[\text{rad}], 0.05\,[\text{rad/s}]]^T$. L'évolution de la fonction objective ainsi que le nombre des règles actives au cours de l'exécution de l'AGM sont données par les figures 3.32 et 3.33, respectivement. La base de connaissances floues obtenue à la dernière itération de l'algorithme de synthèse hybride est illustrée par le tableau 3.16.

Les résultats des simulations effectuées montrent les bonnes performances de poursuite de la méthode proposée (Fig. 3.34). Le signal de commande en fonction de l'état du système (angle & vitesse) sera représenté par la figure 3.35.

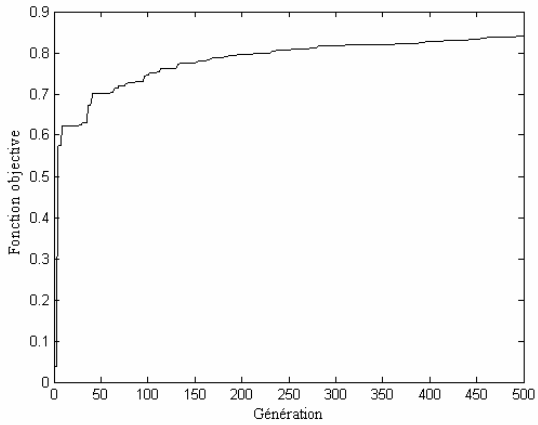


Fig. 3.32 : Evolution de la fonction objective.

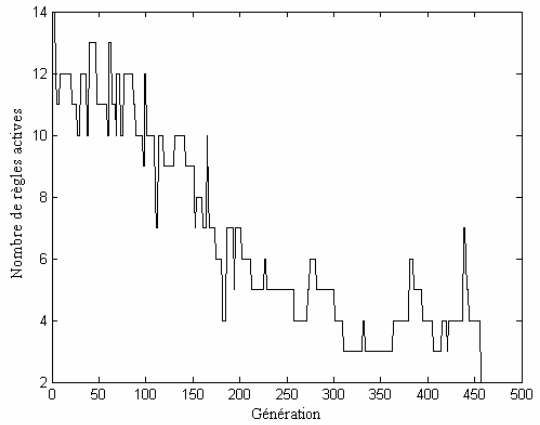


Fig. 3.33 : Evolution de la base des règles.

Tab. 3.16 : Base de connaissances floues.

Prémisse	Conséquence
Règles actives	
R_8 : SI $\tilde{x}_1$ est *NM* ET $\tilde{x}_2$ est *ZE* ALORS $u = (-0.290\cdot\tilde{x}_1 + 0.387\cdot\tilde{x}_2)$ R_{19} : SI $\tilde{x}_1$ est *PM* ET $\tilde{x}_2$ est *PM* ALORS $u = (-0.899\cdot\tilde{x}_1 - 0.435\cdot\tilde{x}_2)$	
$\left(\begin{cases}\tilde{K}_{x_1} = 7.479 & \begin{cases}L_{\tilde{x}_1} = 0.396\times 2\end{cases}\\ \tilde{K}_{x_2} = 14.072 & \begin{cases}L_{\tilde{x}_2} = 0.195\times 2\end{cases}\end{cases}\right)$	$\{\tilde{w}_8 = 333.118\}$ $\{\tilde{w}_{19} = 901.374\}$

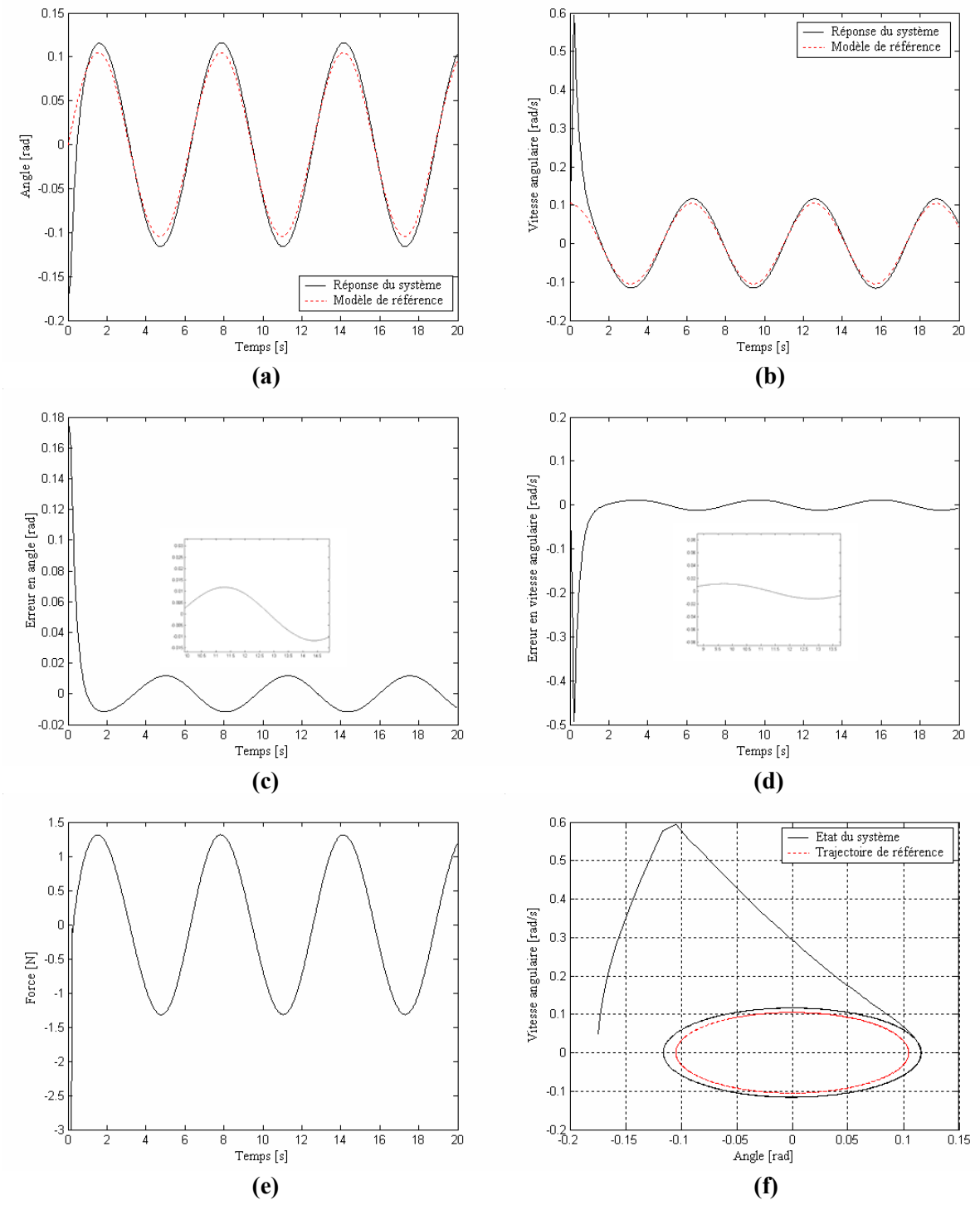


Fig. 3.34 : Performance du système : **(a) & (b) :** Etat du système. **(c) & (d):** Erreurs en angle et en vitesse angulaire. **(e) :** Signal de commande. **(f) :** Plan de phase.

La robustesse du contrôleur est testée dans le cas de changement de la longueur du pendule. Les valeurs considérées sont $\left(\underbrace{0.5}_{1}, \underbrace{1.0}_{2}, \underbrace{1.5}_{3}, \underbrace{2.0}_{4}, \underbrace{2.5}_{5}, \underbrace{3.0}_{6}[\mathrm{m}] \right)$ pour les longueurs du segment. La figure 3.36 illustre la réaction du contrôleur dans le plan de phase devant ces

changements. On remarque que les trajectoires réelles convergent vers les trajectoires désirées.

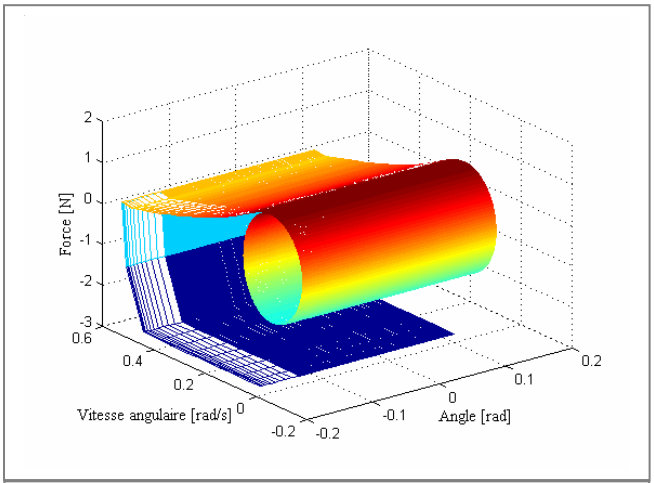


Fig. 3.35 : Commande générée en fonction des entrées du contrôleur.

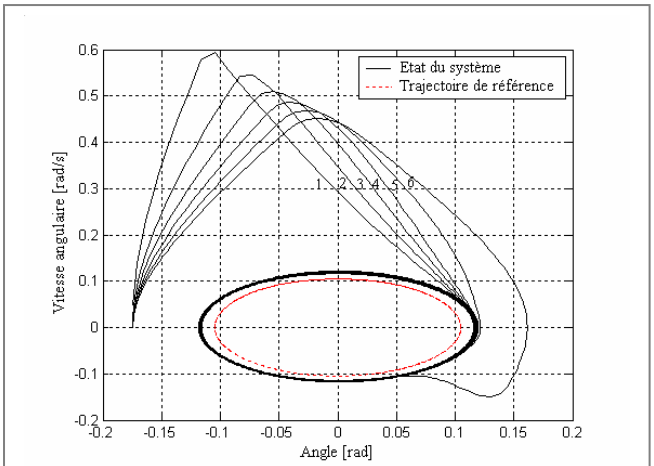


Fig. 3.36 : Réponses aux changements de poids du segment : Plan de phase.

3.5.3. Commande d'un robot manipulateur à 2 degrés de liberté (NL-PI)

3.5.3.1. Introduction

L'un des problèmes fondamentaux dans la commande des robots manipulateurs est l'aptitude du robot à suivre une trajectoire de référence, préalablement, planifiée avec une précision acceptable. Ceci est dû à la forte non linéarité, au fort couplage et à la dynamique variable du système. De plus, l'existence des incertitudes sur les paramètres des actionneurs et la structure mécanique du robot. Cependant, l'application des TIA pour le contrôle des systèmes, fortement, non linéaires fournit des solutions efficaces pour le cas des robots manipulateurs [310].

3.5.3.2. Description du bras manipulateur

Le modèle dynamique d'un bras manipulateur à *n* degrés de liberté est donné par l'équation matricielle suivante [311] :

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{3.58}$$

où $M(q)\big|_{(n\times n)}$ est la matrice d'inertie (généralement, symétrique positive définie). $C(q,\dot{q})\big|_{(n\times 1)}$ est le vecteur des forces centrifuges et Coriolis. $G(q)\big|_{(n\times 1)}$ est un vecteur de forces et couples de gravitation et $\tau\big|_{(n\times 1)}$ est le vecteur des forces extérieures exogènes (les couples). $q, \dot{q}$ et $\ddot{q}$ sont, respectivement, les vecteurs $(n\times 1)$ des positions, des vitesses et des accélérations angulaires des articulations.

La structure mécanique du bras manipulateur à 2 degrés de liberté considéré dans cette partie de simulation est donnée par le schéma de la figure 3.37. Il se compose de 2 articulations où l'articulation #1 est montée sur une base rigide à l'aide d'une charnière sans friction (frictionless hinge) et l'articulation #2 est montée à la fin de l'articulation #1 à l'aide d'un roulement à billes sans friction (frictionless ball bearing).

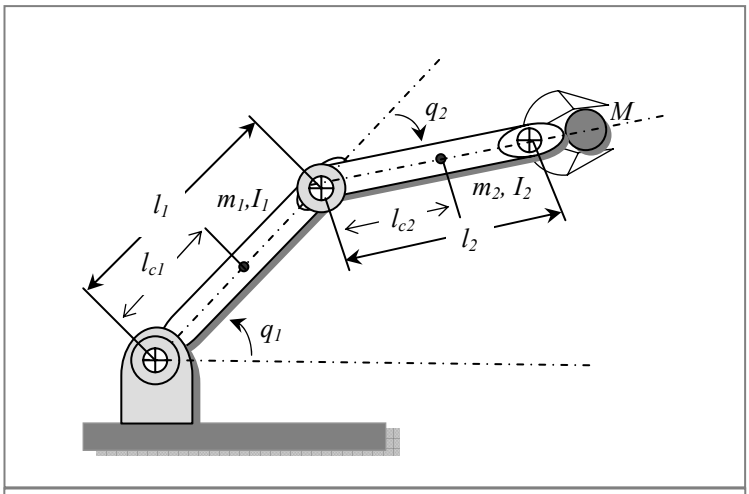


Fig. 3.37 : Structure du robot.

Le modèle découplé du bras manipulateur à 2 degrés de liberté, considéré dans ce travail, est représenté par les expressions suivantes :

$$\begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix}\begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{pmatrix} + \begin{pmatrix} -h\dot{q}_2 & -h\dot{q}_1 - h\dot{q}_2 \\ h\dot{q}_1 & 0 \end{pmatrix}\begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} + \begin{pmatrix} G_1 \\ G_2 \end{pmatrix} = \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} \tag{3.59}$$

avec :

$$\begin{aligned}
H_{11} &= m_1 l_{c1}^2 + I_1 + m_2\big|_M \left(l_1^2 + l_{c2}^2\big| + 2l_1 l_{c2}\big|_M \cos(q_2)\right) + I_2\big|_M \\
H_{12} &= H_{21} = m_2\big|_M l_1 l_{c2}\big|_M \cos(q_2) + m_2\big|_M l_{c2}^2\big|_M + I_2\big|_M \\
H_{22} &= m_2\big|_M l_{c2}^2\big|_M + I_2\big|_M \\
h &= m_2\big|_M l_1 l_{c2}\big|_M \sin(q_2) \\
G_1 &= m_1 l_{c1}\cos(q_1) + m_2\big|_M g\left(l_{c2}\big|_M \cos(q_1+q_2) + l_1\cos(q_1)\right) \\
G_2 &= m_2\big|_M l_{c2}\big|_M g\cos(q_1+q_2)
\end{aligned} \tag{3.60}$$

$$
\begin{aligned}
m_2\big|_M &= m_2\big|_{M=0} + M \\
l_{c2}\big|_M &= \frac{m_2\big|_{M=0} . l_{c2}\big|_{M=0} + M . l_2}{M + m_2\big|_{M=0}} \\
I_2\big|_M &= I_2\big|_{M=0} + \left(l_{c2}\big|_M - l_{c2}\right)^2 + M\left(l_2 - l_{c2}\big|_M\right)
\end{aligned}
\tag{3.61}
$$

Le tableau 3.17 résume les valeurs des paramètres du robot utilisé dans la simulation [312].

Tab. 3.17 : Paramètres du robot.

Paramètre	Valeur
l_1	0.3[m]
l_2	0.32[m]
m_1	2.815[Kg]
$m_2\big\|_{M=0}$	1.64[Kg]
I_1	0.0234[kgm^2]
$I_2\big\|_{M=0}$	0.0234[kgm^2]
l_{c1}	0.15[m]
$l_{c2}\big\|_{M=0}$	0.16[m]

3.5.3.3. Synthèse de la loi de commande

Pour illustrer les performances de la méthode de contrôle proposée, nous considérons la commande en poursuite d'un robot manipulateur à 2 degrés de liberté dont le modèle dynamique est donné par l'expression (3.59).

L'objectif de contrôle est de permettre au robot dont les états $x_1 = (x_{11}, x_{12}) = (q_1, \dot{q}_1)$ et $x_2 = (x_{21}, x_{22}) = (q_2, \dot{q}_2)$ de suivre les trajectoires de référence $x_1^d = (x_{11}^d, x_{12}^d) = (q_1^d, \dot{q}_1^d)$ et $x_2^d = (x_{21}^d, x_{22}^d) = (q_2^d, \dot{q}_2^d)$, respectivement. Ainsi, les erreurs de poursuite doivent être aussi petites que possible et le système en BF doit être globalement stable et robuste, c'est-à-dire, tous ses paramètres sont uniformément répartis ou bornés '*uniformly bounded*' et l'effet des perturbations externes est atténué à un niveau prescrit.

Les modèles de références pour les variables d'état du robot sont prescrits comme des fonctions de temps lisses '*smooth time functions*' [313], donnés par :

$$
\begin{aligned}
\underbrace{q_i^d(t)}_{i=1,2} &= q_i^d(t_0) + \frac{q_i^d(t_p) - q_i^d(t_0)}{2\pi} \cdot (\Theta \cdot t - \sin(\Theta \cdot t)) \\
\underbrace{\dot{q}_i^d(t)}_{i=1,2} &= \left(\frac{\pi}{2p} - \frac{\pi}{2p} \cdot \cos\left(\frac{\pi}{2p} \cdot t\right)\right) \\
\Theta &= \frac{\pi}{2p} \qquad\qquad t_0 \le t \le t_p
\end{aligned}
\tag{3.62}
$$

p est la période de temps nécessaire pour poursuivre la trajectoire entière. t_0 est le temps initial et t_p est le temps final. $q_i^d(t_0)$ et $q_i^d(t_p)$ sont, respectivement, la valeur initiale et finale désirée de la i^{eme} articulation.

Pour la première articulation $\left(q_1^d(t_0 = 0\text{s}), q_1^d(t_p = 3\text{s})\right) = (-\pi/2\text{ rad}, 0\text{ rad})$ et $\left(q_2^d(t_0 = 0\text{s}), q_2^d(t_p = 3\text{s})\right) = (0\text{ rad}, \pi/2\text{ rad})$ pour la deuxième.

Dans cette application, le processus est soumis, respectivement, à 2 entrées de commande τ_1 et τ_2. Par conséquence, 2 contrôleurs du type MISO sont nécessaires. La mise en oeuvre de cette forme de commande est illustrée par la figure 3.38. Les paramètres des 2 contrôleurs sont adaptés en utilisant la forme de commande NL-PI.

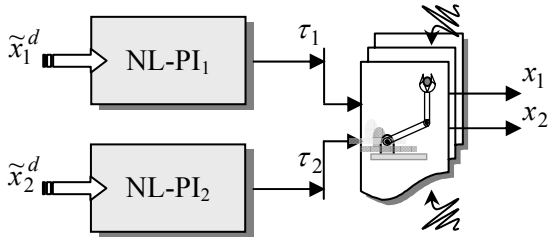


Fig. 3.38 : Structure de la commande du robot.

Le modèle du robot est soumis à des perturbations d'état identiques représentées par la figure 3.39. Cette forme est celle de l'expression (3.51) avec $(p_1, p_2, p_3) = (0.001, 100, 0.05)$. La structure du chromosome adaptée à cette situation a la même forme que celle de la figure 3.13 mais doublée, c'est à dire une partie pour NL-PI$_1$ et une autre pour NL-PI$_2$. Les valeurs spécifiques de la loi de commande ainsi que les contraintes imposées sont données par le tableau 3.18. L'évolution de la fonction objective ainsi que le nombre des règles actives au cours de l'exécution de l'AGM sont représentées, respectivement, par les figures 3.40 et 3.41. La base de connaissances floues obtenue à la fin de l'exécution de l'algorithme d'optimisation hybride est illustrée par le tableau 3.19.

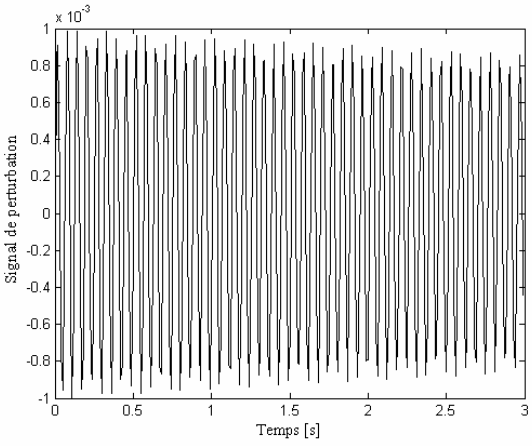


Fig. 3.39 : Signal de perturbation d'état.

Tab. 3.18 : Valeurs spécifiques de l'algorithme de contrôle.

Caractéristique	Valeur
Taille de la population	5
Nombre de génération	300
Facteurs d'échelle	$\widetilde{K}_{x_1}, \widetilde{K}_{x_2} \in [3.0, 16.0]$
Poids des règles	$\widetilde{w} \in]0, 50 \times \max(\lvert u_{\min}\rvert, \lvert u_{\max}\rvert)]$
$[u_{\min}, u_{\max}]$	[0[N], 20[N]]
max_k	300
Max _itération	200

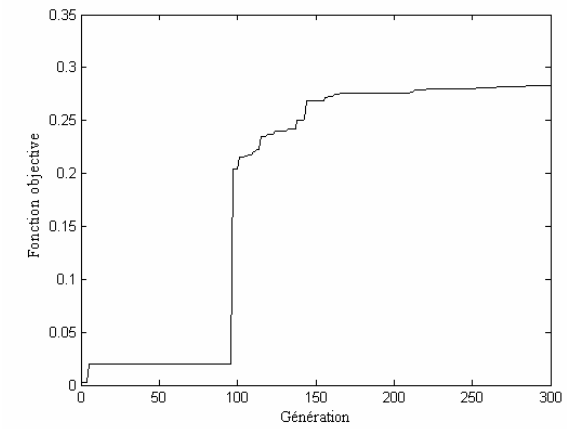


Fig. 3.40 : Evolution de la fonction objective.

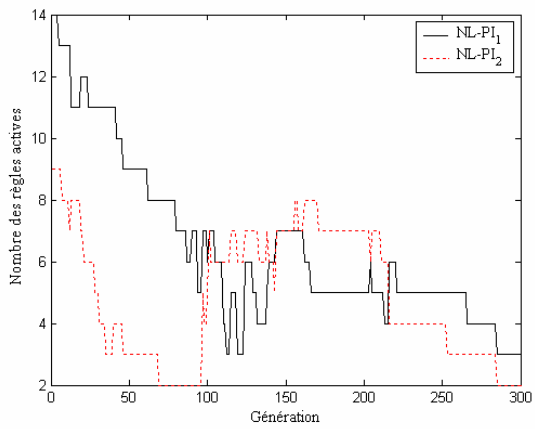


Fig. 3.41 : Evolution de la base des règles.

Tab. 3.19 : Base de connaissances floues pour NL-PI$_1$ & NL-PI$_2$.

	Prémisse	Conséquence
NL-PI$_1$	Règles actives	
	R_8 : SI $\widetilde{x}_1$ est *N*M ET $\widetilde{x}_2$ est *Z*E ALORS $u = (-0.159 \cdot \widetilde{x}_1 - 0.555 \cdot \widetilde{x}_2)$ R_{16} : SI $\widetilde{x}_1$ est *PM* ET $\widetilde{x}_2$ est *N*B ALORS $u = (-0.261 \cdot \widetilde{x}_1 - 0.281 \cdot \widetilde{x}_2)$ R_{19} : SI $\widetilde{x}_1$ est *PM* ET $\widetilde{x}_2$ est *PM* ALORS $u = (-0.016 \cdot \widetilde{x}_1 - 0.420 \cdot \widetilde{x}_2)$	
	$\left(\begin{cases}\widetilde{K}_{x_1} = 12.813 \\ \widetilde{K}_{x_2} = 6.151\end{cases}, \begin{cases}L_{\widetilde{x}_1} = 0.293 \times 2 \\ L_{\widetilde{x}_2} = 0.725 \times 2\end{cases}\right)$	$\{\widetilde{w}_8 = 449.516\}$ $\{\widetilde{w}_{16} = 209.645\}$ $\{\widetilde{w}_{19} = 335.285\}$
NL-PI$_2$	Règles actives	
	R_{19} : SI $\widetilde{x}_1$ est *PM* ET $\widetilde{x}_2$ est *PM* ALORS $u = (0.463 \cdot \widetilde{x}_1 + 0.858 \cdot \widetilde{x}_2)$ R_{25} : SI $\widetilde{x}_1$ est *PB* ET $\widetilde{x}_2$ est *PB* ALORS $u = (-0.177 \cdot \widetilde{x}_1 + 0.336 \cdot \widetilde{x}_2)$	
	$\left(\begin{cases}\widetilde{K}_{x_1} = 8.623 \\ \widetilde{K}_{x_2} = 12.612\end{cases}, \begin{cases}L_{\widetilde{x}_1} = 0.408 \times 2 \\ L_{\widetilde{x}_2} = 0.296 \times 2\end{cases}\right)$	$\{\widetilde{w}_{19} = 540.501\}$ $\{\widetilde{w}_{25} = 199.867\}$

Les réponses en positions et en vitesses articulaires sont schématisées par les figures 3.42 (a), (b), (c), (d), (e) et (f). Nous constatons que les 2 contrôleurs résultants NL-PI_1 et NL-PI_2 développent, pratiquement, un effort semblable pour réaliser l'objectif de poursuite.

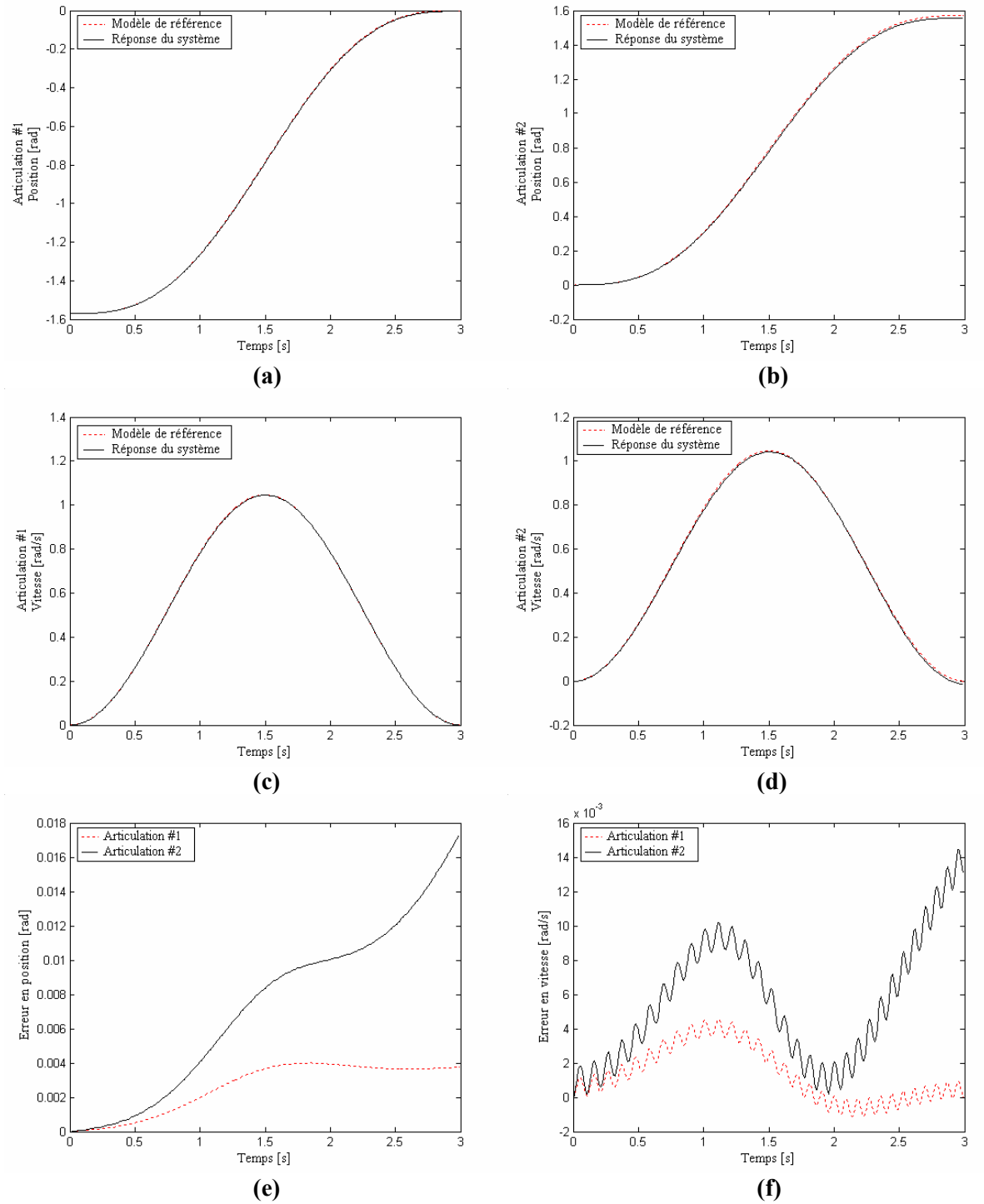


Fig. 3.42 : Etat du système : **(a) & (b) :** Positions articulaires. **(c) & (d) :** Vitesses articulaires. **(e) & (f) :** Erreurs de poursuite.

Les formes des commandes générées en fonctions des entrées des contrôleurs NL-PI_1 et NL-PI_2 sont schématisées par les figures 3.43 et 3.44, respectivement. Dans le but de tester la robustesse de cette forme de commande vis-à-vis aux perturbations éventuelles, nous avons simulé le robot avec une charge M de différents poids : $M = \{0.5, 1.5, 2.5\}[Kg]$.

L'influence de ces variations sur la précision de poursuite des trajectoires de référence est récapitulée par la figure 3.45. Ces variations ont peu d'effets sur les performances désirées. La variation de masse de charge cause des erreurs de poursuite, particulièrement, pour la deuxième articulation.

Il est visible, à partir de ces figures, que l'algorithme de commande réalise une bonne performance de poursuite avec une erreur acceptable.

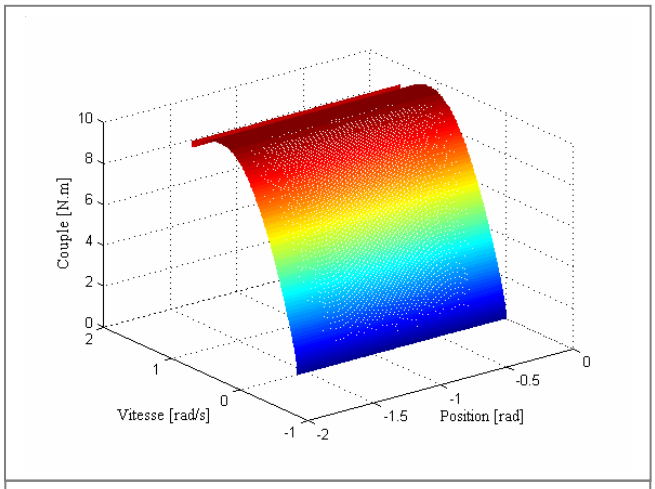


Fig. 3.43 : Couple généré pour l'articulation #1.

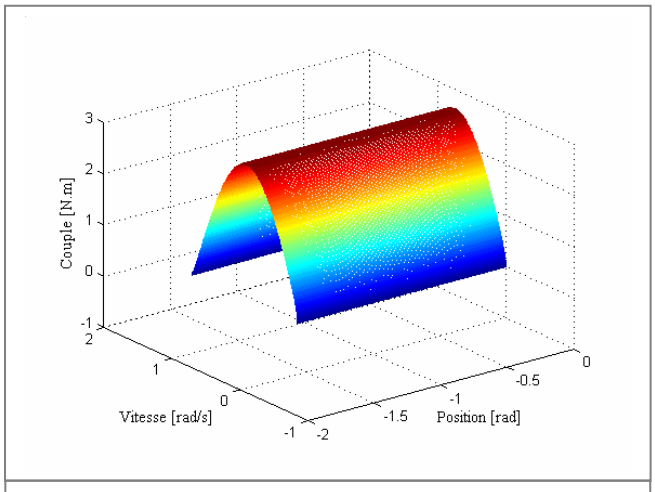


Fig. 3.44 : Couple généré pour l'articulation #2.

3.6. Conclusions

Le développement d'une stratégie de commande doit assurer non seulement la stabilité, mais aussi la robustesse en présence des phénomènes perturbateurs. Ces derniers peuvent être de nature externe (influence de l'environnement) ou de nature interne (erreurs de modélisation ou d'approximation).

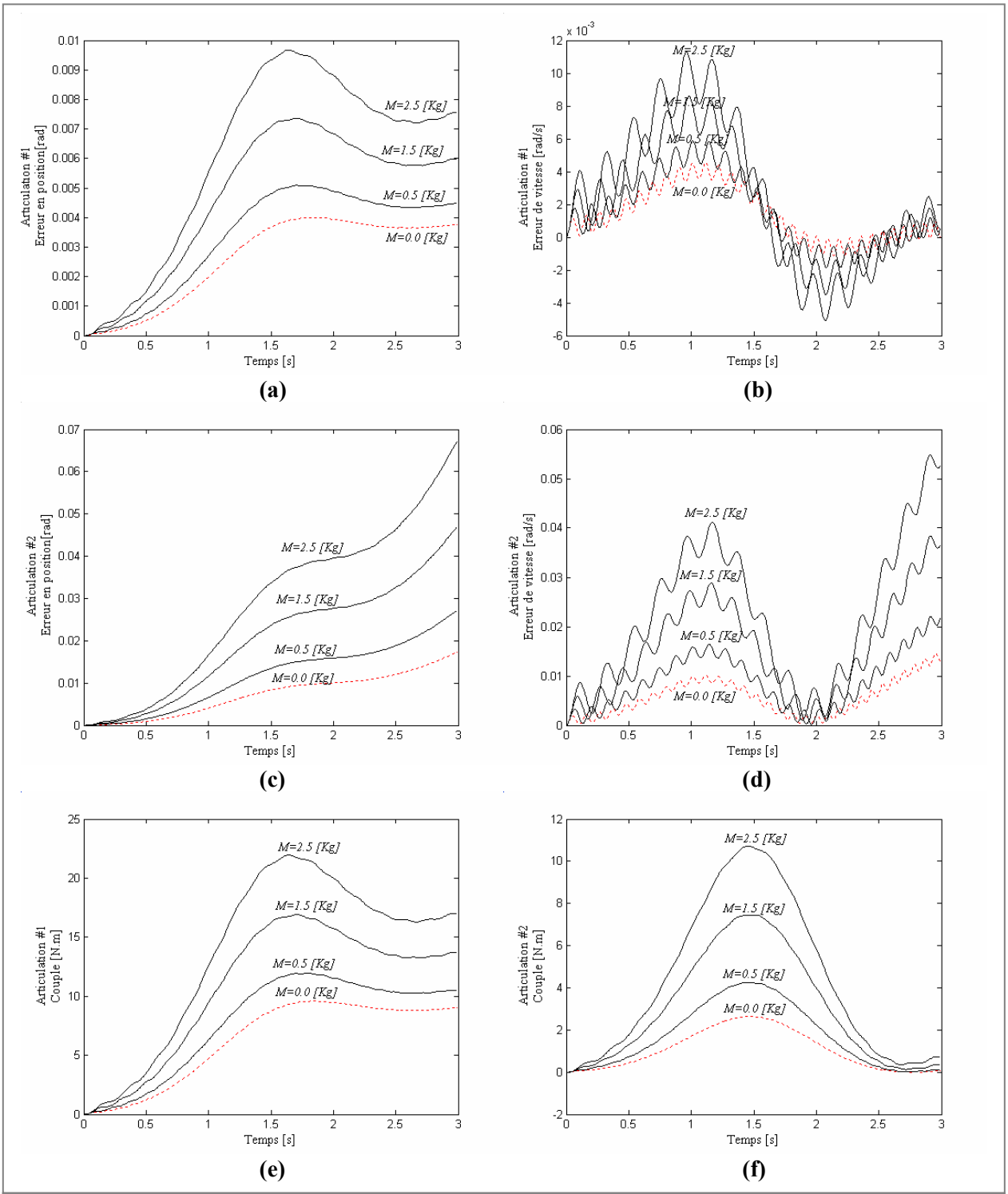


Fig. 3.45 : Test de la robustesse : **(a) & (c) :** Erreurs de positions articulaires. **(b) & (d) :** Erreurs de vitesses articulaires. **(e) & (f) :** Commandes générées.

Le problème de modélisation consiste à choisir une structure appropriée du SIF puis à concevoir des lois d'ajustement de paramètres satisfaisant un critère de performances prédéterminé par le concepteur.

Dans ce chapitre, nous avons proposé une méthodologie de synthèse d'une loi de commande simple, efficace, stable et robuste pour une classe de systèmes non linéaires. La stratégie de commande proposée, dans ce travail, repose sur l'hybridation NF. Le RBF-flou

étendu au raisonnement approximatif du type TS1 est utilisé. La modélisation du RBF-flou à partir d'une base de données numériques consiste en :

- Normalisation des variables d'entrée.
- Partitionnement des univers de discours de ces variables. Les partitions doivent être fortes.
- Génération des paramètres de TS ainsi que les poids des règles.
- Construire une base de règles réduite. La stratégie de réduction de la base de règles consiste à ne retenir que les règles qui assurent une couverture de l'espace d'entrée.
- Apprentissage fin des paramètres du réseau RBF-flou. Plus le nombre de règles sera faible, plus le SIF correspondant pourra affiner ses conclusions par des descentes de gradient complémentaires.

La réduction de la base des règles s'accompagne, inévitablement, d'une baisse des performances numériques en apprentissage. Pour remédier à ce problème, nous avons inclus des contraintes structurelles et fonctionnelles dans le critère de performances à optimiser. Le traitement des contraintes imposées nous permet d'avoir des modèles satisfaisant les objectifs de modélisation et de contrôle :

- La précision & la stabilité.
- La réduction de complexité de la loi de commande.
- La réduction de l'effort de contrôle à appliquer sur le procédé.

L'objectif de la synthèse d'une loi de commande doit satisfaire les objectifs mentionnés dans le cahier des charges. Une caractéristique importante est la stabilité du système en BF ainsi que sa robustesse vis-à-vis d'incertitudes de modélisation et/ou de perturbations vis-à-vis des variations des paramètres fonctionnels du système.

Dans le cadre de ce travail, un algorithme hybride est utilisé pour une optimisation structurelle et paramétrique du contrôleur proposé. Deux étapes d'apprentissage sont présentées :

- Une optimisation globale de l'ensemble structure/paramètres du réseau contrôleur proposé. L'algorithme utilisé est un AG à objectifs multiples.
- Après l'identification initiale de la topologie et les paramètres du réseau contrôleur, la deuxième étape de l'algorithme d'apprentissage est l'utilisation de la technique de la descente du gradient pour un réglage plus fin des paramètres des conséquences des règles.

L'AGM représente les caractéristiques suivantes :

- Taille réduite des chromosomes. La stratégie de partitionnement de l'espace d'entrée permet de rendre minimale la représentation chromosomiale.
- Concept de changement d'échelles. L'analyse des objectifs de modélisation et la notion de '*compétition imposée'* de ces objectifs est une façon de changer d'une manière dynamique l'échelle de la fonction d'évaluation de l'algorithme d'optimisation. Ce mécanisme assure une diversité des solutions dans la population de l'AGM.
- Types des mutations appliquées : non linéaire & uniforme forcée.
- Environnement de simulation dynamique (pratique virtuelle).

Il est important de souligner la nécessité pour la méthode de synthèse développée d'être peu gourmande en temps de calcul (Voir Annexe B). Les procédures de test des contraintes et la recherche des corrections 'solutions' en cas de violation de ces contraintes représentent l'inconvénient majeur de l'algorithme, en particulier la contrainte sur les règles assurant une bonne couverture de l'espace d'entrée. La lourde procédure d'optimisation et d'apprentissage peut, dans un premier temps, être évitée par une bonne initialisation des paramètres fonctionnels de l'algorithme avec une représentation

minimale. Dans ce cas, le savoir faire du concepteur 'l'expert' peut améliorer le temps de convergence de l'algorithme vers l'objectif visé.

D'après les résultats de simulation des différents procédés non linéaires testés, nous avons pu constater que la stratégie de commande proposée ne présente pas uniquement une robustesse vis-à-vis des incertitudes, mais aussi une robustesse en présence d'un changement dans les paramètres des systèmes non linéaires ainsi que le changement des conditions de fonctionnement.

Chapitre 4

Concept de multimodèle : De l'analyse à la synthèse des SIF

Résumé :

Le travail développé dans ce chapitre vise à améliorer le rapport performances / complexité des systèmes de commande floue. Le concept de multimodèle est l'élément de base dans cette partie de la thèse. La synthèse d'une loi de commande floue par l'analyse de la stabilité au sens de Lyapunov est exposée. La formulation et le développement de l'algorithme de synthèse est conçu autour du concept de multimodèle, de la commande par retour d'état, de la stabilité en BF et d'une exploration génétique de l'espace de recherche structurelle/paramétriques de la configuration de conception. Des résultats de simulations effectuées sur un bioprocédé de traitement des eaux usées dans l'industrie du papier sont donnés pour montrer les performances de l'algorithme développé.

La connaissance est le seul bien

qu'on puisse partager

sans rien en perdre.

Jacques Bonnet

(Bonnet, 2000)

Sommaire

4. Concept de multimodèle : De l'analyse à la synthèse des SIF

4.1. Introduction

Les SIF ont connus un succès considérable dans la modélisation et le contrôle des procédés industriels hautement non linéaires. L'analyse de la stabilité est plus complexe. Ceci est dû à la nature non linéaire des SIF. Néanmoins, il devrait être important de préciser qu'une des critiques principales des SIF est, juste, le manque d'une méthodologie générale pour l'analyse (aspect primitif et artisanal indéniable).

L'approche multimodèle a connue un intérêt certain depuis la publication des travaux de Johannsen & Foss [15]. L'idée de cette approche est d'appréhender le comportement non linéaire d'un système par un ensemble de modèles locaux (linéaires ou affines) caractérisant le fonctionnement du système dans différentes zones de fonctionnement.

La synthèse des lois de commande des processus modélisés grâce à l'approche multimodèle utilise, souvent, la représentation d'état afin d'étendre aux cas non linéaires les techniques de commande par retour d'état.

Les modèles de TS sont assimilables à des multimodèles. Dans les SIF, les zones de fonctionnement sont définies en termes de propositions sur les variables des prémisses. La conception des SIF par l'analyse de la stabilité a attiré l'attention de plusieurs chercheurs [134], [303] & [314]. Le concept de multimodèle est le plus souvent utilisé pour l'analyse et la synthèse des SIF. Les multimodèles représentent les systèmes non linéaires sous forme d'une interpolation entre des modèles linéaires locaux. Chaque modèle local est un système dynamique valable autour d'un point de fonctionnement. Selon l'information dont on dispose, 3 méthodes distinctes peuvent être utilisées pour l'obtention d'un multimodèle.

- **Par identification :** Cette approche est utilisée dans le cas où l'on ne dispose que des mesures des entrées et des sorties du système. Les méthodes d'estimation basées sur les moindres carrés peuvent être utilisées pour identifier les paramètres du multimodèle.
- **Par linéarisation :** Dans le cas où l'on dispose d'un modèle non linéaire explicite que l'on souhaite "simplifier" on pourra procéder par linéarisation autour de différents points de fonctionnement [15].
- **Par transformation :** Dans ce cas, on veut transformer un système non linéaire affine en la commande. On se base sur une transformation polytopique convexe de fonctions scalaires origines de la non linéarité [14].

La structure multimodèle permet de simplifier et d'étudier aisément les points suivants [315] :

- La stabilité d'un système non linéaire, grâce à l'outil numérique LMI qui permet de trouver des solutions aux équations de Lyapunov.
- La synthèse des lois de commande assurant la stabilité des boucles de commande.

Dans ce travail, nous présentons une méthodologie de synthèse permettant la prise en compte explicite de plusieurs spécifications structurelles et fonctionnelles simultanément. Nous avons exploité le concept de multimodèle pour la synthèse d'une loi de commande floue tout en assurant la stabilité en BF. L'idée consiste à déterminer un ensemble réduit de règles avec la définition des zones de fonctionnement (fonction d'appartenance) pour chaque variable d'état du système. Autour de ces zones sera établie le multimodèle par la

linéarisation de modèle du système. Le comportement global résulte de la *fusion* par le formalisme de TS de l'ensemble de comportements locaux.

4.2. Concept de multimodèle

La structure de la boucle de contrôle pour un système dynamique non linéaire est schématisée par la figure 4.1.

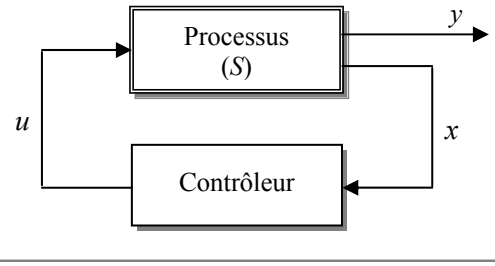


Fig. 4.1 : Structure de contrôle.

La représentation d'état décrivant le comportement dynamique du processus est donnée par le modèle suivant :

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) \\ y(t) &= g(x(t), u(t)) \end{aligned} \tag{4.1}$$

$x(t) \in \Re^n$ est le vecteur d'état. $u(t) \in \Re^m$ est le vecteur d'entrée (de commande). $y(t) \in \Re^q$ est le vecteur de sortie. $f(\cdot) \in \Re^{n \times n}$ et $g(\cdot) \in \Re^{q \times m}$ sont des fonctions non linéaires. Afin de faciliter l'analyse des systèmes, une étape de décomposition de ces derniers en sous systèmes élémentaires interconnectés est, souvent, proposée.

Le système non linéaire (4.1) doit être représenté par un multimodèle, composé de plusieurs modèles locaux linéaires ou affines. Chaque modèle local est obtenu en linéarisant le système non linéaire (4.1) autour d'un point de fonctionnement arbitraire $(x_i, u_i) \in \Re^n \times \Re^m$, que l'on peut l'écrire autour de (x_i, u_i) sous la forme matricielle suivante [316] :

$$\begin{aligned} \dot{x}(t) &= \overbrace{A(x_i, u_i)}^{A_i} \cdot x(t) + \overbrace{B(x_i, u_i)}^{B_i} \cdot u(t) + N_x^i \\ y(t) &= \underbrace{C(x_i, u_i)}_{C_i} \cdot x(t) + \underbrace{D(x_i, u_i)}_{D_i} \cdot u(t) + N_y^i \end{aligned} \tag{4.2}$$

avec :

$$A_i = \left.\frac{\partial f(x,u)}{\partial x}\right|_{\substack{x=x_i \\ u=u_i}} = \begin{bmatrix} \left.\frac{\partial f_1}{\partial x_1}\right|_{\substack{x=x_i \\ u=u_i}} & \cdots & \left.\frac{\partial f_1}{\partial x_n}\right|_{\substack{x=x_i \\ u=u_i}} \\ \vdots & \ddots & \vdots \\ \left.\frac{\partial f_n}{\partial x_1}\right|_{\substack{x=x_i \\ u=u_i}} & \cdots & \left.\frac{\partial f_n}{\partial x_n}\right|_{\substack{x=x_i \\ u=u_i}} \end{bmatrix}, \quad B_i = \left.\frac{\partial f(x,u)}{\partial u}\right|_{\substack{x=x_i \\ u=u_i}} = \begin{bmatrix} \left.\frac{\partial f_1}{\partial u_1}\right|_{\substack{x=x_i \\ u=u_i}} & \cdots & \left.\frac{\partial f_1}{\partial u_m}\right|_{\substack{x=x_i \\ u=u_i}} \\ \vdots & \ddots & \vdots \\ \left.\frac{\partial f_n}{\partial u_1}\right|_{\substack{x=x_i \\ u=u_i}} & \cdots & \left.\frac{\partial f_n}{\partial u_m}\right|_{\substack{x=x_i \\ u=u_i}} \end{bmatrix}$$

$$C_i = \left.\frac{\partial g(x,u)}{\partial x}\right|_{\substack{x=x_i\\u=u_i}} = \begin{bmatrix} \left.\frac{\partial g_1}{\partial x_1}\right|_{\substack{x=x_i\\u=u_i}} & \cdots & \left.\frac{\partial g_1}{\partial x_n}\right|_{\substack{x=x_i\\u=u_i}} \\ \vdots & \ddots & \vdots \\ \left.\frac{\partial g_q}{\partial x_1}\right|_{\substack{x=x_i\\u=u_i}} & \cdots & \left.\frac{\partial g_q}{\partial x_n}\right|_{\substack{x=x_i\\u=u_i}} \end{bmatrix},\ D_i = \left.\frac{\partial g(x,u)}{\partial u}\right|_{\substack{x=x_i\\u=u_i}} = \begin{bmatrix} \left.\frac{\partial g_1}{\partial u_1}\right|_{\substack{x=x_i\\u=u_i}} & \cdots & \left.\frac{\partial g_1}{\partial u_m}\right|_{\substack{x=x_i\\u=u_i}} \\ \vdots & \ddots & \vdots \\ \left.\frac{\partial g_n}{\partial u_1}\right|_{\substack{x=x_i\\u=u_i}} & \cdots & \left.\frac{\partial g_n}{\partial u_m}\right|_{\substack{x=x_i\\u=u_i}} \end{bmatrix}$$

$$N_x^i = f(x_i,u_i) - A_i \cdot x_i - B_i \cdot u_i,\ N_y^i = g(x_i,u_i) - C_i \cdot x_i - D_i \cdot u_i \tag{4.3}$$

Les matrices $A(x_i,u_i) \in \Re^{n\times n}$ et $B(x_i,u_i) \in \Re^{n\times m}$ sont les jacobiens de $f(\cdot)$ par rapport à x et u évalué en (x_i,u_i), respectivement. $C(x_i,u_i) \in \Re^{q\times n}$ et $D(x_i,u_i) \in \Re^{q\times m}$ sont les jacobiens de $g(\cdot)$ par rapport à x et u évalué en (x_i,u_i), respectivement.

Autour du point (x_i,u_i), le comportement du système s'approxime donc par les équations d'état suivantes :

$$\begin{aligned} \dot{x}(t) &= f(x_i,u_i) + A_i(x(t)-x_i) + B_i(u(t)-u_i) \\ y(t) &= g(x_i,u_i) + C_i(x(t)-x_i) + D_i(u(t)-u_i) \end{aligned} \tag{4.4}$$

Le point (x_i,u_i) est un point de fonctionnement (point de polarisation) si $f(x_i,u_i) \cong 0$. Si $u_i = 0$, on parle d'un point d'équilibre. On remarque que le système n'évolue pas si on maintient la commande $u(t) = u_i$. Dans ce cas, la sortie $y(t)$ a pour valeur $y(t) = y_i = g(x_i,u_i)$. Autour du point de fonctionnement (x_i,u_i), le système S admet pour système tangent [317] :

$$\begin{aligned} \dot{x}(t) &= A_i(x(t)-x_i) + B_i(u(t)-u_i) \\ y(t) &= y_i + C_i(x(t)-x_i) + D_i(u(t)-u_i) \end{aligned} \tag{4.5}$$

Posons $\tilde{u}(t) = u(t) - u_i$, $\tilde{x}(t) = x(t) - x_i$ et $\tilde{y}(t) = y(t) - y_i$. Ces vecteurs sont appelés les variations de $u(t), x(t)$ et $y(t)$, respectivement. Pour des petites variations $\tilde{u}, \tilde{x}$ et $\tilde{y}$, on a :

$$\begin{aligned} \dot{\tilde{x}}(t) &= A_i \cdot \tilde{x}(t) + B_i \cdot \tilde{u}(t) \\ \tilde{y}(t) &= C_i \cdot \tilde{x}(t) + D_i \cdot \tilde{u}(t) \end{aligned} \tag{4.6}$$

Le système (4.6) est appelé système linéarisé de S autour du point de fonctionnement (x_i,u_i). Chaque sous modèle n'est qu'une description locale du comportement du système. On obtient donc une représentation linéaire du système dynamique non linéaire (4.1) au voisinage d'un point de fonctionnement (x_i,u_i). Il est important de souligner que le modèle linéarisé ainsi obtenu dépend du point de fonctionnement autour duquel la linéarisation a été effectuée [316].

En conclusion, les modèles de TS sont assimilables à des multimodèles. Dans cette situation, les zones de fonctionnement sont définies en termes de propositions sur les variables des prémisses.

4.3. Méthodes d'analyse de la stabilité

Le problème de la stabilité des systèmes dynamiques est un sujet de préoccupation majeur du travail des mathématiciens, des physiciens et des ingénieurs depuis le dernier siècle. Les critères d'analyse de la stabilité peuvent être classés en 2 catégories :

- Les critères fréquentiels (diagrammes de Bode, de Nyquist, ...etc.).
- Les critères temporels (méthodes de Lyapunov, ...etc.).

D'une façon générale, les systèmes non linéaires sont les plus difficiles à étudier parce qu'il est délicat d'en faire l'étude dans le domaine fréquentiel (fonction de transfert difficile à exploiter). Les modèles flous du type TS sont composés de modèles linéaires (les règles) interconnectés par des fonctions scalaires non linéaires ayant la propriété de somme convexe. L'analyse des systèmes de contrôle flous pour des applications spécifiques dépend en grande partie des connaissances et de l'expertise de l'opérateur humain (concepteur).

La synthèse d'un SIF présente un aspect primitif et artisanal indéniable. Le choix des fonctions d'appartenance, de leur nombre, de la défuzzification, voir même de l'inférence floue, est très arbitraire. Finalement, il est important de mentionner la difficulté de garantir la cohérence & l'interprétabilité des règles floues. En particulier pour des systèmes multivariables où le nombre des règles devient très élevé. Les différentes approches de l'analyse des SIF sont mentionnées dans le schéma de la figure 4.2 [318].

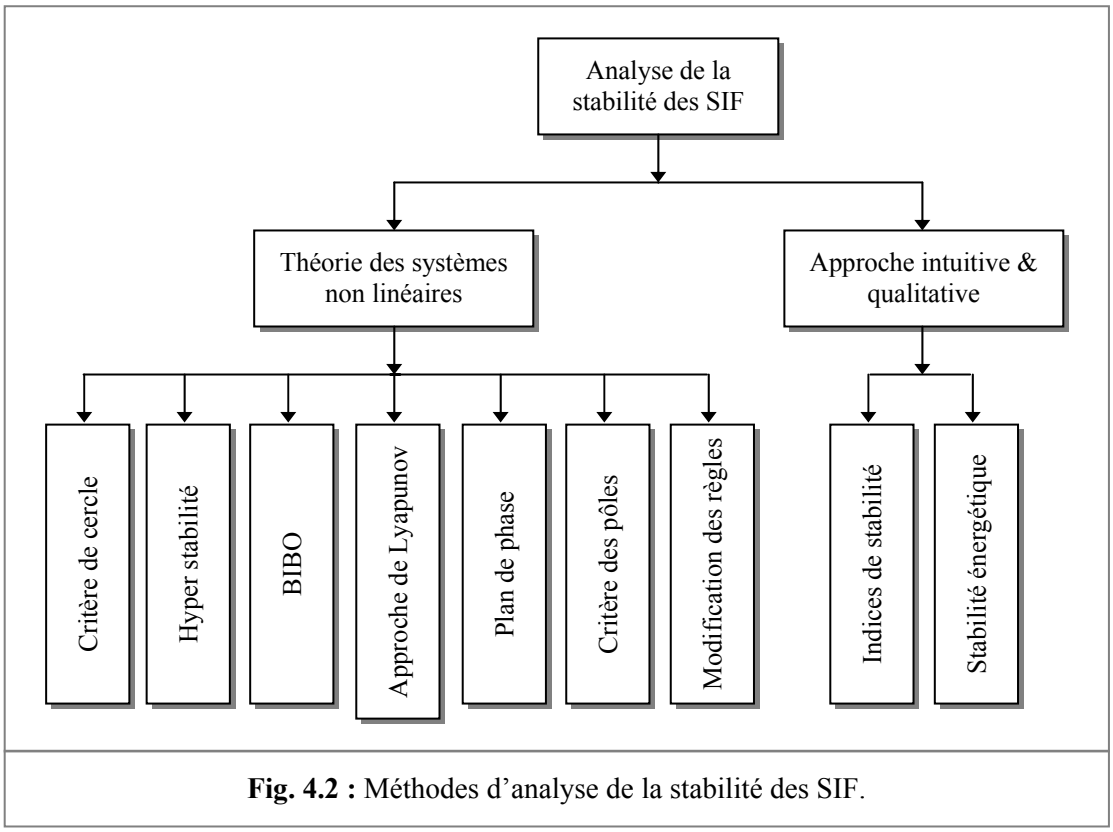


Fig. 4.2 : Méthodes d'analyse de la stabilité des SIF.

Pour plus d'informations et compléments algorithmiques des méthodes d'analyse des SIF, nous citons le livre de F. Cuesta qui regroupe en détail l'ensemble des méthodes et des travaux dans ce contexte [303]. Dans ce travail, nous avons insisté sur 2 approches différentes pour l'analyse. La méthode BIBO décrite dans le chapitre précédent. Les résultats obtenus sont satisfaisants. Dans le reste de la thèse, nous avons développé un

algorithme de synthèse d'un système de contrôle flou en se basant sur l'analyse de la stabilité au sens de Lyapunov.

Des méthodes qualitatives pour l'analyse de stabilité sont basées sur l'application de la théorie qualitative des systèmes non linéaires. Dans ces méthodes, l'analyse consiste à une interprétation géométrique du vecteur des variables (des trajectoires) associées au procédé à commander et le système de contrôle flou. L'analyse des trajectoires permet d'identifier les différents dynamiques du système en BF (cycle limite, oscillations, attracteurs, ...etc.) d'un côté. De l'autre côté, des conclusions qualitatives au sujet de la stabilité ou de l'instabilité du système en BF sont tirées. Par conséquent, ces informations peuvent être utilisées pour modifier la base des règles floues pour assurer le comportement dynamique désiré [318]. Dans ce contexte, l'une des premières approches utilise le plan de phase [319]. Les modèles utilisés sont décrits par l'expression (3.25). La réduction aux systèmes du second ordre est la représentation de base pour l'analyse de la stabilité. Le plan de phase est alors découpé en fonction de la répartition des entrées du contrôleur flou et une analyse qualitative (graphique) permet de caractériser le comportement du système en BF.

4.3.1. Méthodes basées sur l'approche de Lyapunov

La méthode la plus utilisée pour l'analyse de la stabilité des SIF est basée sur l'approche énergétique de Lyapunov. Le livre de Kazuo Tanaka représente une référence de base. Il regroupe l'ensemble des travaux dans le cadre de l'analyse de la stabilité des systèmes de contrôle flous non linéaires.

Le modèle de TS est l'objet principal de ces travaux en tenant compte de l'analyse de la stabilité, des méthodologies 'systématiques' de conception, des spécifications des performances voulues, robustesse, facteurs d'optimalité et l'implémentation [134].

La méthode directe de Lyapunov s'appuie sur une observation physique fondamentale : *Si l'énergie totale (scalaire) d'un système (linéaire ou non linéaire) est continûment dissipée, alors on peut espérer que le système tend vers un poins d'équilibre. Ainsi, l'idée de Lyapunov est d'examiner la variation d'une fonction scalaire pour étudier la stabilité d'un système donné* [13].

Par la suite, la méthodologie de l'analyse de la stabilité au sens de Lyapunov sera décrite en détail.

4.3.2. Passage d'un modèle non linéaire à un modèle flou de TS

Les modèles flous de TS sont composés de modèles linéaires interconnectés par des fonctions scalaires non linéaires ayant la propriété de somme convexe. Plusieurs approches permettant le passage d'un modèle non linéaire à une représentation (dans un domaine compact des variables d'état) sous forme de TS (représentant TS d'un système) [320] :

- Dans le cas où aucun modèle de connaissance du système n'est disponible, une identification du modèle flou TS à partir d'un jeu de mesures entrées/sorties peut être réalisée. Les travaux de Takagi & Sugeno en 1985 représentent la plate forme de ce type de modélisation et d'identification [5].
- Une deuxième méthode consiste à linéariser un modèle non linéaire ou un système autour de plusieurs points de fonctionnement et de relier ces linéarisés par des fonctions non linéaires [14] - [15], [134] & [321] - [322]. Cette façon de faire est plus proche des techniques dites multimodèles et présentent l'inconvénient de ne pas représenter exactement le modèle non linéaire. C'est cette façon de procéder qui sera utilisée dans le reste du mémoire.

Les conditions de stabilité et de stabilisation de ces modèles sont, généralement, basées sur les fonctions de Lyapunov [134]. Des algorithmes issus, notamment, de l'optimisation convexe sont, généralement, utilisés pour résoudre ces conditions dans le cas où elles sont exprimées sous la forme de LMI [18]. L'analyse de la stabilité des SIF du type TS nécessite :

- La disponibilité du modèle flou.
- La disponibilité d'un contrôleur. Le contrôleur lui même est un système flou dont la structure correspond à celle du modèle flou du procédé. Cette idée introduite par Wang et al. est appelée dans le cas d'un SIF de type TS : Compensation parallèle distribuée (PDC) [16].

La plupart des travaux concernant la stabilité des modèles TS (discrets & continus) sont basés sur l'approche PDC [322] - [328]. Le principe de l'approche PDC (resp. PPDC proposé par D. H. Lin et al. consiste à calculer une loi de commande statique linéaire par retour d'état $u(t) = -K \cdot x(t)$ pour chaque sous modèle (modèle local) du modèle flou du système [329]. D'une manière générale, on peut rechercher une loi de commande permettant d'agréger une famille de loi de commandes linéaires locales $u_i(t) = -K_i \cdot x(t)$ (resp. $u_i(t) = -K_i \cdot k_i \cdot x(t)$). $k_i \in \Re$ est un coefficient de proportionnalité relatif à chacun des modèles locaux. La forme générale de la boucle de contrôle par retour d'état est illustrée par la figure 4.3. Le vecteur (resp. la matrice) des gains de contre réaction sont, généralement, déterminés par les techniques de commande linéaires conventionnelles. Les plus utilisées sont celles de placement des pôles, commande optimale et robuste. La synthèse de la loi de commande revient donc à déterminer pour chaque modèle local le vecteur (resp. matrice) des gains de la boucle de retour K_i (resp. K_i & k_i).

A. Soukkou et al. ont exploité la méthode PPDC pour la commande d'un réacteur chimique (CSTR). L'approche génétique est utilisée pour la construction d'une configuration structurelle et paramétrique optimale de la loi de commande. Les résultats obtenus sont satisfaisants et montrent la validité de l'approche PPDC à apprentissage génétique [330].

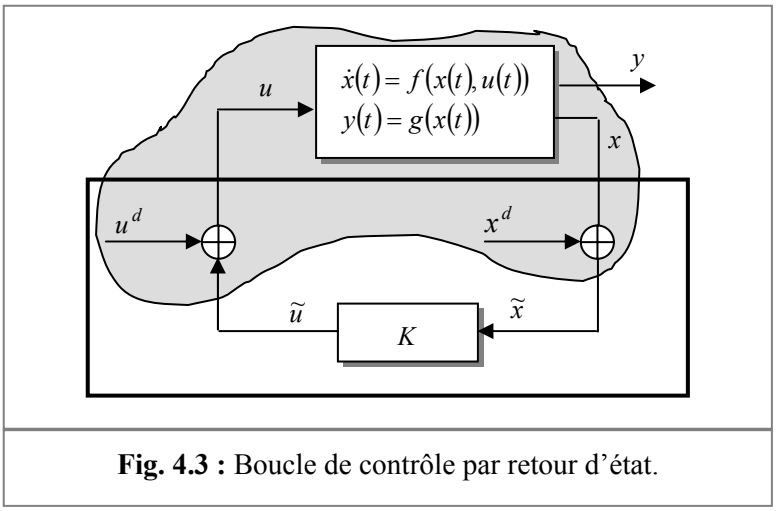


Fig. 4.3 : Boucle de contrôle par retour d'état.

Pratiquement, le vecteur (resp. la matrice) des gains de contre réaction $\{K_i\}_{i=1}^{i=N}$ est uniformément borné [331] :

$$\sup_{1 \le i \le N} \|K_i\| \le M \le \infty \quad \& \quad M > 0 \tag{4.7}$$

avec $\|\cdot\|$ dénote la norme spectrale d'une matrice de dimension finie (la plus grande valeur singulière de la matrice).

4.3.3. Analyse de la stabilité

Le concept PDC est schématisé par la figure 4.4 [16] & [323]. Trois blocs principaux caractérisent cette stratégie d'analyse et de synthèse des lois de commande : Le modèle flou issu du processus à commander par linéarisation autour des points de fonctionnement choisis, le contrôleur flou et le système à étudier.

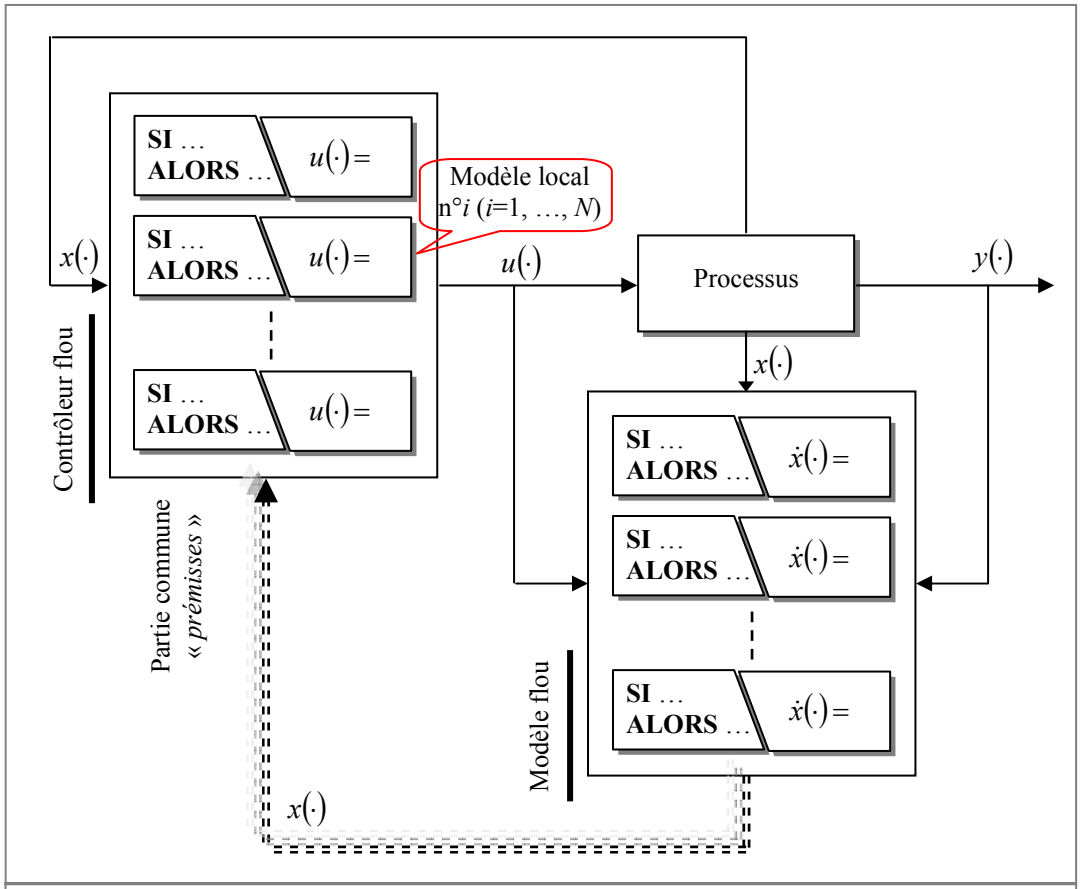


Fig. 4.4 : Représentation du concept PDC.

Les modèles locaux issus d'une linéarisation autour de N points de fonctionnement (x_i, u_i) sont supposés, individuellement, définis par la forme générale suivante :

$$\begin{aligned} \dot{x}(t) &= A_i \cdot x(t) + B_i \cdot u(t) \\ y(t) &= C_i \cdot x(t) + D_i \cdot u(t) \quad i = 1, ..., N \end{aligned} \tag{4.8}$$

A_i, B_i, C_i et D_i sont des matrices constantes appropriées. $A_i \in \Re^{n \times n}$ est la matrice d'état ou d'évolution. $B_i \in \Re^{n \times m}$ est la matrice d'entrée. $C_i \in \Re^{q \times n}$ est la matrice de sortie ou d'observation. $D_i \in \Re^{q \times m}$ est la matrice de transmission directe des entrées sur les sorties. La matrice D_i est généralement nulle, car, pour un système réel, les sorties ne peuvent, en général, pas réagir instantanément aux variations des entrées.

Les modèles flous du type TS décrit par des règles floues « **SI** … **ALORS** » qui représentent des modèles linéaires sont donnés par [331] :

$$\underbrace{R^i}_{i=1,\ldots,N} : \textbf{SI } \left(x_1 \textbf{ est } Z_1^i\right) \ldots \textbf{ET } \left(x_n \textbf{ est } Z_n^i\right) \textbf{ ALORS } \begin{cases} \dot{x}(t) = A_i \cdot x(t) + B_i \cdot u(t) \\ y(t) = C_i \cdot x(t) \end{cases} \tag{4.9}$$

$\left[x_1, \ldots, x_n\right]^T \in X \in \Re^n$ sont les variables des prémisses (variables d'état et/ou d'entrées). $Z_1^i \sim Z_n^i$ sont les sous ensembles flous des prémisses. Il est à souligner que A_i et B_i, $(i = 1, \ldots, N)$ sont uniformément bornées, c'est-à-dire, qu'il existe 2 constantes N_A et N_B tels que :

$$\sup_{1 \le i \le N} \|A_i\| \le N_A \le \infty \quad \& \quad \sup_{1 \le i \le N} \|B_i\| \le N_B \le \infty \tag{4.10}$$

Les modèles flous sont supposés localement contrôlables ($\forall i \in \{1, \ldots, N\}$ les paires (A_i, B_i) sont contrôlables) et localement observables ($\forall i \in \{1, \ldots, N\}$ les paires (A_i, B_i) sont observables) [332].

Soit $\mu_i(x(t)) \in [0,1]$ une fonction permettant d'indiquer la validité (degré d'appartenance) du modèle local n° i (la i^{eme} règle floue). Le processus étudié peut alors être décrit par une représentation d'état (4.11) où chaque modèle local y contribue avec un degré de pertinence mesuré par un coefficient de validité $w_i(x(t))$. C'est dans ce sens, qu'un modèle global a été introduit pour représenter l'évolution du système étudié sur tout l'espace d'état (domaine de fonctionnement) [333] :

$$\begin{aligned} \dot{x}(t) &= A_g(\cdot) \cdot x(t) + B_g(\cdot) \cdot u(t) \\ y(t) &= C_g(\cdot) \cdot x(t) + D_g(\cdot) \cdot u(t) \end{aligned} \tag{4.11}$$

où les matrices $A_g(\cdot)$, $B_g(\cdot)$, $C_g(\cdot)$ et $D_g(\cdot)$ sont obtenues par fusion des matrices caractéristiques des modèles locaux (après le calcul des inférences et la défuzzification) :

$$\begin{aligned} A_g(\cdot) &= \sum_{i=1}^{N} w_i(x(t)) A_i(\cdot) \\ B_g(\cdot) &= \sum_{i=1}^{N} w_i(x(t)) B_i(\cdot) \\ C_g(\cdot) &= \sum_{i=1}^{N} w_i(x(t)) C_i(\cdot) \\ D_g(\cdot) &= \sum_{i=1}^{N} w_i(x(t)) D_i(\cdot) \end{aligned} \tag{4.12}$$

Le multimodèle flou (4.11) doit être complètement défini sur le domaine de fonctionnement du système (propriété de convexité), c'est-à-dire :

$$0 \le \mu_i(x(t)) \le 1 \quad \& \quad \sum_{i=1}^{N} \mu_i(x(t)) \ne 0 \quad \forall i$$
$$w_i(x(t)) = \frac{\mu_i(x(t))}{\sum_{i=1}^{N} \mu_i(x(t))} \ge 0 \quad \& \quad \sum_{i=1}^{N} w_i(x(t)) = 1 \tag{4.13}$$

Concernant le contrôleur flou, la i^{eme} règle floue s'écrit [331] :

$$\underbrace{R^i}_{i=1,\dots,N} : \textbf{SI } (x_1 \textbf{ est } Z_1^i) \dots \textbf{ET } (x_n \textbf{ est } Z_n^i) \textbf{ ALORS } u(t) = -K_i \cdot [x(t) - x^d] \tag{4.14}$$

$[x_1^d, \dots, x_n^d]^T$ est le vecteur des états désirés.

Chaque règle de contrôle R^i est réalisée à partir de celle du modèle flou. Le contrôleur flou garde les mêmes prémisses ainsi que les mêmes fonctions d'appartenance.

Il s'agit de déterminer une loi de commande $u(t)$ permettant de stabiliser le multimodèle défini par l'expression (4.11). En se basant sur l'approche PDC, l'expression générale de la loi de commande est la suivante :

$$u(t)|_{PDC} = -\sum_{j=1}^{N} w_j(x(t)) \cdot K_j \cdot [x(t) - x^d]$$
$$u(t)|_{PPDC} = -\sum_{j=1}^{N} w_j(x(t)) \cdot K_j \cdot k_j \cdot [x(t) - x^d] \tag{4.15}$$
$$w_j(x(t)) = \frac{\mu_j(x(t))}{\sum_{j=1}^{N} \mu_j(x(t))}$$

Les fonctions $\mu_j(x(t))$ sont les mêmes que celles du multimodèle flou (4.11). La configuration commune entre le modèle et le contrôleur flou de la figure 4.4 est le partage de la partie prémisse des règles floues.

La réalisation du contrôleur flou consiste à déterminer les gains de contre réaction K_i dans les conclusions des règles satisfaisants les conditions de stabilité mentionnées par la suite du chapitre. La substitution du système (4.15) dans le système (4.11) donne le système en BF suivant :

$$\begin{aligned} \dot{x}(t) &= \left(\sum_{i=1}^{N} A_i \cdot w_i(x(t))\right) \cdot x(t) + \left(\sum_{i=1}^{N} B_i \cdot w_i(x(t))\right) \cdot u(t) \\ &= \left(\sum_{i=1}^{N} A_i \cdot w_i(x(t)) - \left(\sum_{i=1}^{N} B_i \cdot w_i(x(t))\right) \cdot \left(\sum_{j=1}^{N} K_j \cdot w_j(x(t))\right)\right) \cdot x(t) = f(x(t)) \end{aligned} \tag{4.16}$$

Les valeurs de $\mu_i(x(t))$ (resp. $w_i(x(t))$) sont définies de sorte que le modèle (4.11) possède une solution unique dépendante des conditions initiales de fonctionnement $(x(0) = x_0)$. Notons que le nombre des modèles locaux dépend de la précision de modélisation souhaitée, de la complexité du système non linéaire et du choix de la structure des fonctions d'activation [13].

L'obtention des contrôleurs flous et de la stabilité globale du système en BF (Fig. 4.4) à partir des méthodes de Lyapunov peut se faire en résolvant un système d'équations linéaires matricielles (LMI) [18]. Pour l'analyse de la stabilité, le choix le plus classique consiste à sélectionner une fonction de Lyapunov (fonction de Lyapunov candidate) sous forme quadratique :

$$V(x(t)) = x(t)^T \cdot P \cdot x(t) \quad P > 0 \ \& \text{ symétrique} \tag{4.17}$$

$x(t)$ représente la solution de l'équation d'état (4.1). Dans le cas général, il n'existe pas de méthode pour trouver toutes les fonctions candidates de Lyapunov. Dès lors, la théorie de Lyapunov conduit à des conditions suffisantes de stabilité dont le pessimisme dépend de la forme particulière imposée à la fonction $V(x(t))$ et de la structure du système. Cependant, il existe des familles de fonctions de Lyapunov souvent utilisées et dont l'adoption dépend de la nature du système à étudier (systèmes linéaires, systèmes continus par morceaux, systèmes à retard, systèmes linéaires incertains, ...etc.) [315]. Etant donnée P symétrique positive définie, on a :

$$\begin{aligned} &V(x(t)) > 0 && \forall\, x(t) \neq 0 \\ &V(x(t)) = 0 && \forall\, x(t) = 0 \quad \& \\ &V(x(t)) \to \infty \quad \text{si} && \|x(t)\| \to \infty \end{aligned} \tag{4.18}$$

Dans les conditions (4.18), l'origine $x(t) = 0$ est un point d'équilibre, globalement asymptotiquement, stable pour le système en BF si :

$$\dot{V}(x(t)) = \left(x(t)^T \cdot P \cdot \dot{x}(t) + \dot{x}(t)^T \cdot P \cdot x(t)\right) < 0 \tag{4.19}$$

$\dot{V}(x(t))$ représente la dérivée par rapport au temps de la fonction de Lyapunov le long de la trajectoire de la variable $x(t)$. En remplaçant $\dot{x}(t)$ par son expression (4.16) dans l'équation (4.19), on a [334] :

$$\begin{aligned} \dot{V}(x) = x(t)^T \cdot P \cdot \left(\frac{\sum_{i=1}^{N} A_i \cdot \mu_i(x(t))}{\sum_{i=1}^{N} \mu_i(x(t))} \times \left\{ \frac{\sum_{j=1}^{N} \mu_j(x(t))}{\sum_{j=1}^{N} \mu_j(x(t))} \right\} - \left(\frac{\sum_{i=1}^{N} B_i \cdot \mu_i(x(t))}{\sum_{i=1}^{N} \mu_i(x(t))} \right) \cdot \left(\frac{\sum_{j=1}^{N} K_j \cdot \mu_j(x(t))}{\sum_{j=1}^{N} \mu_j(x(t))} \right) \right) \cdot x(t) \\ + x(t)^T \cdot \left(\frac{\sum_{i=1}^{N} A_i \cdot \mu_i(x(t))}{\sum_{i=1}^{N} \mu_i(x(t))} - \left(\frac{\sum_{i=1}^{N} B_i \cdot \mu_i(x(t))}{\sum_{i=1}^{N} \mu_i(x(t))} \right) \cdot \left(\frac{\sum_{j=1}^{N} K_j \cdot \mu_j(x(t))}{\sum_{j=1}^{N} \mu_j(x(t))} \right) \right)^T \cdot P \cdot x(t) \end{aligned} \tag{4.20}$$

On dénote par $\sum_{i,j}$ la somme sur toutes les combinaisons possibles de i et j, $(i = 1, ..., N \ \& \ j = 1, ..., N)$, l'expression (4.20) sera :

$$\dot{V}(x) = x(t)^T \cdot P \cdot \left(\frac{\sum_{i,j} A_i \cdot \mu_i(x(t)) \cdot \mu_j(x(t))}{\sum_{i,j} \mu_i(x(t)) \cdot \mu_j(x(t))} - \frac{\sum_{i,j} B_i \cdot K_j \cdot \mu_i(x(t)) \cdot \mu_j(x(t))}{\sum_{i,j} \mu_i(x(t)) \cdot \mu_j(x(t))} \right) \cdot x(t)$$
$$+ x(t)^T \cdot \left(\frac{\sum_{i,j} A_i \cdot \mu_i(x(t)) \cdot \mu_j(x(t))}{\sum_{i,j} \mu_i(x(t)) \cdot \mu_j(x(t))} - \frac{\sum_{i,j} B_i \cdot K_j \cdot \mu_i(x(t)) \cdot \mu_j(x(t))}{\sum_{i,j} \mu_i(x(t)) \cdot \mu_j(x(t))} \right)^T \cdot P \cdot x(t) \quad (4.21)$$

Finalement, la dérivée temporelle de $V(x)$ s'écrit :

$$\dot{V}(x) = x(t)^T \cdot \left(\frac{\sum_{i,j} \left[P \cdot (A_i - B_i \cdot K_j) + (A_i - B_i \cdot K_j)^T \cdot P \right] \cdot \mu_i(x(t)) \cdot \mu_j(x(t))}{\sum_{i,j} \mu_i(x(t)) \cdot \mu_j(x(t))} \right) \cdot x(t) \quad (4.22)$$

De (4.13), on a :

$$0 \le \frac{\mu_i(x(t)) \cdot \mu_j(x(t))}{\sum_{i,j} \mu_i(x(t)) \cdot \mu_j(x(t))} \le 1 \quad (4.23)$$

Ce qui conduit à :

$$\dot{V}(x(t)) \le \left(\sum_{i,j} x(t)^T \cdot \left[P \cdot (A_i - B_i \cdot K_j) + (A_i - B_i \cdot K_j)^T \cdot P \right] \right) \cdot x(t) \quad (4.24)$$

L'inégalité (4.24) est vérifiée si :

$$P \cdot (A_i - B_i \cdot K_j) + (A_i - B_i \cdot K_j)^T \cdot P < 0 \qquad \forall i, j = 1, ..., N \quad (4.25)$$

Théorème [134] **:**

Le système complet (en BF) continu est globalement asymptotiquement stable via la loi de commande PDC (4.15) *s'il existe une matrice* P *commune positive définie* $(P > 0)$ *et des matrices* K_i *qui vérifient :*

$$P \cdot (A_i - B_i \cdot K_j) + (A_i - B_i \cdot K_j)^T \cdot P < 0 \quad (4.26)$$

Pour tout $i = 1, ..., N$ *et* $j = 1, ..., N$.
Le terme $A_i - B_i \cdot K_j$ *est dit croisé quand* $i \neq j$ *et dominant quand* $i = j$. *La condition* (4.26) *peut être réécrite sous forme :*

$$\begin{aligned} &(A_i - B_i \cdot K_i)^T \cdot P + P \cdot (A_i - B_i \cdot K_i) < 0 && (i = 1, ..., N) \\ &G_{ij}^T \cdot P + P \cdot G_{ij} < 0 && (1 \le i < j \le N) \\ &G_{ij} = \frac{A_i - B_i \cdot K_j + A_j - B_j \cdot K_i}{2} \end{aligned} \quad (4.27)$$

Une simplification, particulièrement, intéressante du résultat ci-dessus est obtenue si toutes les B_i sont identiques. Dans ce cas, l'énoncé du théorème précédent se réduit à ce qui suit [331] :

Supposons que $B_1 = B_2 = ... = B_N$. *Le système complet (en BF) continu est globalement asymptotiquement stable via la loi de commande PDC* (4.15) *s'il existe une matrice* P *commune positive définie* $(P > 0)$ *et des matrices* K_i *qui vérifient :*

$$\begin{aligned} &(A_i - B \cdot K_i)^T \cdot P + P \cdot (A_i - B \cdot K_i) < 0 && (i = 1, ..., N) \\ &G_{ij}^T \cdot P + P \cdot G_{ij} < 0 && (1 \le i < j \le N) \\ &G_{ij} = \frac{A_i - B \cdot K_j + A_j - B \cdot K_i}{2} \end{aligned} \qquad (4.28)$$

Remarque :

Dans le cas des systèmes discrets, la même démarche est valable, il suffit de remplacer $x(t)$ par l'état échantillonné $x(k)$ et la dérivée $\dot{x}(t)$ par la translation $x(k+1)$.

La condition (4.26) constitue un système d'inégalités matricielles linéaires (LMI) en P. Pour la résolution de ce type de problèmes, il existe actuellement des algorithmes de calculs efficaces. Le livre «*Linear Matrix Inequalities in System and Control Theory*» est une source bibliographique importante dans ce cadre [18].

Des outils issus de l'optimisation convexe, plus particulièrement des LMI sont utilisés pour la détermination de la matrice P et des gains de contre réaction K_i, $(i = 1, ..., N)$ satisfaisant l'inégalité (4.26) [331].

Certains outils LMI sont disponibles à l'aide du logiciel MATLAB «*LMI Control Toolbox*». Les problèmes d'optimisation convexes peuvent être résolus par différents types de méthodes :

- Méthodes du type simplexe.
- Méthodes des points intérieurs [335].
- Méthodes évolutionnaires [323] & [336] - [338].

Durant ces dernières années, de nombreux travaux concernant la synthèse des lois de commande pour les multimodèles ont été publiés. Diverses approches sont utilisées pour résoudre des problèmes de stabilité et de stabilisation [323], [331] - [332] & [336] - [338].

4.4. Méthodologie de synthèse

Le potentiel offert par le concept du multimodèle permet l'utilisation de toutes les techniques associées à la théorie des systèmes linéaires. Cette forme est bien adaptée au cas des systèmes multivariables. Cependant, si le nombre des règles N est important, il devient plus difficile de trouver P et K satisfaisant les conditions de stabilité.

La détermination du nombre des modèles capables de représenter, convenablement, un système non linéaire et la position des points d'équilibre étaient toujours des problèmes cruciaux qui sont en relation directe avec la nature du système. Pour éviter d'obtenir un nombre de règles trop important, nous avons proposé un algorithme de synthèse par l'emploi d'un AGM qui abouti à un modèle flou composé d'un nombre minimal de règles.

La stratégie de conception proposée dans ce travail est schématisée par la figure 4.5. Les blocs I_1, I_2, I_3 et I_4 sont des interrupteurs. Le tableau 4.1 illustre cette stratégie.

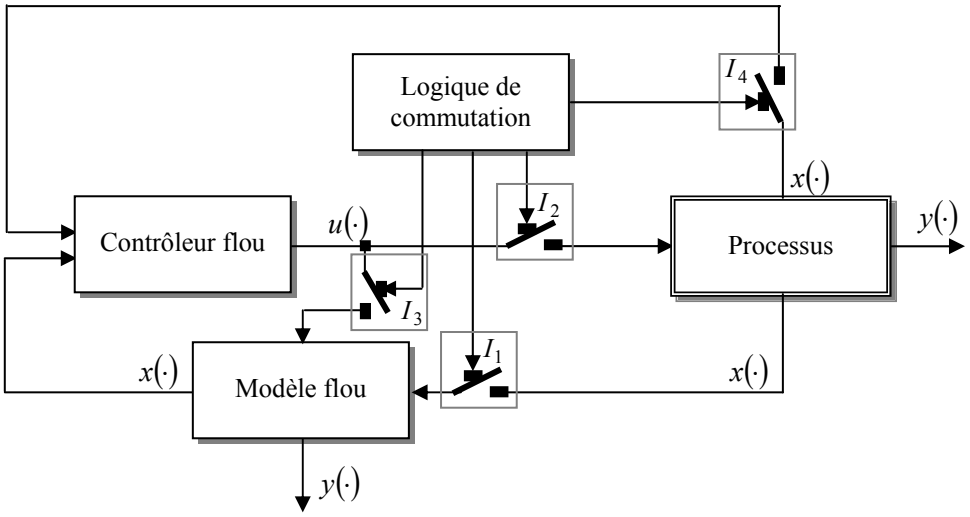


Fig. 4.5 : Stratégie de conception.

Tab. 4.1 : Stratégie de commutation adoptée pour la conception.

Phase	Logique de commutation
Etape 1	I_1 I_2 I_3 I_4 0 1 0 1
	• Déterminer le nombre des règles. • Définir les points de fonctionnement cruciaux.
Etape 2 & 3	I_1 I_2 I_3 I_4 1 0 1 0
	• Linéarisation du système autour des points de fonctionnement prédéfinis. • Synthèse de la loi de commande vérifiant les conditions de stabilité au sens énergétique de Lyapunov.

Le principe consiste en 3 étapes :

- **Etape 1 :** Initialisation de la base des règles. Cette étape ne concerne que la partie commande par retour d'état. Dans cette phase, l'AGM présenté dans le chapitre précédent avec quelques modifications permet de trouver :
  - Un nombre minimal des règles floues assurant l'efficacité de la loi de commande par retour d'état.
  - Définition des fonctions d'appartenance (zones de fonctionnement) intervenants dans la détermination des points de fonctionnement les plus significatifs.

  La détermination du nombre de modèles (nombre des règles) permettant de mieux représenter le système non linéaire ainsi que le choix de la position des points de fonctionnement (points d'équilibre) ont une grande importance et sont en relation directe avec la nature du système à commander.

- **Etape 2 :** Construction des modèles locaux du système par linéarisation de modèle du système autour des points de fonctionnement les plus significatifs. Ces points sont déterminés à partir de la partie prémisse qui est commune entre le contrôleur et le modèle. Les centres des fonctions d'appartenance (zones de fonctionnement) correspondants aux définitions des variables d'état représentent les points de fonctionnement autour duquel la linéarisation sera établie. L'exemple de la figure 4.6 est une démonstration de la procédure de spécification des zones de fonctionnement les plus significatives.
- **Etape 3 :** Etablissement des conditions de stabilité du modèle complet. L'approche génétique est adaptée dans cette étape. La méthode LMI est remplacée par une exploration génétique de l'espace de recherche des paramètres justifiant les conditions de stabilité.

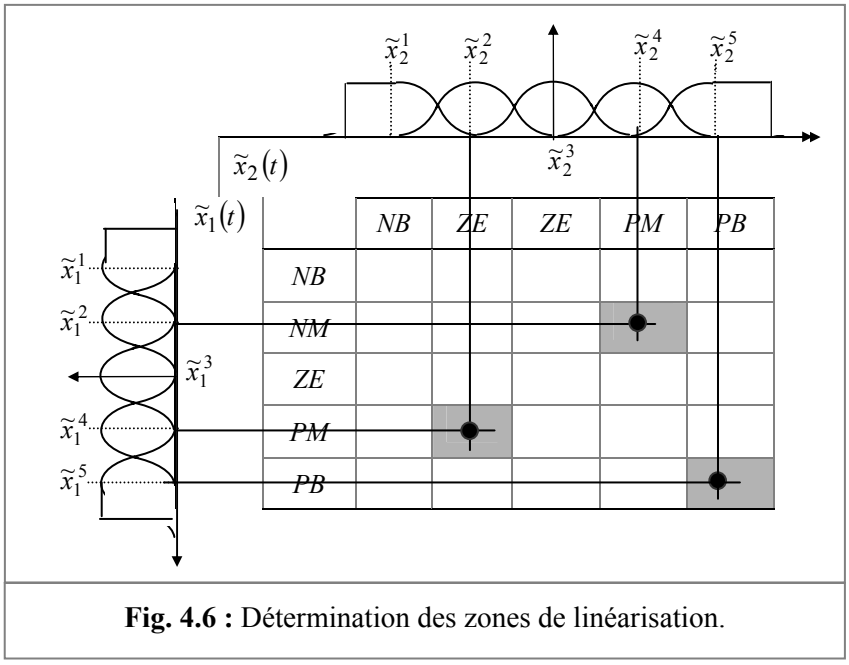


Fig. 4.6 : Détermination des zones de linéarisation.

Les points de coordonnées $\{(P_1, P_2, P_3)\} = \{(\tilde{x}_1^2, \tilde{x}_2^4), (\tilde{x}_1^4, \tilde{x}_2^2), (\tilde{x}_1^5, \tilde{x}_2^5)\}$, dans l'exemple de la figure 4.6, représentent les états autour duquel la linéarisation du système sera établie. La méthode proposée permet d'établir la synthèse d'une loi de commande et de donner des conditions suffisantes de stabilité du modèle en BF. Le nombre minimal de règles (resp. le nombre des modèles locaux) est l'avantage de l'algorithme développé.

L'algorithme de synthèse de la loi de commande par le concept de multimodèle, en intégrant une couche de l'exploration génétique, est illustré par l'organigramme de la figure 4.7.

L'AGM d'optimisation présenté dans l'étape 1 de l'algorithme a le même principe que celui développé dans le chapitre 3 (Section 3.4.2). La différence réside dans la forme (resp. la taille) du chromosome et au niveau du traitement des contraintes conceptuelles. La partie des poids des règles et les paramètres de TS dans la structure du chromosome de la figure 3.13 est remplacée par le vecteur (resp. matrice) des gains de contre réaction de la loi de commande floue PDC_1. La loi de commande par retour d'état (non linéaire) du type PDC_1 a été appliquée avec succès pour le contrôle des systèmes chaotiques. L'oscillateur de Duffing, de Van der Pol et le circuit de Chua sont des exemples d'application de la loi de commande (4.15). Pour le contrôle de circuit de Chua (système MIMO), l'approche hiérarchique est exploitée pour simplifier la synthèse de la loi de commande. Les résultats obtenus sont satisfaisants [93].

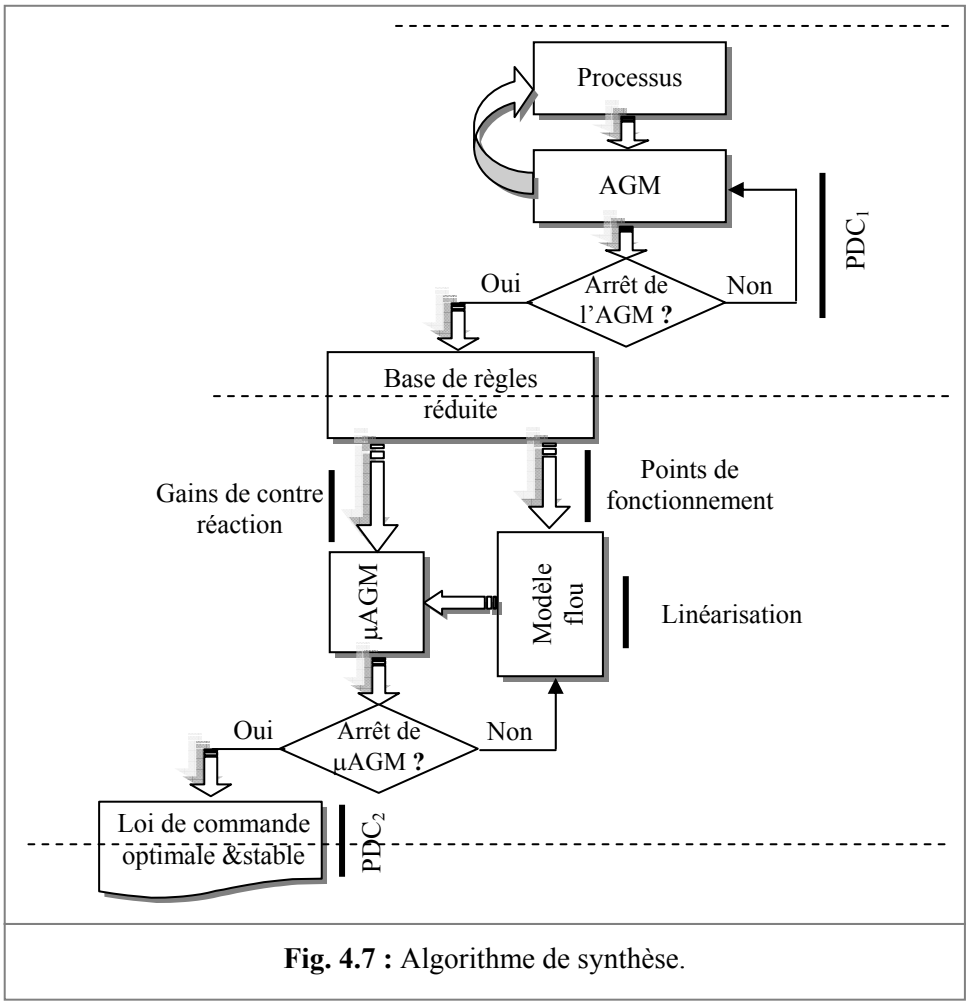


Fig. 4.7 : Algorithme de synthèse.

Dans cette partie de l'algorithme, nous avons utilité le concept du *but programmé 'goal programming*', appelé aussi vecteur à optimisation objective '*target vector optimisation'*. Cette méthode et d'autres sont étudiées en détails par Alain Berro [339].

Le livre de Yann Collette est aussi une base importante des méthodes d'optimisation [340]. Dans cette approche le concepteur fixe au préalable le but Obj_i à atteindre pour chaque objectif. Ces valeurs sont, ensuite, ajoutées au problème d'optimisation comme des contraintes supplémentaires.

Le critère des performances à minimiser, adapté au cas des systèmes multivariables MIMO, est choisi sous forme :

$$J_{E_1} = \sum_{k=0}^{\max\_k-1} \tilde{x}^T(k)\cdot Q(k)\cdot \tilde{x}(k) + \sum_{k=0}^{\max\_k-1} u^T(k)\cdot R(k)\cdot u(k) + \frac{\beta_2}{\max\_règle}\cdot \sum_{i=1}^{\max\_règle} \left|\tilde{c}_F^i - Obj_R\right| \tag{4.29}$$

$\tilde{x}(k) = x^d(k) - x(k)$ est l'erreur de poursuite à l'instant k. $Obj_R > 1$ est l'objectif à atteindre formulé en terme de la taille voulue de la base de règles (le nombre des modèles).

Remarque :

> L'adoption d'un seul modèle linéaire pour modéliser un système non linéaire est très restrictive. En effet, une telle hypothèse écrase les non linéarités du système et

néglige les informations provenant des conditions opératoires. C'est cette réflexion qui a fait naître l'idée de modélisation d'une base de modèle [341].

Les contraintes 1 et 2 exprimées dans (3.35) restent valables dans cette partie de l'algorithme. Les matrices de pondération $Q(k)$ et $R(k)$ sont choisies de sorte que les objectifs cités au chapitre 3 seront respectés.

$$\begin{aligned} Q(k) &= \beta_1(k)\cdot \mathrm{I}_{n\times n} \\ R(k) &= (1-(\beta_1(k)+\beta_2(k)))\cdot \mathrm{I}_{m\times m} \end{aligned} \tag{4.30}$$

$\mathrm{I}_{n\times n}$ est la matrice identité de dimension $n\times n$. Le critère d'arrêt de l'AGM est défini par un nombre bien déterminé de générations. Le chromosome ayant l'aptitude la plus élevée dans la dernière génération de l'AGM représente la base (le repère) de développement des étapes restantes de l'algorithme de synthèse. Le nombre des modèles coïncide avec le nombre des règles obtenu à la fin de l'exécution de l'AGM.

Le modèle flou du processus est conçu par linéarisation du processus autour des points de fonctionnement les plus significatifs (x_i, u_i) par la procédure (4.2). Ces points sont déterminés par les centres des fonctions d'appartenance résultantes dans la partie prémisses des règles floues.

La synthèse de la loi de commande floue par l'analyse de la stabilité s'effectue par un μAGM développé pour réaliser cette tâche. La partie des gains de contre réaction dans les chromosomes de la population de la dernière génération représente la phase d'initialisation d'une partie des chromosomes de la population initiale du μAGM. La partie restante des chromosomes représente les éléments de la matrice P de l'expression (4.17).

Le nouveau problème peut être énoncé de la façon suivante : Etant donnée le modèle flou du système (A_i, B_i), trouver le vecteur (resp. matrice) des gains de contre réaction K_i de la loi de commande PDC_2 et la matrice $P>0$ définissant la fonction de Lyapunov, satisfaisant le critère des performances désirées. Ce dernier est un compromis entre la précision et la stabilité au sens énergétique de Lyapunov. Le critère à minimiser, dans ce cas, sera donné par :

$$J_{E_2} = \sum_{k=0}^{\max\_k-1} \tilde{x}^T(k)\cdot Q(k)\cdot \tilde{x}(k) + \sum_{k=0}^{\max\_k-1} u^T(k)\cdot R(k)\cdot u(k)$$

Sous les contraintes :

$$\begin{cases} & P>0 & \\ (C_{ii}): & (A_i - B_i\cdot K_i)^T\cdot P + P\cdot(A_i - B_i\cdot K_i)<0 & (i=1,...,N) \\ (C_{ij-ji}): & G_{ij}^T\cdot P + P\cdot G_{ij}<0 & (1\le i<j\le N) \\ & G_{ij} = \dfrac{A_i - B_i\cdot K_j + A_j - B_j\cdot K_i}{2} & \end{cases} \tag{4.31}$$

Les matrices de pondération $Q(k)$ et $R(k)$ sont données par (4.30) avec $\beta_2 = 0$. Dans cette partie de la thèse, l'analyse et le traitement des contraintes de la stabilité dans le μAGM sont effectués d'une manière différente que celle décrite dans le chapitre précédent. Une alternative consiste à imposer les contraintes de stabilité comme critère d'arrêt de

l'algorithme de synthèse. Le processus d'exécution peut être résumé par l'organigramme de la figure 4.8. C_{ij} est la condition sur les modèles i et j et ξ est un seuil prédéfini.

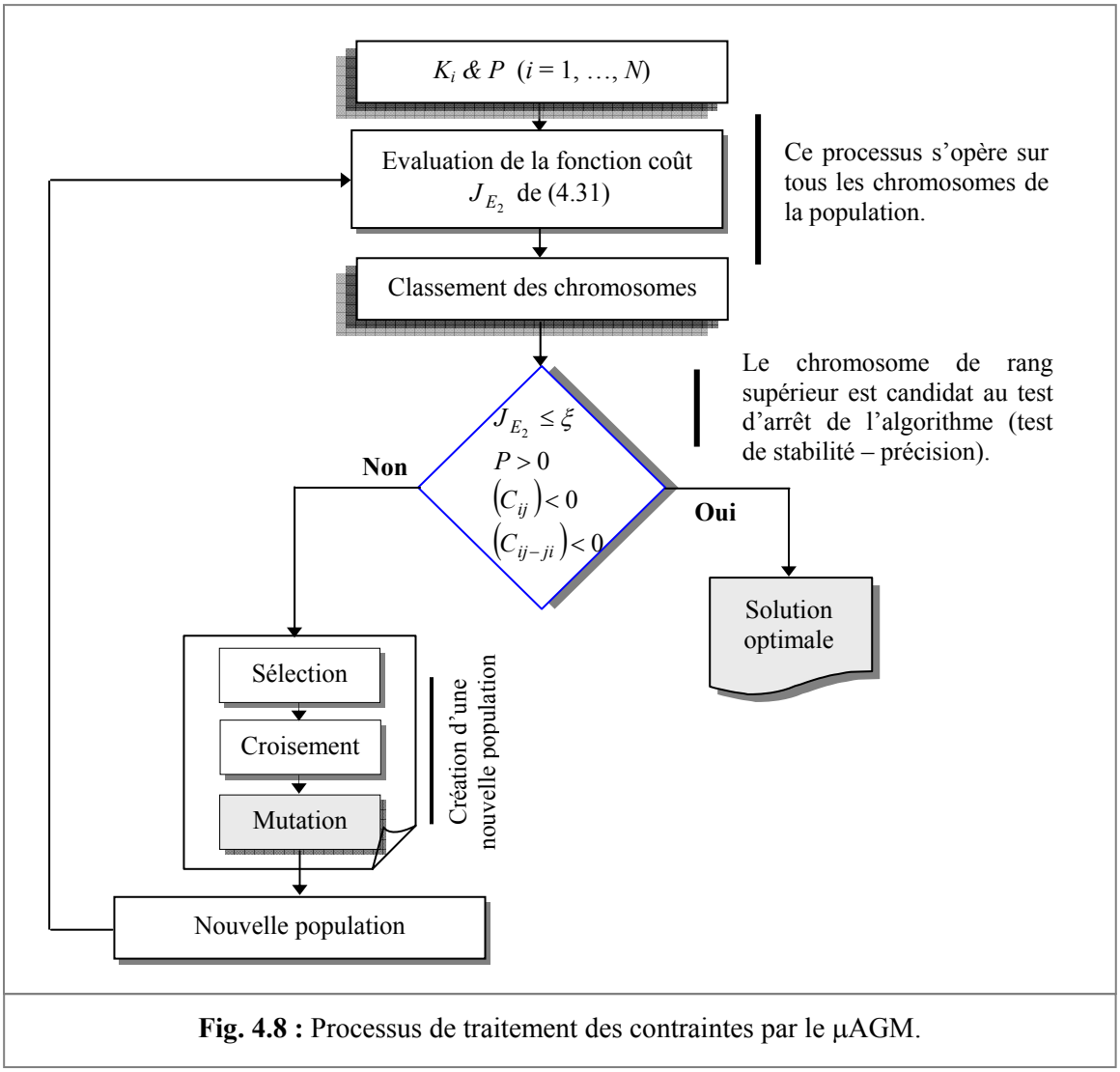


Fig. 4.8 : Processus de traitement des contraintes par le μAGM.

Pour illustrer la validité de l'approche proposée pour la modélisation floue de systèmes non linéaires, en particulier les systèmes MIMO, la suite du chapitre est dédiée à l'application de l'algorithme développé à la commande d'un bioprocédé de traitement des eaux usées dans l'industrie du papier.

4.5. Simulation

Pour illustrer les performances de l'algorithme proposé pour la synthèse d'une loi de commande floue tout en assurant la stabilité en BF, nous considérons la commande en poursuite d'un bioprocédé de traitement des eaux usées dans l'industrie du papier.

4.5.1. Commande d'un bioprocédé de traitement des eaux usées

4.5.1.1. Généralités

En raison de la complexité inhérente des mécanismes de traitement, l'application des TIA en collaboration avec les méthodes conventionnelles dans le domaine de l'engineering des bioprocédés ont été proposées comme alternative de traitement et d'analyse. L'objectif

principal est de réduire au minimum les impacts environnementaux et économiques simultanément [342]. Cependant, la commande des bioréacteurs est un problème sensible puisque la majorité des modèles biologiques disponibles sont, uniquement, déterminés ou caractérisés par des approximations.

En effet, les systèmes biologiques sont des procédés fortement variables, non linéaires et difficiles à modéliser de sorte qu'aucune loi biologique fiable ne soit, actuellement, disponible [343]. Un aspect important du contrôle des bioprocédés est de leur imposer un fonctionnement en temps réel stable, peu sensible aux différentes perturbations, proche d'un certain état désiré susceptible de correspondre à un état de fonctionnement optimal.

Le livre «*Automatique des Bioprocédés* » s'articule autour de la description du mode de fonctionnement des bioprocédés, de la modélisation mathématique et s'intéresse au développement des concepts de base de la commande automatique des bioprocédés. Les méthodes de commandes avancées appliquées aux bioprocédés sont aussi abordées [344].

G. Boari et al. ont présenté un panorama des technologies et des procédés qui sont utilisés dans les traitements des eaux et des eaux résiduaires pour l'enlèvement des différentes composantes caractéristiques de la pollution : Solides, carbone organique, azote, phosphore, sels inorganiques et métaux, pathogènes [345].

Les bioprocédés sont caractérisés par des modèles qui intègrent des non linéarités, des couplages entre les variables d'entrées, des variations paramétriques et des entrées non mesurables [341]. La modélisation et la commande de tels procédés sont donc une tâche très délicate.

La principale question abordée dans cette application sera l'amélioration et l'optimisation des performances du procédé à l'aide d'outils de modélisation multimodèle flous et d'analyse de la stabilité du système en BF. Le schéma synoptique du procédé est donné par la figure 4.9.

L'expérience a montrée que l'application des contrôleurs conventionnels linéaires de type PID à ce type de systèmes complexes est limitée [346] - [348]. Il est, par conséquent, nécessaire de s'intéresser à des techniques de commande plus sophistiquées qui tiennent comptes des caractéristiques des bioprocédés [342] & [349].

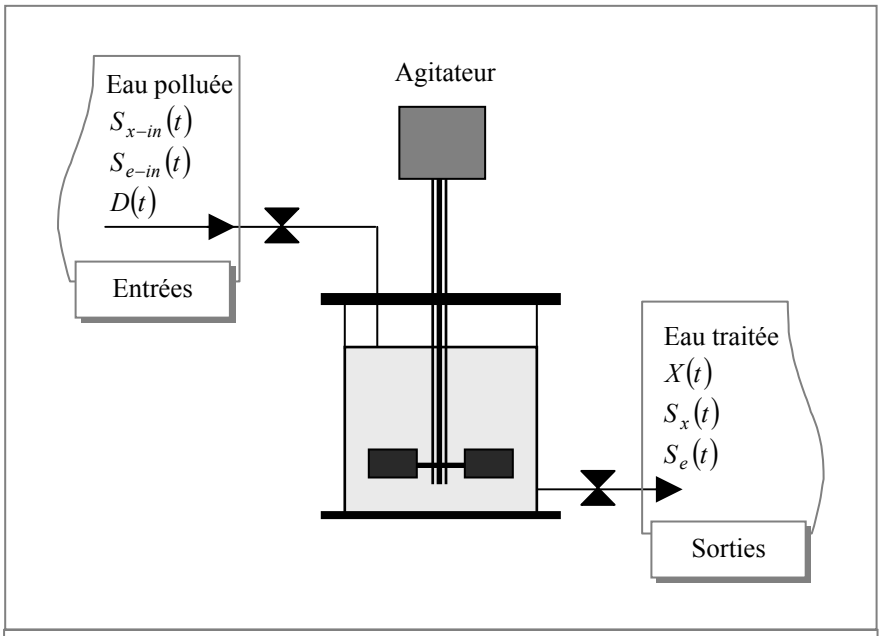


Fig. 4.9 : Schéma synoptique du bioprocédé.

4.5.1.2. Modèle du bioprocédé

Généralement, l'entrée du système est définie en fonction des objectifs de contrôle monovariable ou multivariable que l'on désire se fixer. Dans le cas de multivariable, les grandeurs de commande correspondent au taux de dilution $D(t)$ du réacteur et à la concentration en substrats xénobiotique et énergétique du milieu d'alimentation $S_{x-in}(t)$ et $S_{e-in}(t)$. En pratique (Fig. 4.10), il suffit de dissocier l'alimentation classique en 2 alimentations : L'une apportant de l'eau et l'autre le milieu contenant les 2 sources de carbone aux concentrations maximales $S_{x-in}^{\max}(t)$ et $S_{e-in}^{\max}(t)$. La sortie est un débit d'eau traitée. Les relations qui lient $D(t)$, $S_{x-in}(t)$ et $S_{e-in}(t)$ aux variables de commande auxiliaires $u_1(t)$ et $u_2(t)$ sont données par :

$$\begin{aligned} D(t) &= u_1(t) + u_2(t) \\ S_{x-in}(t) &= \frac{u_2(t)}{u_1(t)+u_2(t)} \cdot S_{x-in}^{\max}(t) \\ S_{e-in}(t) &= \frac{u_2(t)}{u_1(t)+u_2(t)} \cdot S_{e-in}^{\max}(t) \end{aligned} \tag{4.32}$$

$u_1(t)$ et $u_2(t)$ correspondent, respectivement, aux taux de dilution en eau et en substrats.

Le système comporte 2 variables de commande en entrées $u(t) = (u_1(t), u_2(t))$ et 3 variables de sorties $(x_1(t), x_2(t), x_3(t)) = (X(t), S_x(t), S_e(t))$. $u_1(t)$ est le taux de dilution en eau et $u_2(t)$ est le taux de dilution en substrat. $X(t), S_x(t)$ et $S_e(t)$ sont les concentrations de biomasse, de substrats xénobiotique et énergétique, respectivement.

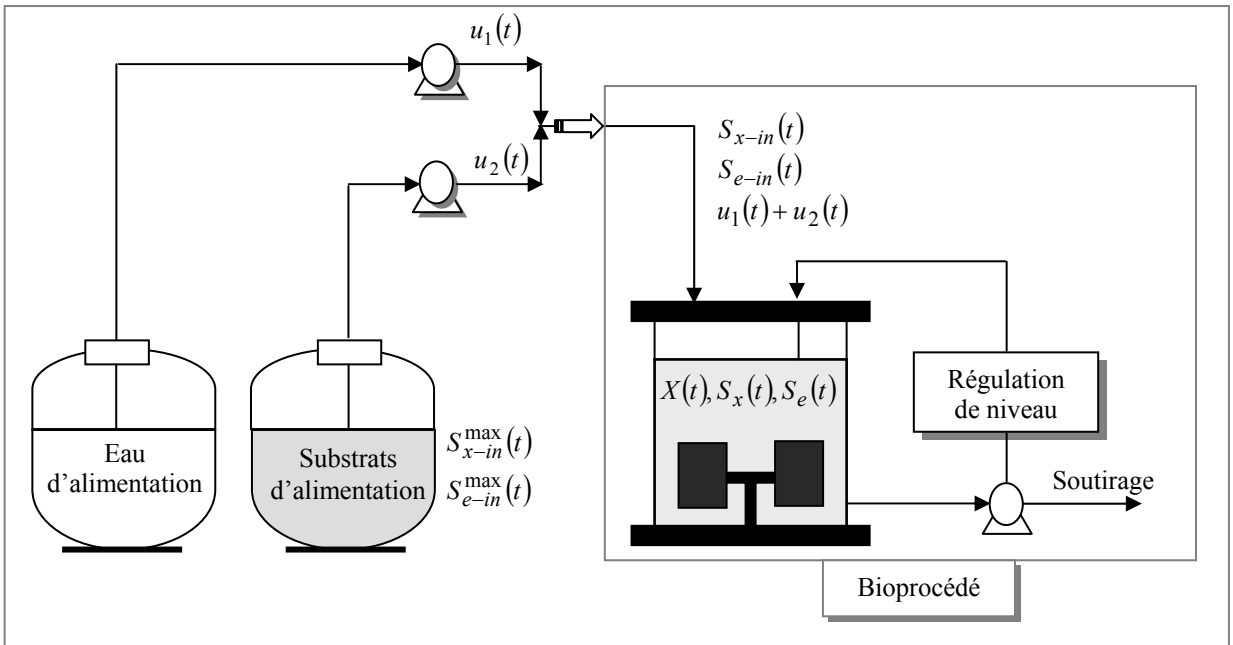


Fig. 4.10 : Structure de bioprocédé de traitement (cas multivariable).

La première étape dans la synthèse de la loi de commande est d'établir un modèle mathématique du processus à commander. La dynamique des procédés industriels est très complexe et, souvent, mal connue. Les dynamiques des processus biologiques (resp.

processus biotechnologiques) sont, habituellement, données en termes d'équations différentielles non linéaires établies à partir des équations bilans matières pour chaque élément macroscopique (principaux composants) de la réaction biologique [342] & [350] - [353].

L'ensemble des équations différentielles décrivant le comportement dynamique du bioprocédé est donné par [342] & [350] - [353] :

$$\begin{aligned}
\dot{x}_1(t) &= \left(\mu_{Sx}^{\max} \frac{x_2(t)}{K_{Sx} + x_2(t) + a_{Se} x_3(t)} + \mu_{Se}^{\max} \frac{x_3(t)}{K_{Se} + x_3(t) + a_{Sx} x_2(t)} \right) x_1(t) - (u_1(t) + u_2(t)) x_1(t) - b x_1(t) \\
\dot{x}_2(t) &= -\frac{\mu_{Sx}^{\max}}{Y_{X/S_x}} \frac{x_2(t)}{K_{Sx} + x_2(t) + a_{Se} x_3(t)} x_1(t) - u_1(t) x_2(t) + \left(S_{x-in}^{\max} - S_{x-in}(t) \right) u_2(t) + d_{ist}(t) \\
\dot{x}_3(t) &= -\frac{\mu_{Se}^{\max}}{Y_{X/S_e}} \frac{x_3(t)}{K_{Se} + x_3(t) + a_{Sx} x_2(t)} x_1(t) - u_1(t) x_3(t) + \left(S_{e-in}^{\max} - S_{e-in}(t) \right) u_2(t)
\end{aligned} \tag{4.33}$$

Les différents paramètres et variables intervenant dans le modèle d'équation (4.33), décrivant le comportement dynamique du bioprocédé, sont donnés dans le tableau 4.2 [353].

Les valeurs numériques des différents paramètres du bioprocédé sont données dans le tableau 4.3.

Tab. 4.2 : Paramètre du bioprocédé.

Paramètre	Description
$X(t) = x_1(t)\,[g/L]$	Concentration de biomasse (micro-organismes)
$S_x(t) = x_2(t)\,[g/L]$	Concentration de substrat xénobiotique
$S_e(t) = x_3(t)\,[g/L]$	Concentration de substrat énergétique
$\mu_{Sx}^{\max}$	Taux spécifique de croissance maximal dû au substrat xénobiotique
$\mu_{Se}^{\max}$	Taux spécifique de croissance maximal dû au substrat énergétique
Y_{X/S_x}	Rendement de conversion du substrat xénobiotique en biomasse
Y_{X/S_e}	Rendement de conversion du substrat énergétique en biomasse
K_{Sx}, K_{Se}	Constantes de Michaëlis-Menten pour les substrats xénobiotique & énergétique
a_{Sx}	Constante d'inhibition du substrat énergétique sur le substrat xénobiotique
a_{Se}	Constante d'inhibition du substrat xénobiotique sur le substrat énergétique
b	Terme de mortalité bactérienne dû à la respiration endogène
$u_1(t)$	Taux de dilution d'eau d'alimentation (eau propre)
$u_2(t)$	Taux de dilution de substrats d'alimentation (eau résiduaire)
$S_{x-in}(t)$	Concentration en substrat xénobiotique du milieu d'alimentation
$S_{e-in}(t)$	Concentration en substrat énergétique du milieu d'alimentation
$S_{x-in}^{\max}(t), S_{e-in}^{\max}(t)$	Concentrations maximales en substrats dans le milieu d'alimentation

Tab. 4.3 : Valeurs numériques des paramètres du bioprocédé.

Paramètre	Valeur
Y_{X/S_x}	0.7
Y_{X/S_e}	0.2
$\mu_{Se}^{\max}$	0.2 $[h^{-1}]$
$\mu_{Se}^{\max}$	0.2 $[h^{-1}]$
b	0.005 $[h^{-1}]$
K_{Sx}	1.5 $[gL^{-1}]$
K_{Se}	1 $[gL^{-1}]$
a_{Sx}	0.1
a_{Se}	10
$S_{x-in}^{\max}$	6 $[gL^{-1}]$
$S_{e-in}^{\max}$	6 $[gL^{-1}]$

Les limitations physiques des actionneurs sont les suivantes :

$$0.0001 \le u_i(t) \le 0.04\,[h^{-1}] \qquad i = 1, 2. \tag{4.34}$$

L'objectif de contrôle est de permettre aux variables $X(t)$ et $S_x(t)$ de suivre, fidèlement, des trajectoires prédéterminées $X^d(t)$ et $S_x^d(t)$, respectivement, par action sur le taux de dilution $u(t) = (u_1(t), u_2(t))$. Le modèle de référence est donné par :

$$x_1^d(k) = \begin{cases} 0.6\,[g/L] & k < 80\,[h^{-1}] \\ 0.7\,[g/L] & 80 \le k < 220\,[h^{-1}] \\ 0.6\,[g/L] & 220 \le k \le 300\,[h^{-1}] \end{cases} \tag{4.35}$$

$$x_2^d(k) = \begin{cases} 0.3\,[\mathrm{g/L}] & k < 180\,[h^{-1}] \\ 0.2\,[\mathrm{g/L}] & 180 \le k < 300\,[h^{-1}] \end{cases} \tag{4.36}$$

Pour avoir une idée sur l'évolution du système, c'est-à-dire, la plage de variation des variables d'état décrivant son comportemet, nous avons simulé le modèle (4.33) afin de tenir compte des situations *réelles* de son comprtement.

4.5.1.3. Elaboration d'un jeu de données entrée-sortie

Pour générer le jeu de données, nous supposons que toutes les variables d'état sont accessibles à la mesure. Afin de disposer d'un jeu de données approprié pour l'identification du bioprocédé, il est préférable d'exciter le système dans toutes la gamme des variables considérées tant en amplitude qu'en fréquence.

On applique au bioprocédé des excitations aléatoires distribuées ayant les bornes spécifiques au processus $0.0001 \le u_i(t) \le 0.04\,[h^{-1}]$, $(i = 1, 2)$. La période d'échantillonnage est choisie égale à 1 heure, avec un horizon de temps de $1000\,[h]$. Les signaux d'excitations générés sont représentés par les figures 4.11 et 4.12, respectivement.

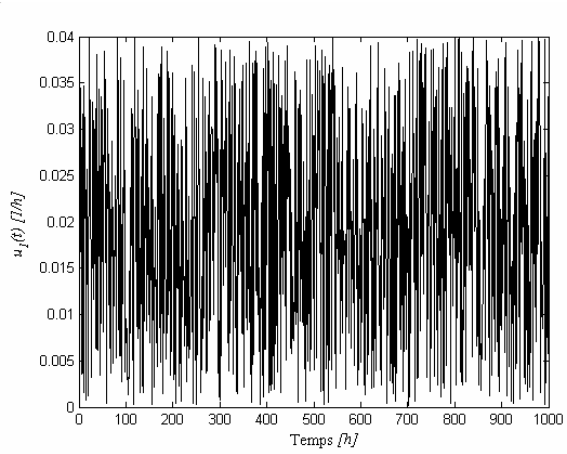


Fig. 4.11 : Signal $u_1(t)$ pour l'identification.

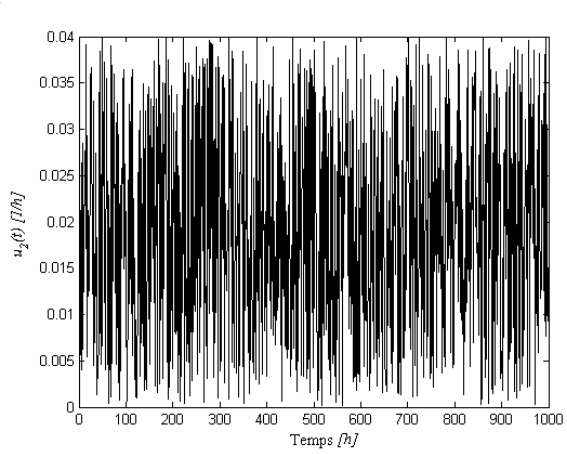


Fig. 4.12 : Signal $u_2(t)$ pour l'identification.

La répartition du jeu de données peut aussi être visualisée dans le plan de phase $(u_1(t), u_2(t))$ de la figure 4.13.

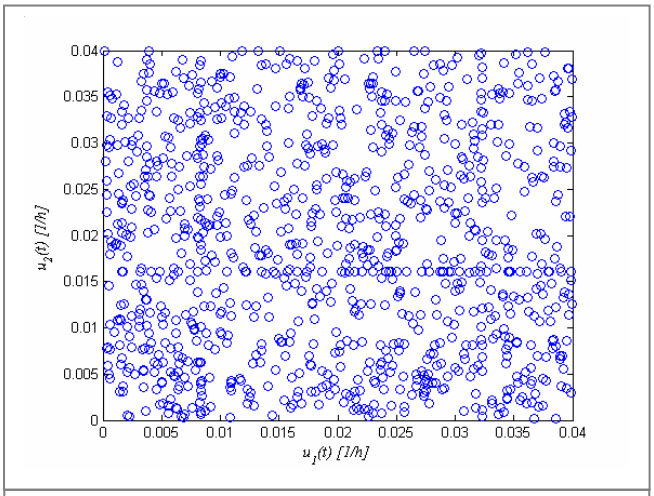


Fig. 4.13 : Répartition des données d'entrées.

On peut noter que les signaux d'entrées sont, relativement, bien distribués dans la plage des variations autorisées des grandeurs d'entrées. Le jeu des données d'identification a été obtenu avec des conditions initiales $(X(0), S_x(0), S_e(o))^T = (0.5, 0.45, 0.2)^T\,[gL^{-1}]$. Ces conditions sont prises en considération dans la phase du synthèse de la loi de commande du bioprocédé [353].

L'évolution des variables d'état du bioprocédé est visualisé sur les figures 4.14, 4.15 et 4.16, respectivement.

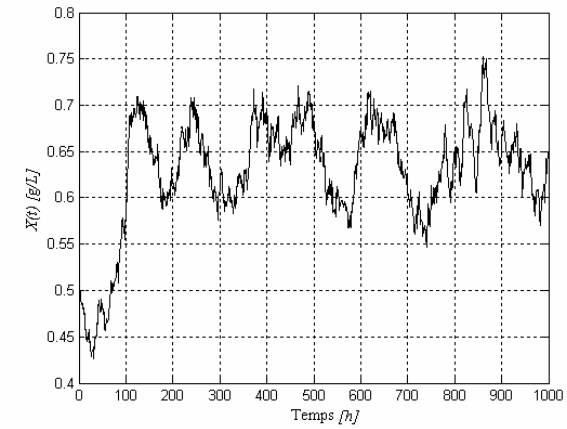


Fig. 4.14 : Evolution de la variable $X(t)$.

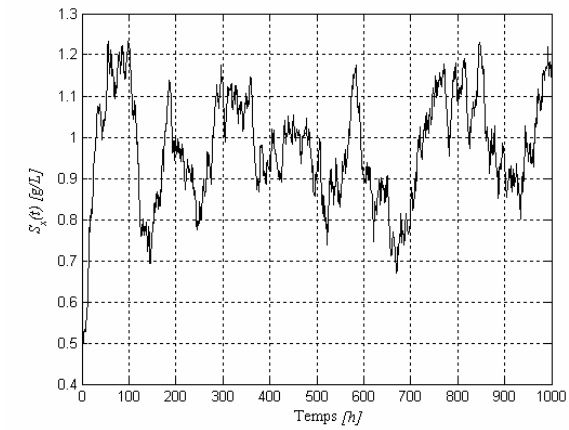


Fig. 4.15 : Evolution de la variable $S_x(t)$.

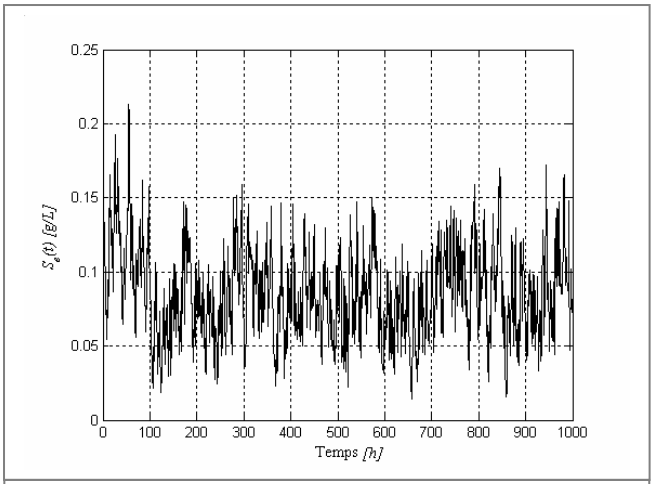


Fig. 4.16 : Evolution de la variable $S_e(t)$.

La figure 4.17 montre la répartition correspondante des données dans l'espace des sorties des variables à commander, des concentrations de biomasse, du substrat xénobiotique et énergétique.

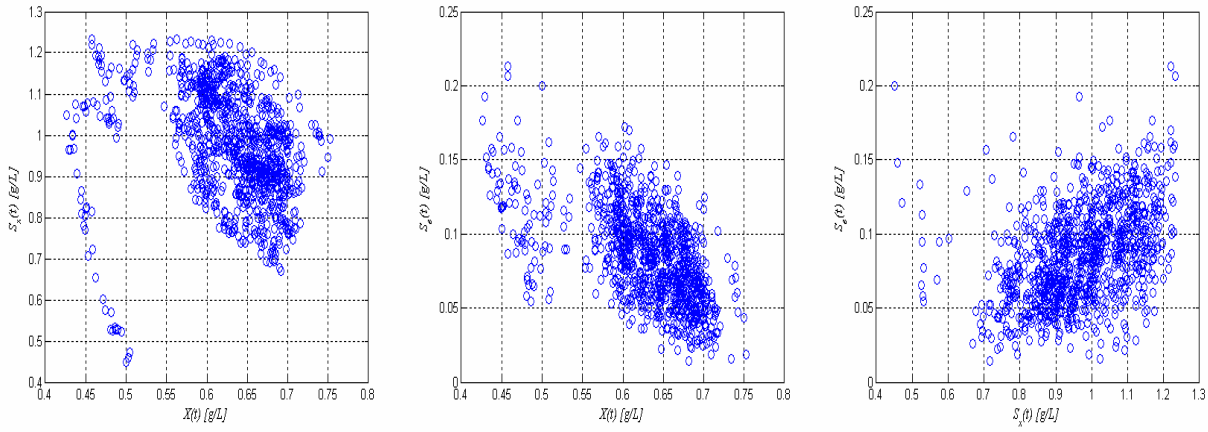


Fig. 4.17 : Répartition des données dans l'espace des sorties.

Remarques :

Si le bioprocédé évolue dans des conditions normales, telles que celles décrites dans (4.35) et (4.36), la concentration du substrat énergétique $S_e(t)$ est résiduelle du fait de sa haute biodégradabilité. Ceci explique, pourquoi on ne tient pas compte de la régulation du substrat énergétique au niveau de la loi de commande [353]. Cette information nous permet de simplifier la représentation et la synthèse de la loi de commande. Cherif Ben Youssef et al. ont proposé une approche d'estimation non linéaire des paramètres d'un bioprocédé. Une loi de commande adaptative avec modèle de référence a été développée. Par contre, l'analyse et la synthèse ont été effectuées sans prise en compte de la variable $S_e(t)$ (Fig. 4.18) [354].

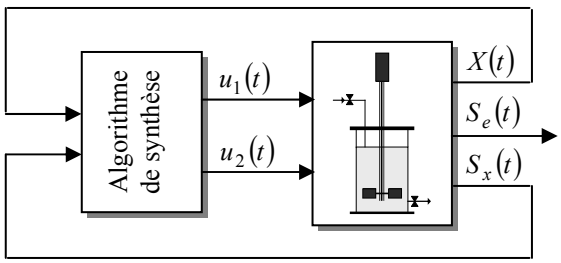


Fig. 4.18 : Variables de modélisation.

La simulation du comportement du bioprocédé nous permet de définir la plage de variation des variables d'état. Par la suite, l'univers de discours de ces variables est défini en se basant sur les résultats obtenus dans la simulation précédente. Les valeurs spécifiques de la loi de commande ainsi que les contraintes imposées sont données par le tableau 4.4.

Tab. 4.4 : Valeurs spécifiques de l'algorithme de synthèse.

Caractéristique	Valeur
Taille de la population	10
Nombre de génération	variable
Univers de discours	L_{x_1} & $L_{x_2} \in]0,1.5]$
Gains de PDC	$K \in [-2,2]$
max_k	300
Max _ règle	25

Les fonctions d'appartenance sont choisies du type gaussien symétrique équidistant avec un chevauchement de 50 %. L'optimisation des bases de ces fonctions est similaire à celle développée au chapitre 3 (Section 3.4.2.1). Le potentiel offert par cette représentation est la réduction de la taille des chromosomes dans l'AGM.

4.5.1.4. Stratégie de modélisation multimodèle

L'évolution de la fonction objective ainsi que le nombre des règles actives au cours de l'exécution de l'AGM sont illustrées par les figures 4.19 et 4.20, respectivement. On

remarque que l'objectif auxiliaire $Obj_R = 2$ introduit dans le critère (4.24) est atteint à partir de la génération numéro 97.

On fait relevé les valeurs des zones de fonctionnement au moment où l'objectif Obj_R se stabilise à la valeur désirée, puis on fait appel à l'étape 2 & 3 de l'algorithme de synthèse. Le multimodèle voulu a une configuration minimale égale à 2 modèles locaux. La figure 4.21 représente la base de règles floues constituant le multimodèle.

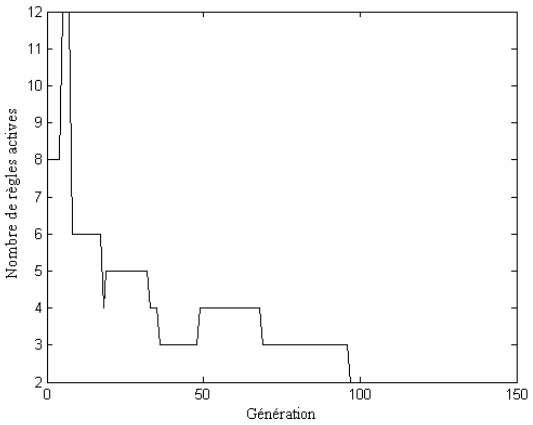


Fig. 4.19 : Evolution de la base de règles.

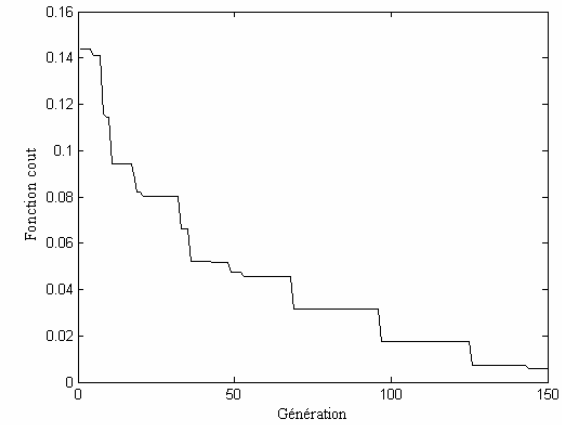


Fig. 4.20 : Evolution de la fonction coût.

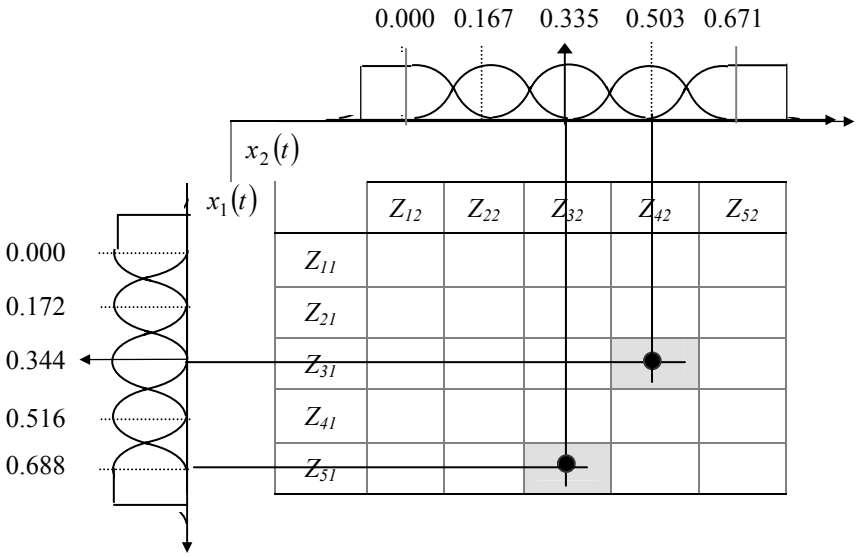


Fig. 4.21 : Base des règles actives.

Les zones de fonctionnement définies par l'AGM sont spécifiées par les fonctions d'appartenance Z_{31} et Z_{51} pour la variable $X(t)$ et par Z_{32} et Z_{52} pour la variable $S_x(t)$, respectivement. Les résultats obtenus sont illustrés par la figure 4.22.

Par application de l'algorithme PDC_1 ou PPDC (4.14) avec la procédure d'optimisation AGM, on peut avoir des résultats satisfaisants en poursuite d'une trajectoire de référence donnée. Seule, l'analyse de la stabilité reste à confirmer [93] & [330]. Pour que le critère de stabilité soit prouvé, les étapes 2 & 3 de l'algorithme de synthèse sont

nécessaires. Les points de fonctionnement représentant la base de linéarisation sont définis par :

$$P_1 = \left(\underbrace{0.344, 0.503, 0.018}_{x_i \ i=1,2,3}, \underbrace{0.0001, 0.0001}_{u_i \ i=1,2} \right) \& \ P_2 = \left(\underbrace{0.686, 0.335, 0.018}_{x_i \ i=1,2,3}, \underbrace{0.033, 0.04}_{u_i \ i=1,2} \right) \quad (4.37)$$

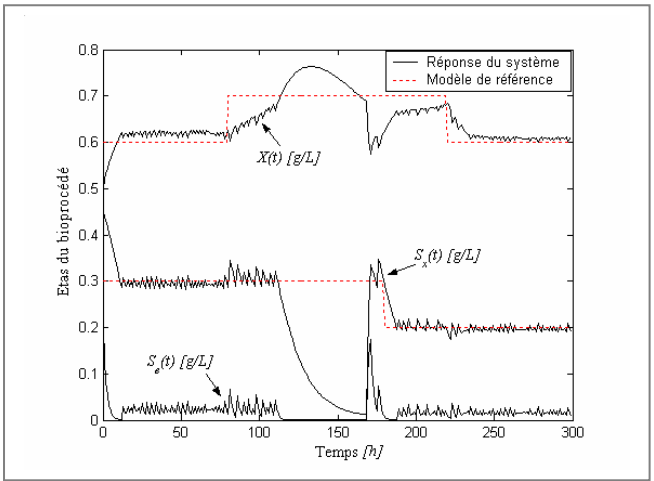


Fig. 4.22 : Evolution des états du bioprocédé.

D'après la figure 4.22, la variable $S_e(t)$ peut être considérée comme constante égale à 0.018 [*g/L*]. Les matrices des gains de contre réaction du PDC_1 sont :

$$K_1 = \begin{bmatrix} 0.1188 & 0.1434 \\ -0.0815 & -2.8177 \end{bmatrix} \ \& \ K_2 = \begin{bmatrix} 0.1995 & 0.8847 \\ 0.1803 & 1.9199 \end{bmatrix} \quad (4.38)$$

Une base de modèles capable de représenter le système non linéaire (4.33) sur des plages de variation du point d'équilibre (zone de fonctionnement) peut être décrite par :

$$\begin{aligned}
a_{11}^i &= \left.\frac{\partial f_1}{\partial x_1}\right|_{\substack{x=x_i \\ u=u_i}} = \left[\mu_{Sx}^{\max} \frac{x_{2i}}{K_{Sx} + x_{2i} + a_{Se} x_{3i}} + \mu_{Se}^{\max} \frac{x_{3i}}{K_{Se} + x_{3i} + a_{Sx} x_{2i}} \right] - (u_{1i} + u_{2i}) - b \\
a_{12}^i &= \left.\frac{\partial f_1}{\partial x_2}\right|_{\substack{x=x_i \\ u=u_i}} = x_{1i} \left[-\mu_{Sx}^{\max} \frac{(K_{Sx} + a_{Se} x_{3i})}{(K_{Sx} + x_{2i} + a_{Se} x_{3i})^2} + \mu_{Sx}^{\max} \frac{a_{Sx} x_{3i}}{(K_{Se} + x_{3i} + a_{Sx} x_{2i})^2} \right] \\
a_{21}^i &= \left.\frac{\partial f_2}{\partial x_1}\right|_{\substack{x=x_i \\ u=u_i}} = -\frac{\mu_{Sx}^{\max}}{Y_{X/S_x}} \frac{x_{2i}}{K_{Sx} + x_{2i} + a_{Se} x_{3i}} \\
a_{22}^i &= \left.\frac{\partial f_2}{\partial x_2}\right|_{\substack{x=x_i \\ u=u_i}} = +\frac{\mu_{Sx}^{\max}}{Y_{X/S_x}} \frac{(K_{Sx} + a_{Se} x_{3i})}{(K_{Sx} + x_{2i} + a_{Se} x_{3i})^2} x_{1i} - u_{1i}
\end{aligned} \quad (4.39)$$

$$
\begin{aligned}
b_{11}^{i} &= \left.\frac{\partial f_1}{\partial u_1}\right|_{\substack{x=x_i \\ u=u_i}} = -x_{1i} \\
b_{12}^{i} &= \left.\frac{\partial f_1}{\partial u_2}\right|_{\substack{x=x_i \\ u=u_i}} = -x_{1i} \\
b_{21}^{i} &= \left.\frac{\partial f_2}{\partial u_1}\right|_{\substack{x=x_i \\ u=u_i}} = -x_{2i} - S_{x-in}^{\max}\left(\frac{u_{2i}}{u_{1i}+u_{2i}}\right)^2 \\
b_{22}^{i} &= \left.\frac{\partial f_2}{\partial u_2}\right|_{\substack{x=x_i \\ u=u_i}} = S_{x-in}^{\max}\left[1-\left(\frac{u_{2i}}{u_{1i}+u_{2i}}\right)^2\right]
\end{aligned}
\tag{4.40}
$$

Les matrices du multimodèle sont alors :

$$
\begin{aligned}
A_1 &= \begin{bmatrix} 0.021212 & -0.012019 \\ -0.032917 & 0.017225 \end{bmatrix}, \; A_2 = \begin{bmatrix} -0.058709 & -0.028161 \\ -0.023750 & 0.006792 \end{bmatrix}, \\
B_1 &= \begin{bmatrix} -0.344000 & -0.344000 \\ -2.003000 & 4.500000 \end{bmatrix}, \; B_2 = \begin{bmatrix} -0.686000 & -0.686000 \\ -2.099651 & 4.235349 \end{bmatrix}
\end{aligned}
\tag{4.41}
$$

D'après les valeurs obtenues, on remarque que les paires (A_i, B_i), $(i = 1, 2)$ sont commandables. L'évolution du critère de performances du μAGM (4.31) est représentée par la figure 4.23.

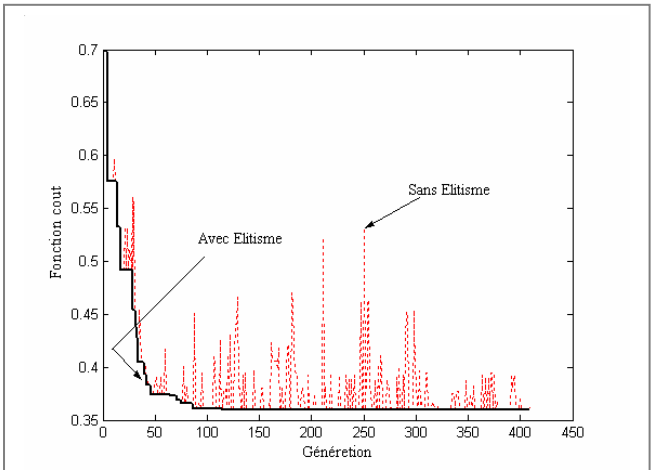


Fig. 4.23 : Evolution de la fonction coût.

Les matrices des gains de contre réaction et la matrice P obtenues à la génération 409, où toutes les conditions de stabilité sont bien vérifiées, sont données par :

$$
P = \begin{bmatrix} 4.200 & -0.068 \\ -0.068 & 0.024 \end{bmatrix}, \; K_1 = \begin{bmatrix} -0.325 & 1.023 \\ -0.129 & 0.660 \end{bmatrix} \; \& \; K_2 = \begin{bmatrix} 0.392 & 0.952 \\ -0.599 & -0.206 \end{bmatrix}
\tag{4.42}
$$

On peut, facilement, démontrer que les conditions de stabilité sont satisfaisantes :

$$
\begin{aligned}
&P > 0 \\
&\left(A_1 - B_1 \cdot K_1\right)^T \cdot P + P \cdot \left(A_1 - B_1 \cdot K_1\right) < 0 \\
&\left(A_2 - B_2 \cdot K_2\right)^T \cdot P + P \cdot \left(A_2 - B_2 \cdot K_2\right) < 0 \\
&G_{12}^T \cdot P + P \cdot G_{12} < 0 \\
&G_{12} = \frac{A_1 - B_1 \cdot K_2 + A_2 - B_2 \cdot K_1}{2}
\end{aligned}
\tag{4.43}
$$

La loi de commande PDC_2 est aussi donnée par :

$$u = -w_1 \cdot K_1 \cdot \left(x - x^d\right) - w_2 \cdot K_2 \cdot \left(x - x^d\right) \tag{4.44}$$

w_1 et w_2 sont les valeurs des degrés d'appartenance pour les règles (modèles locaux) 1 et 2, respectivement. La loi de commande PDC_2 (non linéaire) garantie la stabilité de la boucle de contrôle flou (modèle flou + contrôleur PDC flou).

Le schéma de la figure 4.24 montre le comportement des états commandés $X(t)$ et $S_x(t)$ et de leur trajectoire de référence désirée correspondante données par les expressions (4.35) et (4.36), respectivement. Au points de changement de consignes (des transitions) $(80\,[h], 180\,[h], 220\,[h])$, nous pouvons observer que l'effet de l'accouplement entre les 2 variables commandées a été atténué et que les trajectoires de références ont été atteintes après un temps de transition court. Il s'agit d'une bonne compensation des interactions entre les variables d'état. Les résultats de simulation montrent les bonnes performances de l'algorithme développé et confirment l'efficacité de la loi de commande en poursuite des trajectoires désirées qui sont, théoriquement, possibles avec un taux de dilution $D(t) = (u_1(t), u_2(t))$ non nul. Les signaux de commande sont schématisés par les figures 4.25 et 4.26. Le comportement du modèle non linéaire (4.33) par l'action de commande développée PDC_2 est représentée par les figures 4.27, 4.28 et 4.29, respectivement.

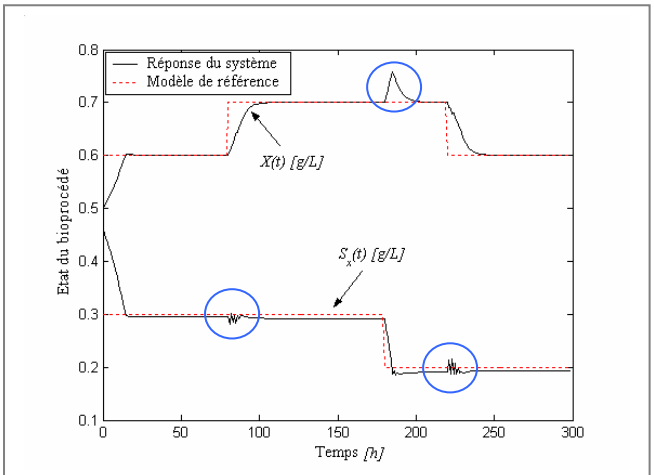


Fig. 4.24 : Evolution des états du bioprocédé.

Sans modifier la loi de contrôle développée, la robustesse de ce dernier est testée en simulant des perturbations sur les variables d'état du bioprocédé. Ces perturbations sont identiques et décrites par le modèle (3.51) où $(p_1, p_2, p_3) = (0.03, 2, 0.005)$. Les résultats

obtenus schématisés par les figures 4.30, 4.31 et 4.32 montrent que le contrôleur arrive à se stabiliser le système autour de la trajectoire désirée. Le comportement de bioprocédé (4.33) dans les conditions précédentes est schématisé par les figures 4.33, 4.34 et 4.35, respectivement. En conclusion, les résultats obtenus sont encourageants et montrent le potentiel de la démarche.

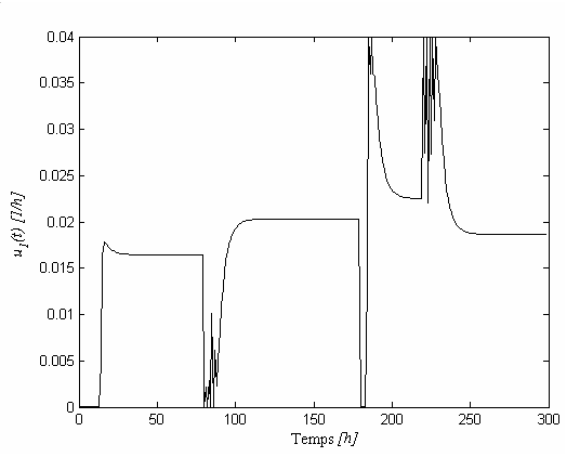


Fig. 4.25 : Signal de commande $u_1(t)$.

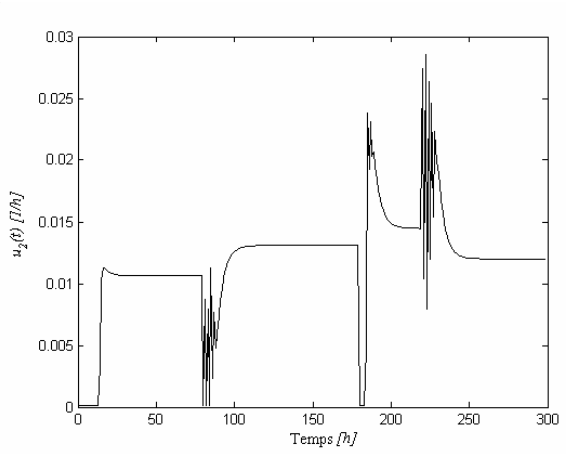


Fig. 4.26 : Signal de commande $u_2(t)$.

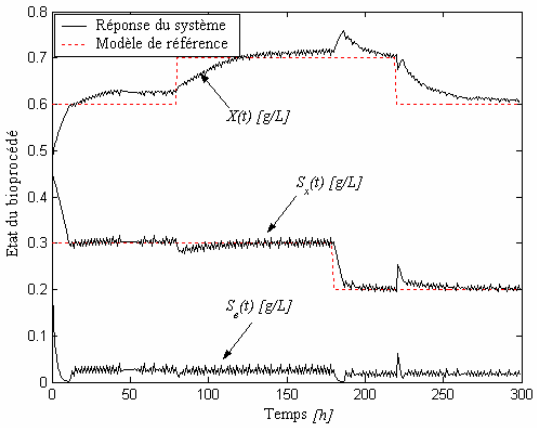


Fig. 4.27 : Evolution du bioprocédé non linéaire.

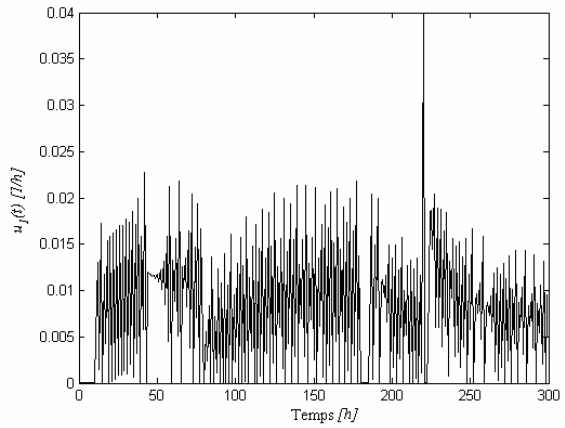


Fig. 4.28 : Signal de commande $u_1(t)$.

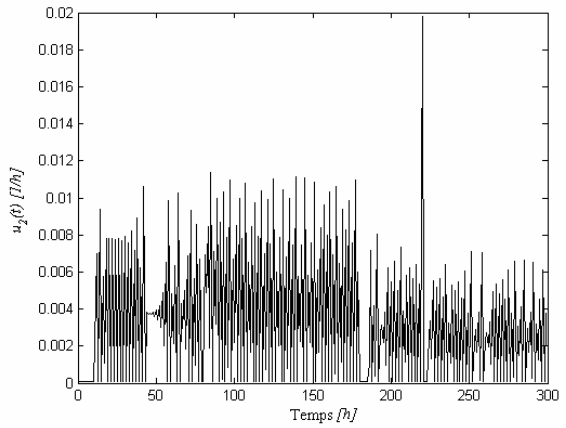


Fig. 4.29 : Signal de commande $u_2(t)$.

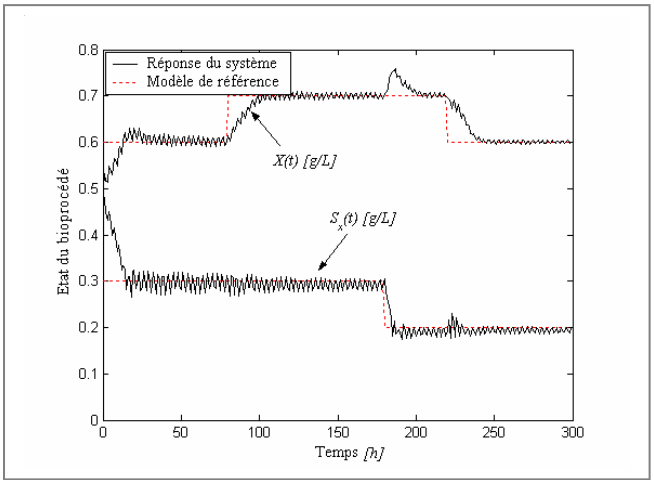


Fig. 4.30 : Evolution des états du bioprocédé.

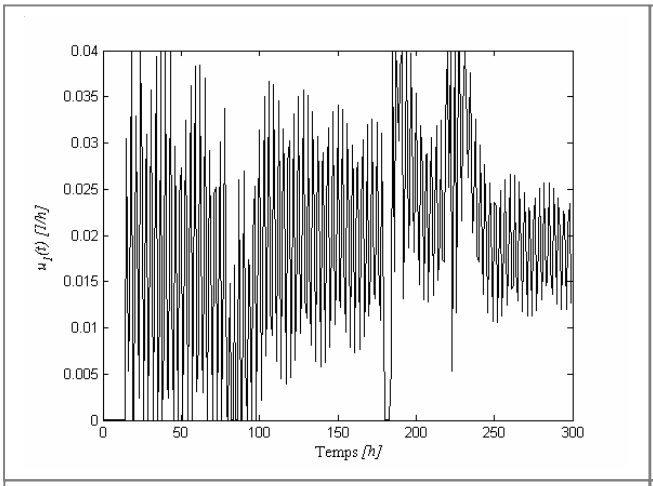


Fig. 4.31 : Signal de commande $u_1(t)$.

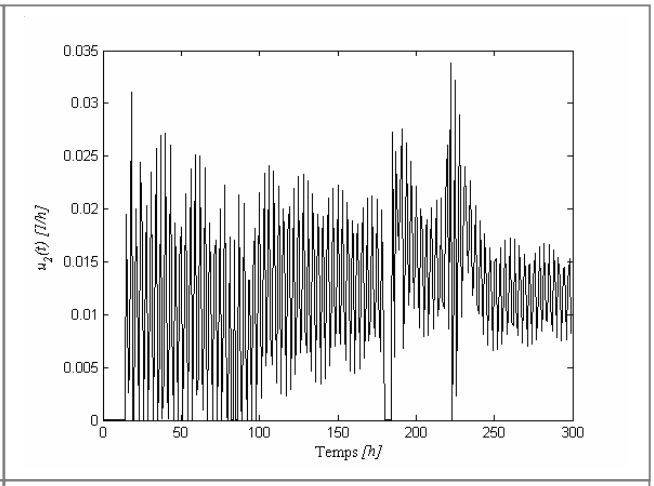


Fig. 4.32 : Signal de commande $u_2(t)$.

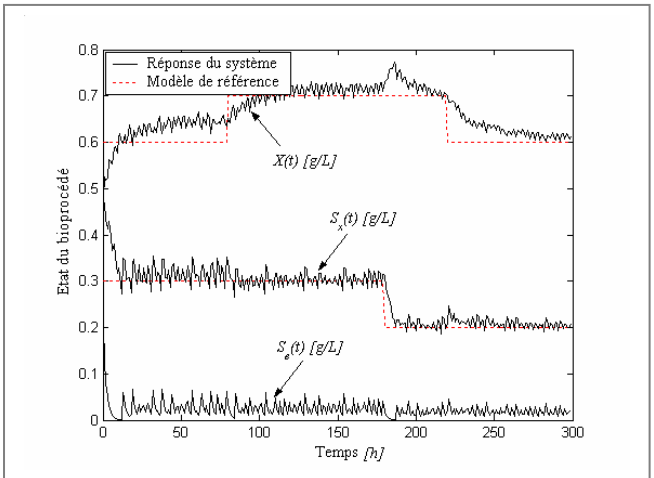


Fig. 4.33 : Evolution des états du bioprocédé.

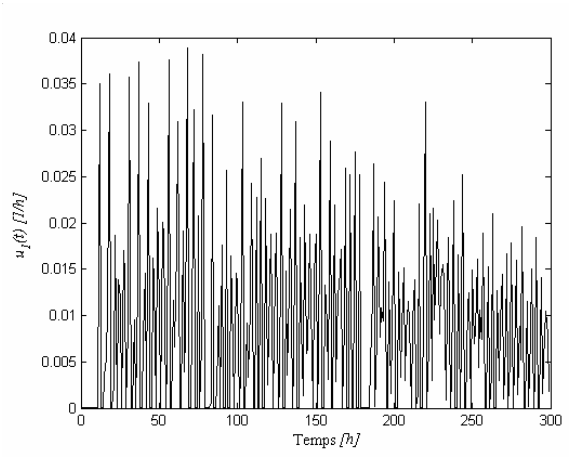


Fig. 4.34 : Signal de commande $u_1(t)$.

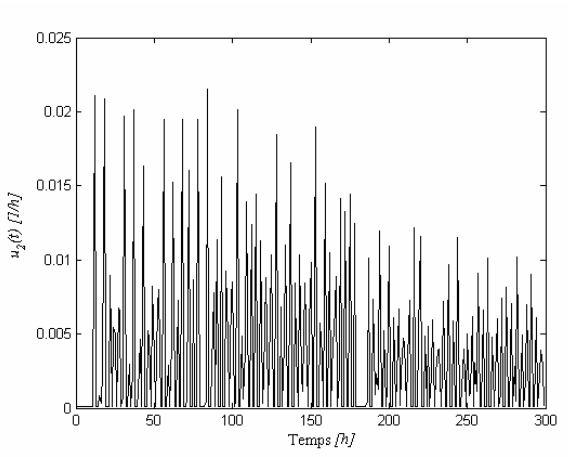


Fig. 4.35 : Signal de commande $u_2(t)$.

4.6. Conclusions

L'objectif de ce chapitre est de présenter les différentes possibilités qu'offrent les modèles flous du type TS pour la commande des systèmes non linéaires multivariables en utilisant la stratégie de commande PDC. La structure des règles de TS est bien adaptée à la représentation multimodèle. Chaque règle est un modèle local. Le concept de multimodèle flou est une technique de modélisation et d'identification des systèmes non linéaires. Ce concept simplifie bien l'analyse de la stabilité par l'approche énergétique de Lyapunov.

Après avoir introduit la notion du concept multimodèle, nous avons donné un aperçu général sur les méthodes de l'analyse des SIF. Le problème de la stabilité des modèles flous de type TS est aussi abordé. La plupart des travaux font appel à l'approche énergétique de Lyapunov pour la vérification des conditions de stabilité. Dans la phase de synthèse, la détermination de la loi de commande revient à déterminer pour chaque modèle local une matrice (resp. vecteur) des gains de retour d'état. La formulation des problèmes de stabilité sous une forme LMI est l'un des aspects intéressants du nouveau concept multimodèle.

Dans ce chapitre, nous avons aussi développé une méthodologie de synthèse permettant la prise en compte explicite de plusieurs spécifications structurelles et fonctionnelles simultanément. Nous avons exploité le concept du multimodèle pour la synthèse d'une loi de commande floue tout en assurant la stabilité en BF. L'idée consiste à déterminer un ensemble réduit de règles avec la définition des zones de fonctionnement (fonction d'appartenance) pour chaque variable d'état du système. Autour de ces zones sera établie le multimodèle par la linéarisation du modèle du système (phase de modélisation floue). Le comportement global résulte de la « fusion » par le formalisme de TS de l'ensemble des comportements locaux. Les AG représentent l'outil d'optimisation intervenant dans toutes les étapes de synthèse de la loi de commande. L'algorithme développé est appliqué pour la commande en poursuite d'un bioprocédé de traitement des eaux usées dans l'industrie du papier. Le bioprocédé est modélisé par un système multivariable fortement non linéaire. La base des modèles linéaires est obtenue en se basant sur la variation des états d'équilibre du système sur des zones de linéarités prédéfinies d'une manière approximative par la simulation du comportement du bioprocédé. La phase d'exploration génétique nous permet de spécifier les zones de fonctionnement du procédé. La principale question abordée dans ce travail est l'amélioration et l'optimisation des performances du bioprocédé à l'aide d'outils de modélisation multimodèle floue : *i*) Choix du nombre de modèles. *ii*) La garantie de la stabilité en BF. *iii*) La robustesse par rapport aux erreurs d'approximations et aux perturbations.

Les résultats de simulation montrent les bonnes performances de l'algorithme développé et confirment l'efficacité de la loi de commande en poursuite des trajectoires désirées.

Conclusions & perspectives

A – Conclusions

La motivation principale de ce travail *a été accomplie et concrétisée* par la mise en oeuvre des capacités offertes par les TIA, c'est-à-dire des systèmes hybrides, en vue de développer des approches pour la synthèse des lois de commande simples, efficaces et robustes. La formulation des problèmes de commande a été faite avec une combinaison des SIF, des RNA et des AG.

Notre travail a expliqué, explicitement, les problèmes rencontrés dans la phase de conception des SIF. Il s'agit, essentiellement, de :

- Problèmes de dimensionnement de la base des règles floues du SIF, c'est-à-dire, sa structure.
- Problèmes de configuration des paramètres de la base de connaissances.
- Problèmes de stabilité/robustesse de l'ensemble contrôleur/procédé à commander.

Les travaux présentés, dans cette thèse, s'articulent autour des principaux axes de la modélisation et de la commande floue du type TS1 pour les systèmes non linéaires complexes. Ces travaux ont dépisté, d'une façon cohérente, 2 grandes voies :

a. La première a relaté la commande floue des systèmes non linéaires.
b. La seconde s'est occupée de la modélisation floue d'un système non linéaire à partir des données numériques et de la commande floue du type PDC.

La première piste a finalisé la synthétisation d'une loi de commande floue stable et robuste. De même, l'hybridation neuronale floue a confirmé la stratégie de commande proposée dans les objectifs de cette étude. Une structure RBF-floue étendue au raisonnement approximatif du type TS1 est utilisée. La modélisation de la RBF-floue à partir d'une base de données numériques a été basée sur :

- La normalisation des variables d'entrée.
- Partitionnement des univers de discours de ces variables. Les partitions doivent être fortes.
- La génération des paramètres de TS, ainsi que les poids des règles intervenant dans l'analyse de la stabilité en BF.
- La construction d'une base des règles réduites (minimisation du nombre des couches). La stratégie de réduction de la base de règles consiste à ne retenir que les règles qui assurent une couverture de l'espace d'entrée.
- L'apprentissage fin des paramètres du réseau RBF-floue. Plus le nombre de règles sera faible, plus le SIF correspondant pourra affiner ses conclusions par des descentes de gradient complémentaires.

L'algorithme hybride proposé et utilisé a rendu l'optimisation structurelle et paramétrique du contrôleur RBF-flou une réalité indéniable. Les 2 étapes d'apprentissage sélectionnées et appliquées ont relaté :

- L'optimisation globale de l'ensemble structure & paramètres du réseau contrôleur proposé. L'algorithme utilisé est un AG à objectifs multiples (multiobjectifs).
- La confirmation en second lieu, après l'identification initiale de la topologie et les paramètres du réseau contrôleur, de l'efficacité de l'algorithme d'apprentissage employé. Ce dernier est basé sur l'application de la technique de la descente du gradient pour un réglage plus fin des paramètres des conséquences des règles.

La simplicité de la loi de commande, l'efficacité, la stabilité et la robustesse sont des facteurs, de premier ordre, intervenant dans le critère des performances à optimiser. L'étude réalisée et son application, en plus de l'efficacité, de la simplicité et de la sécurité de fonctionnement du procédé, la stabilité de la boucle de contrôle et la robustesse du contrôleur vis-à-vis des perturbations ont montré l'importance de ses facteurs dans l'optimisation des performances des systèmes complexes fortement non linéaires. L'ordre de priorité a été mesuré par la capacité du système de contrôle pour réaliser d'une façon très efficaces les objectifs visés par le concepteur. Dans notre cas, les objectifs sont classés en fonction du degré d'importance des facteurs conceptuels.

- La précision & la stabilité.
- La réduction de complexité de la loi de commande.
- La réduction de l'effort de contrôle (énergie) à appliquer sur le procédé.

On doit, une fois de plus, souligner et confirmer que selon le type des variables d'entrées, la loi de commande synthétisée, on peut avoir des similarités avec les contrôleurs conventionnels du type PID. Les non linéarités des gains des actions de proportionnalité, de dérivation et d'intégrale intervenant dans la construction de la loi de commande ont, vraiment, confirmé la différence et la supériorité du contrôleur ainsi développé sur les PID conventionnels.

De nombreuses applications, en simulation, ont été faites pour valider les méthodologies proposées et mettre en évidence que le compromis performances/simplicité recherché est tout à fait réalisable. Ces applications ont été faites sur divers procédés pilotes, notamment, un système de contrôle de la température d'un bain d'eau, un pendule inversé et un robot manipulateur à 2 degrés de liberté. Elles ont, particulièrement, permit de justifier les spécifications et des exigences du cahier de charges.

Dans la deuxième partie de la thèse, les différentes possibilités qu'offrent les modèles flous du type TS pour la modélisation et la commande des systèmes non linéaires multivariables en utilisant la stratégie de commande PDC . Le concept du multimodèle est exploité pour la synthèse d'une loi de commande floue. Ce concept simplifie bien l'analyse de la stabilité par l'approche énergétique de Lyapunov. L'idée consistait à déterminer un ensemble réduit de modèle (de règles) avec la définition des zones de fonctionnement (fonction d'appartenance) pour chaque variable d'état du système. Autour de ces zones a été établie le multimodèle par la linéarisation du modèle du système (phase de modélisation floue). Le comportement global résulte de la « fusion » par le formalisme de TS de l'ensemble des comportements locaux. Les AG représentent l'outil d'optimisation intervenant dans toutes les étapes de synthèse de la loi de commande et de l'analyse de la stabilité en BF.

La stratégie de l'analyse et de la synthèse des lois de commande dans le contexte multimodèle a été conçue autour de 3 axes :

- Le modèle flou issu du processus à commander par linéarisation autour des points de fonctionnement choisis. Le modèle flou modélise d'une manière approximative le processus à étudier.
- La loi de commande floue par retour d'état (PDC).
- Des conditions de stabilité du modèle en BF.

La stratégie de commande, ainsi développée, a effectivement assuré, non seulement, la stabilité du système mais aussi sa robustesse en présence de phénomènes perturbateurs.

L'algorithme développé est appliqué pour la commande en poursuite d'un bioprocédé de traitement des eaux usées dans l'industrie du papier. Le bioprocédé est modélisé par un système multivariable fortement non linéaire. Les résultats de simulation montrent les bonnes performances de l'algorithme développé et confirment l'efficacité de la loi de commande en poursuite des trajectoires désirées.

En premier lieu, nous avons établies les fondations nécessaires à la maîtrise des TIA : LF, RNA et AG. Après avoir introduit des concepts de base concernant la structure générale et les différents types des modèles flous, nous avons étudié plus particulièrement les caractéristiques structurelles et paramétriques de ces modèles. A l'issue de cette étude, nous avons constaté qu'ils sont des systèmes non linéaires adaptatifs, capables de manipuler non seulement des données numériques mais également des informations linguistiques. La propriété d'approximation universelle ne donne pas une méthode générale de conception et de construction du SIF. Elle garantit, seulement, son existence. La modélisation de ces systèmes n'était pas toujours évidente. C'est pourquoi le flou adaptatif a fait son apparition. Les lois d'adaptation proposées dans la littérature agissent sur le comportement global du SIF : Sa configuration structurelle & paramétrique. La difficulté de sa mise en oeuvre réside dans la mise au point d'une base de connaissances floues (structure & paramètres du SIF). Ce problème peut être résolu par l'utilisation des méthodes d'extraction automatique des connaissances : Les méthodes analytiques, les TIA, ...etc. Par la suite, nous avons présentés les caractéristiques fondamentales des RNA. La topologie du réseau, les fonctions d'activation des neurones, le seuil de ces fonctions et l'algorithme d'apprentissage utilisés sont des facteurs significatifs dans l'emploi des modèles connexionnistes. Nous avons, aussi, exposé en détail les différentes étapes nécessaires à l'exécution d'un AG, configuration paramétriques, initialisation de la population, évaluation, sélection pour la reproduction, test d'arrêt et les améliorations de l'AG de base pour l'aider à se converger vers une solution optimale. De plus, une étude bibliographique a été, exhaustivement, synthétisée dont le but est d'étudier les propositions pour l'amélioration des performances de ces TIA.

En seconde phase, nous avons exposé les différents modes d'hybridation des TIA : SIF-AG, SIF-RNA et SIF-RNA-AG. Une étude bibliographique approfondie sur les différentes méthodes de modélisation des SIF a été établie. Nous avons commencé par les différentes techniques d'optimisation des SIF par les AG. Ensuite, l'accent est mis sur les différentes structures de réseaux neuronaux - flous «RNF», développées au cours de ces dernières années, alliés avec leurs algorithmes d'apprentissage. Dans ce contexte, un classement de différentes structures RNF a été présenté, en se basant sur le type du raisonnement utilisé ainsi que sur la dynamique développée par le SIF.

En troisième position, une méthodologie de synthèse d'une loi de commande simple, efficace, stable et robuste pour une classe des systèmes non linéaires a été établie. La stratégie de commande proposée dans cette phase repose, essentiellement, sur l'hybridation RNF. Un réseau RBF-flou étendu au raisonnement approximatif du type TS1 a été utilisé. Le problème de dimensionnement de la base des règles floues du SIF (structure), le problème de configuration des paramètres de la base de connaissances et les problèmes de stabilité/robustesse de l'ensemble contrôleur/procédé à commander ont fait l'objet de notre nouvelle étude. Nous avons commencé par exposé les problèmes rencontrés dans la phase de conception des SIF et les solutions proposées dans la littérature. Une recherche bibliographique approfondie sur les méthodes de résolution du problème de dimensionnement des SIF a été faite. Par la suite, nous avons exposé en détail la méthodologie de synthèse de la loi de commande, les caractéristiques structurelles et algorithmiques développées pour cette synthèse ainsi que des simulations sur la commande de quelques systèmes non linéaires. D'après les résultats de simulations, nous avons pu constater que la stratégie de commande proposée ne présente pas, uniquement, une robustesse vis-à-vis des incertitudes, mais aussi une robustesse en présence d'un changement dans les paramètres des systèmes non linéaires ainsi que le changement des conditions de fonctionnement.

Dans la dernière phase, après avoir introduit la notion du concept multimodèle, nous avons donné un aperçu général sur les méthodes de l'analyse des SIF. Le problème de la stabilité des modèles flous de type TS a été, aussi, abordé et concrétisé par notre simulation. La plupart des travaux font appel à l'approche énergétique de Lyapunov pour la vérification des conditions de stabilité. Dans la phase de synthèse, la détermination de la loi de commande revient à déterminer pour chaque modèle local une matrice (resp. vecteur) des gains de retour d'état. La formulation des problèmes de stabilité sous une forme LMI est l'un des aspects intéressants du concept multimodèle. Une stratégie de modélisation et de synthèse des lois de commande floues a été développée dans cette partie de la thèse. Le concept de multimodèle a été utilisé. Une exploration génétique de l'espace de recherche structurelle et paramétrique a été exploitée pour la synthèse d'une loi de commande floue assurant la précision, l'efficacité, la simplicité et la stabilité au sens de Lyapunov. Pour illustrer la validité de l'algorithme proposé, un exemple de systèmes non linéaires du type MIMO est examiné.

La méthode proposée permet d'établir la synthèse d'une loi de commande et de donner des conditions suffisantes de stabilité du modèle en BF. Le nombre minimal de règles (resp. le nombre des modèles locaux capables de représenter, convenablement, le système non linéaire) est l'avantage du nouveau algorithme développé. Pour illustrer les performances de l'algorithme proposé pour la synthèse d'une loi de commande floue tout en assurant la stabilité en BF, nous avons considéré la commande en poursuite d'un bioprocédé de traitement des eaux usées dans l'industrie du papier. Les résultats de simulation montrent les bonnes performances de l'algorithme développé et confirment l'efficacité de la loi de commande en poursuite des trajectoires désirées.

Enfin, le succès des stratégies de commande floue provient de l'algorithme de synthèse de l'ensemble structure & paramètres du SIF d'un côté. De l'autre côté et en grande partie, il repose sur leur potentiel de transfert de connaissances expertisées sur le processus en règles de commande floue. De plus, l'écriture de la base de règles et la définition de la structure du contrôle flou nécessite un minimum d'informations (connaissances) sur le procédé à commander.

B – Perspectives & propositions

Au niveau de l'implantation des commandes synthétisées à base des TIA, de récentes études se sont consacrées au développement de composants intégrant des processeurs flous ou des RNA. Les commandes spécifiques, de plus en plus, rapides ont fait évoluer des dispositifs à base de DSP vers des structures moins coûteuses à base de FPGA et d'ASIC. Il faut souligner que le nombre des réalisations pratiques reste encore limité.

La finalité du contrôle étant de commander des systèmes pour qu'ils atteignent un objectif, préalablement, fixé. Il est important pour toute théorie d'étudier le passage de la théorie à la pratique. Du Soft Computing à ce qu'on appelle le ***Hard Computing***.

Dans le cadre de développement futur de ce travail, spécialement, dans le cadre de la préparation de l'habilitation universitaire pour la direction de la recherche, on va se concentrer et se dédier beaucoup plus vers :

- Le développement et l'application des *TIA* ***hybrides améliorées***.
- ***L'amélioration des performances*** par l'amélioration et l'établissement des algorithmes de synthèse.
- Aux applications ***en temps réel*** des algorithmes ainsi développés.

Annexe A

Simple objective ou multiobjective ?

A.1. Contribution 1 :

Le critère de performances désirées peut être formulé de la façon suivante :

Objective 1 :

Minimiser J_1

$$\text{avec} \quad J_1 = \frac{\beta_1}{\max\_k} \cdot \sum_{k=0}^{\max\_k-1} \left| e(k) \right|$$

Sous les contraintes :

$$\begin{cases} 0 < \beta_1 < 1 \\ \sum_{i=1}^{n} \left| \Theta_i(\cdot) \right| \cdot \xi_i \le C_u \end{cases}$$

Objective 2 :

Minimiser J_2

$$\text{avec} \quad J_2 = \frac{\beta_2}{\max\_règle} \cdot \sum_{i=1}^{\max\_règle} \left\{ \tilde{c}_F^{\,i} \right\}$$

Sous les contraintes :

$$\begin{cases} 0 < \beta_2 < 1 \\ \sum_{i=1}^{\max\_règle} \left\{ \tilde{c}_F^{\,i} \right\} \ge 1 \\ \sum_{l=1}^{N} \rho_i(\tilde{x}) \ne 0 \end{cases}$$

Objective 3 :

Minimiser J_3

$$\text{avec} \quad J_3 = \frac{\beta_3}{\max\_k} \cdot \sum_{k=0}^{\max\_k-1} \left| u(k) \right|$$

Sous les contraintes :

$$\begin{cases} 0 < \beta_3 < 1 \\ \sum_{i=1}^{n} \left| \Theta_i(\cdot) \right| \cdot \xi_i \le C_u \end{cases}$$

Ces objectives peuvent être recombinés pour constituer un critère qui satisfait toutes les conditions de conception. L'expression (3.35) est une solution adéquate qui peut simplifiée la tâche d'optimisation.

A.2. Contribution 2 :

Même principe avec la deuxième contribution. Il s'agit du système multivariable.

Annexe B

Les différents algorithmes développés dans cette thèse sont exécutés dans l'environnement de programmation Borland C++ 4.5. Un PC Pentium 4, 3Ghz et 256 Mo RAM est utilisé comme outil de conception et de développement. Les caractéristiques fondamentales de différents contrôleurs sont données dans les tableaux B.1 et B.2 pour les 2 contributions 1 & 2, respectivement.

B.1. Contribution 1 :

Tab. B.1 : Caractéristiques des contrôleurs proposés pour la contribution 1.

Contrôleur	*Max_gen*	*Max_itération*	Temps d'exécution	Remarques
NL_P (3TS)	100	100	1.34 [min]	• La phase d'analyse des contraintes est l'étape où le temps d'exécution sera long. • La taille du chromosome (resp. celle de la population) est un facteur significatif. • Le choix du type du contrôleur dépend du type de système à commander et des objectifs visés.
NL_PI (55TS)	300	200	8.52 [min]	
NL_PD (55TS)	200	150	6.28 [min]	

B.2. Contribution 2 :

Tab. B.2 : Caractéristiques des contrôleurs proposés pour la contribution 2.

Contrôleur	*Max_gen*	Temps d'exécution	Remarques
55TS	*Variable*	4.34 [min]	• La phase d'analyse (resp. de calcul) des contraintes (resp. les conditions de stabilité) est l'étape où le temps d'exécution sera long. • Le choix du type du contrôleur dépend du type de système à commander et des objectifs visés.

Remarques :

- xyTS signifie x sous ensembles flous pour la première variable et y sous ensembles flous pour la deuxième variable d'entrée. TS signifie un SIF du type TS.
- *Les résultats obtenus peuvent être comparés avec ceux mentionnés dans les références bibliographiques.*

Références bibliographiques

[1] L. A. Zadeh, "Soft computing and fuzzy logic*", IEEE Software,* 1994, pp. 48-56.

[2] L. A. Zadeh, "Fuzzy sets", *Information and Control*, Vol. 8, 1965, pp. 338-353.

[3] Kumpati S. Narendra, and Kannan Parthasarathy, "Identification and control of dynamical systems using neural networks", *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, March 1990, pp. 4-26.

[4] J. H. Holland, "*Adaptation in natural and artificial systems*", University of Michigan, Ann Arbor, 1975.

[5] T. Takagi, M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control'', *IEEE Transactions on Systems, Man an. Cybernetics*, Vol. SMC-15, No. 1, 1985, pp. 116–132.

[6] D. Dubois & H. Prade, "The three semantics of fuzzy sets", *Fuzzy Sets and Systems*, Vol. 90, 1997, pp. 141-150.

[7] L. X. Wang, "Stable adaptive fuzzy control of nonlinear systems", *IEEE Transactions on Fuzzy Systems*, Vol. 1, No. 2, May 1993, pp. 146-155.

[8] L. X. Wang, "*Adaptive fuzzy systems and control: Design and stability analysis*", Prentice-Hall, Englewood Clifs, NJ, 1994.

[9] J. T. Spooner, K. M. Passino, "Stable adaptive control using fuzzy systems and neural networks", *IEEE Transactions on Fuzzy Systems*, Vol. 4, 1996, pp. 339-359.

[10] L. X. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation and orthogonal least-squares learning", *IEEE Transactions on Neural Networks*, Vol. 3, 1992, pp. 807-814.

[11] M.S. Ju, D.L. Yang, "Design of adaptive fuzzy control based on natural control laws", *Fuzzy Sets and Systems*, Vol. 81, 1996, pp. 191-204.

[12] Raymond Hanus, "*Automatique avancées 1 : Techniques d'identification et d'estimation*", Hermes Sciences Publications, Lavoisier, 2007.

[13] Mohammed Chadli, "*Stabilité et commande de systèmes décrits par des multimodèles*", Thèse de docteur de l'Institut National Polytechnique de Loraine, France, 2002.

[14] H. O. Wang, K. Tanaka, M. Griffin, "An approach to fuzzy control of nonlinear systems : Stability and design issues", *IEEE Transactions on Fuzzy Systems*, Vol. 4, No. 1, 1996, pp. 14-23.

[15] T. A. Johansenn, A. B. Foss, "Nonlinear local model representation for adaptive systems", *IEEE International Conference on Intelligent Control and Instrumentation*, Vol. 2, 1992, pp. 677-682.

[16] H. O. Wang, K. Tanaka, M. Griffin, "Parallel distributed compensation of nonlinear systems by Takagi-Sugeno fuzzy model", *Proceeding of the IEEE International Conference on Fuzzy Systems, Fuzz-IEEE'95*, Yokohama, Japan, 1995, pp. 531-538.

[17] M. Margaliot, G. Langholz, "Fuzzy Lyapunov-based approach to the design of fuzzy controllers", *Fuzzy Sets and Systems*, Vol. 106, 1999, pp. 49-59.

[18] Stephen Boyd, Laurent El Ghaoui, Eric Feron and Venkataramanan Balakrishnan, "*Linear Matrix Inequalities in System and Control Theory*", SIAM, Pheladelphia, 1994.

[19] H. B. Verbruggen, H. J. Ziimmermann, R. Babũska, "*Fuzzy algorithms for control*", International Series in Intelligent Technologies, Kluwer Academic publisher, 1999.

[20] Angelo Ciaramella, Roberto Tagliaferri, Witold Pedrycs, "Inference systems by using ordinal sum and genetic algorithms", *IEEE Annual Meeting of the Fuzzy Information Processing NAFIPS '04*, 2004, pp. 629-634.

[21] L. Zadeh, "Fuzzy logic", *IEEE Computer*, Vol. 21, No. 4, 1988, pp. 83-93.

[22] Mamdani, E. H., "Application of fuzzy algorithms for simple dynamic plant", *Proc. IEE*, Vol. 121, No. 12, 1974, pp. 1558-1588.

[23] J. Harris, "*Fuzzy logic applications in engineering science*", Springer, 2006.
[24] J. Echanobe, I. del Campo, J.M. Tarela, "Issues concerning the analysis and implementation of a class of fuzzy controllers", *Fuzzy Sets and Systems*, Vol. 155, 2005, pp. 252-271.
[25] Leocundo Aguilar, Patricia Melin, Oscar Castillo, "Intelligent control of a stepping motor drive using a hybrid neuro-fuzzy ANFIS approach", *Applied Soft Computing*, Vol. 3, 2003, pp. 209-219.
[26] Ali Zilouchian, Mo. Jamshidi, "*Intelligent control systems using soft computing methodologies*", CRC Press LLC, 2001.
[27] Y. Tsukamoto, "*An approach to fuzzy reasoning model*", in Advances in Fuzzy Set Theory and Applications, Amsterdam, North-Holland, 1979.
[28] Christer Carlsson, Robert Fullér, "Multiobjective linguistic optimization", *Fuzzy Sets and Systems*, Vol. 115, 2000, pp. 5-10.
[29] Christer Carlsson, Robert Fullér, "Optimisation under fuzzy if-then rules", *Fuzzy Sets and Systems*, Vol. 111, 2001, pp. 111-120.
[30] Yigang Shi and P.C. Sen, "A new defuzzification method for fuzzy control of power converters", *Industry Applications Conference, 2000. Conference Record of the 2000 IEEE,* Vol. 2, pp. 1202-1209.
[31] M. R. Haghifam, A. R. Sedighi and O. P. Malik, "Development of a fuzzy inference system based on genetic algorithm for high-impedance fault detection", *IEE Proc. Gener. Transm. Distrib.*, Vol. 153, No. 3, May 2006, pp. 359-367.
[32] K. Gasso, G. Mourot et J. Ragot, "Identification of an output error Takagi-Sugeno model", *IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 1, 2000, pp. 14-19.
[33] Piero P. Bonissone, Anil Varma, Kareem S. Aggour, Feng Xue, "Design of local fuzzy models using evolutionary algorithms", *Computational Statistics & Data Analysis*, Vol. 51, 2006, pp. 398-416.
[34] Carlos Andrés Peña-Reyes and Moshe Sipper, "Fuzzy CoCo: A Cooperative-Coevolutionary approach to fuzzy modeling", *IEEE Transactions on Fuzzy Systems*, Vol. 9, No. 5, October 2001, pp. 727–737.
[35] Hisao Ishibuchi and Yusuke Nojima, "Performance Evaluation of Evolutionary Multiobjective Approaches to the Design of Fuzzy Rule-Based Ensemble Classifiers", *Proceedings of the Fifth International Conference on Hybrid Intelligent Systems* (HIS'05), 2005, pp. 1-6.
[36] Tomoharu Nakashima, Gerald Schaefer, YasuyukiYokota, Hisao Ishibuchi, "A weighted fuzzy classifier and its application to image processing tasks", *Fuzzy Sets and Systems*, Vol. 158, 2007, pp. 284-294.
[37] L. T. Kóczy, K. Hirota, "Interpolative reasoning with insufficient evidence in sparse fuzzy rule base", *Information Sciences*, Vol. 71, 1993, pp. 169-201.
[38] Chuen Lee, "Fuzzy logic in control systems: Fuzzy logic controller - part I", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, No. 2, March/April 1990, pp. 404-418.
[39] Jorge Casillas, Oscar Cordón, Maria José Del Jesus, Francisco Herrera, "Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction", *IEEE Transactions on Fuzzy Systems*, Vol. 13, No. 1, February 2005, pp. 13-29.
[40] V. N. Huynh, T. B. Ho, Y. Nakamori, "A parametric representation of linguistic hedges in Zadeh's fuzzy logic", *International Journal of Approximate Reasoning*, Vol. 30, 2002, pp. 203-223.
[41] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes", *IEEE Transactions on Systems, Man and Cybernetics*, SMC-3, 1973, pp. 28-44.
[42] Bin-Da Liu, Chuen-Yau Chen and Ju-Ying Tsao, "Design of adaptive fuzzy logic controller based on Linguistic-Hedge concepts and genetic algorithms", *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 31, No. 1, February 2001, pp. 32-53.
[43] Amitava Chatterjee, Patrick Siarry, "A PSO-aided neuro-fuzzy classifier employing linguistic hedge concepts", *Expert Systems with Applications*, Vol. 33, 2007, pp. 1097-1109.

[44] I. Guyon, "Neural networks and applications: Tutorial", *Physics report*, Vol. 207, Issue 3-5, September 1991; pp. 215-259.
[45] T. Yamada, T. Yabuta, "Neural network controller using auto tuning method for nonlinear functions", *IEEE Transactions on Neural Networks*, Vol. 3, No. 4, July 1992, pp. 595-601.
[46] Robert Babuška, Henk Verbruggen, "Neuro-fuzzy methods for nonlinear system identification", *Annual Reviews in Control*, Vol. 27, 2003, pp. 73-85.
[47] Vijander Singh, Indra Gupta, H.O. Gupta, "ANN-based estimator for distillation using Levenberg–Marquardt approach", *Engineering Applications of Artificial Intelligence*, Vol. 20, 2007, pp. 249-259.
[48] J. M. Renders, "*Algorithmes génétiques et réseaux de neurones : Applications à la commande de processus*", Hermès, Bruxelles, 1995.
[49] R. B. Boozarjomehry, W. Y. Svrcek, "Automatic design of neural network structures", *Computers and Chemical Engineering*, Vol. 25, 2001, pp. 1075-1088.
[50] **A. Soukkou, A. Khellaf, S. Leulmi**, "Commande neuronale à apprentissage génétique d'un bras manipulateur à 02 DOF"*, Conférence sur le Génie Electrique CGE'04*, Ecole Militaire Polytechnique, Alger, 12-13 Avril 2005.
[51] Chandrakumar Bhumireddy and C. L. Philip Chen, "Genetic learning of functional link networks", *Proceedings of the 2003 International Joint Conference on Neural Networks*, Vol. 1, 20-24 July 2003, pp. 432-437.
[52] G. Dreyfus, J. M. Martinez, M. Samulides, M. B. Gordon, F. Badran, S. Thiria, L. Hérault, "*Réseaux de neurones : Méthodologie et applications*", Edition Eyrolles, Paris, 2004.
[53] Parag C. Pendharkar, "A threshold-varying artificial neural network approach for classification and its application to bankruptcy prediction problem", *Computers & Operations Research*, Vol. 32, 2005, pp. 2561–2582.
[54] Bernard Dubuisson, "*Diagnostic, intelligence artificielle et reconnaissance des formes*", Hermes Science Publications, 2001.
[55] D. Nguyen & B. Widrow, "Neural networks for self-learning control systems", *IEEE Control Systems Magazine*, Vol. 10, No. 3, 1994, pp. 18-23.
[56] S. Haykin, "*Neural networks: A comprehensive foundation*", Macmillan College Publisher Comp., 1994.
[57] Léon Personnaz, Isabelle Rivals, "*Réseaux de neurones formels pour la modélisation, la commande et la classification*", CNRS Editions, Paris, 2003.
[58] V. N. Vapnik and A. J. Chervonenkis, "The necessary and sufficient conditions for consistency in the empirical risk minimization method", *Pattern Recognition and Image Analysis*, Vol. 1, No. 3, 1991, pp. 283-305.
[59] V. N. Vapnik, "*Statistical learning theory*", New York, Wiley, 1998.
[60] Stephen J. Verzi, Gregory L. Heileman, Michael Georgiopoulos, "Boosted ARTMAP: Modifications to fuzzy ARTMAP motivated by boosting theory", *Neural Networks*, Vol. 19, 2006, pp. 446-468.
[61] D. E. Rummelhart, G. E. Hinton, R. J. Williams, "Learning internal representations by error propagation", *Parallel Distributed Processing*, Vol. 1, MIT Press, Cambridge, 1986.
[62] R. D. Reed, R. J. Mark II, "*Neural Smithing: Supervised learning in feedforward artificial neural networks*", MIT Press, Cambridge, 1999.
[63] Wen Yu and Xiaoou Li, "Fuzzy identification using fuzzy neural networks with stable learning algorithms", *IEEE Transactions on Fuzzy Systems*, Vol. 12, No. 3, June 2004, pp. 411-420.
[64] Y. C. Liang, D.P. Feng, G.R. Liu, X.W. Yang and X. Han, "Neural identification of rock parameters using fuzzy adaptive learning parameters", *Computers and Structures*, Vol. 81, 2003, pp. 2373–2382.
[65] Jun Liu, Ding Liu**,** Hua-Yu Bai, Pu-Sheng Wu, "The optimal design of neural fuzzy controller", *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, Xi'an, 2-5 November 2003, pp. 544-547.
[66] Gou-Jen Wang, Tsorng-Chi Chen, "A robust parameters self-tuning learning algorithm for multilayer feedforward neural network", *Neurocomputing*, Vol. 25, 1999, pp. 167-189.

[67] Yu-Ju Chen, Tsung-Chuan Huang, Rey-Chue Hwang, "An effective learning of neural network by using RFBP learning algorithm", *Information Sciences*, Vol. 167, 2004, pp. 77-86.
[68] SUN Yan-jing, ZHANG Shen, MIAO Chang-xin, LI Jing-meng, "Improved BP neural network for transformer fault diagnosis", *J. China Univ. Mining & Technol.*, Vol. 17, No. 1, 2007, pp. 138-142.
[69] Amit Bhaya, Eugenius Kaszkurewicz, "Steepest descent with momentum for quadratic functions is a version of the conjugate gradient method", *Neural Networks*, Vol. 17, 2004, pp. 65-71.
[70] Sang-Min Kim, Woo-Yong Han, "Induction motor servo-drive using robust PID-like neuro-fuzzy controller", *Control Engineering Practice*, Vol. 14, No. 5, May 2006, pp. 481-487.
[71] M. R. Azimi Sadjadi, S. Sheedvash, F. O. Trujillo, "Recursive dynamic node creation in multilayer neural networks", *IEEE Transactions in Neural Networks*, Vol. 4, No. 2, March 1993, pp. 242-256.
[72] Kihwan Eom, Kyungkwon Jung, Harsha Sirisena, "Performance improvement of backpropagation algorithm by automatic activation function gain tuning using fuzzy logic", *Neurocomputing*, Vol. 50, 2003, pp. 439-460.
[73] B. Mendil, "Outils structurels et algorithmiques pour la commande intelligente", Thèse de doctorat d'état, Département d'Electronique, Université de Sétif, 2002.
[74] N. Intrator, D.Reisfeld, and Y. Ye Shurun, "Face recognition using a hybrid supervised/insupervised neural networks", *Pattern Recognition*, Vol. 17, 1996, pp. 67-76.
[75] M. N. H. Siddique and M. O. Tokhi, "GA-based neuro-fuzzy controller for flexible-link manipulator", *Proceeding of the 2002 IEEE International Conference on Control Applications*, September 18-20, 2002, Glasgow, U. K, pp. 471-475.
[76] **A. Soukkou, A. Khellaf, S. Leulmi**, "Genetic optimization of a neural controller", *AMSE International Conference on Modelling and Simulation, MS'2004*, Lyon-Villeurbanne France, Jul 5-7 2004, pp. 13.33-13.36.
[77] Pedro Ferreira, Pedro Ribeiro, Ana Antunes, Fernando Morgado Dias, "A high bit resolution FPGA implementation of a FNN with a new algorithm for the activation function", *Neurocomputing*, Vol. 71, 2007, pp. 71-77.
[78] Jesŭs Lăzaro, Jagoba Arias, Armando Astarloa, Unai Bidarte, Aitzol Zuloaga, "Hardware architecture for a general regression neural network coprocessor", *Neurocomputing*, Vol. 71, 2007, pp. 78-87.
[79] Ibrahim Turkoglu, "Hardware implementation of varicap diode's ANN model using PIC microcontrollers", *Sensors and Actuators A*, Vol. 138, 2007, pp. 288-293.
[80] David E. Goldberg, "*Algorithmes génétiques : Exploration, optimisation et apprentissage automatique*", Addison-Wesley, 1994.
[81] Z. Michalewics, "*Genetic Algorithms + Data Structures = Evolution Programs*", Springer Verlag. 3rd Edition, 1999.
[82] Lance Chambers, "*The practical handbook of genetic algorithms: Applications*", Second Edition, Chapman & Hall/CRC, 2001.
[83] JAMSHIDI Mohammad, KROHLING Renato A, Dos Santos Coelho Leandro, Fleming Peter, "*Robust control systems with genetic algorithms*", CRC; First Edition, 2002.
[84] Randy L. Haupt, Sue Ellen Haupt, "*Practical Genetic Algorithms*", Second Edition, John Wiley & Sons, In., Publication, 2004.
[85] B. J. Park, W. Pedrycz and S. K. Oh, "Identification of fuzzy models with the aid of evolutionary data granulation", *IEE Proc. Control Theory Appl.*, Vol. 184*,* No. 5*,* September 2001, pp. 406-418.
[86] Sung-Kwun Oh, Witold Pedrycz, "Genetic optimization-driven multi-layer hybrid fuzzy neural networks", *Simulation Modelling Practice and Theory*, Vol. 14, 2006, pp. 597-613.
[87] Thomas Bäck, "*Evolutionary computation 1: Basic algorithms and operators*", Taylor & Francis, 2000.
[88] Kejing He, Li Zheng, Shoubin Dong, Liqun Tang, Jianfeng Wu, Chunmiao Zheng, "PGO : A parallel computing platform for global optimization based on genetic algorithm", *Computers & Geosciences*, Vol. 33, 2007, pp. 357-366.

[89] Z. Zhu, D. J. Mulvaney, V. A. Chouliaras, "Hardware implementation of a novel genetic algorithm", *Neurocomputing*, Vol. 71, 2007, pp. 95-106.
[90] Viorel Minzu, Liviu Beldiman, "Some aspects concerning the implementation of a parallel hybrid metaheuristic", *Engineering Applications of Artificial Intelligence*, Vol. 20, 2007, pp. 993-999.
[91] K.L. Lo and M.O. Sadegh, "Systematic method for the design of a full–scale fuzzy PID controller for SVC to control power system stability", *IEE Proc. Gener. Transm. Distrib*, Vol. 150, No. 3, May 2003, pp 297-304.
[92] D. B. Fogel, "*Evolutionary computation: Towards new philosophy of machine intelligence*", IEEE Press, Piscataway, N. J, 1995.
[93] **A. Soukkou, S. Leulmi, A. Khellaf, M. Grimes**, "Control of dynamical systems: An intelligent approach", *International Journal of Control, Automation, and Systems (IJCAS),* Vol. 6, No. 4, August 2008, pp. 583-595.
[94] Agoston Eiben, Robert Hinterding, and Zbigniew Michalewicz. "Parameter control in evolutionary algorithms", *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 2, 1999, pp. 124-141.
[95] L. Davis, "*Handbook of genetic algorithms*", VNR, New York, 1991.
[96] J. Grefenstette, "Optimisation of control parameters for genetic algorithms", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 6, No.1, 1986, pp. 122-128.
[97] T. Bãck, "The interaction of mutation rate, selection, and self-adaptation within a genetic algoritms", *Proceeding of the Second Conference on Parallel Problem Solving from Nature,* Brusselles, Belegium, September 1992, pp. 85-94.
[98] Srinivas, Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms", IEEE *Transactions on Systems, Man, and Cybernetics*, Vol. 24, No. 4, 1994, pp. 17-26.
[99] Chang-Kuo Chen, Hang-Hong Kuo, Jun-Juh Yan, Teh-Lu Liao, "GA-based PID active queue management control design for a class of TCP communication networks", *Expert Systems with Applications*, Vol. xx, 2008, pp. xxx-xxx.
[100] K.C. Tan, C.K. Goh, Y.J. Yang, T.H. Lee, "Evolving better population distribution and exploration in evolutionary multi-objective optimization", *European Journal of Operational Research,* Vol. 171, No. 2, June 2006, pp. 463-495.
[101] Teo Lian Seng, M. Bin Khalid, R. Yusof, "Tuning of a neuro-fuzzy controller by genetic algorithm", *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics,* Vol. 29, No. 2, April 1999, pp. 226-236.
[102] Sardar Anisul Haque, A.B.M Zohrul Kabir and Ruhul A Sarker, "Optimization model for opportunistic replacement policy using genetic algorithm with fuzzy logic controller", *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.* 8-12 December 2003, Vol. 4, pp. 2837-2843.
[103] Jun Zhang, H. S. H. Chung and B. J. Hu, "Adaptive probabilities of crossover and mutation in genetic algorithms based on clustering technique", *Congress on Evolutionary Computation, 2004. CEC2004*, Vol. 2, 19-23 June 2004, pp. 2280-2287.
[104] Krzysztof Pytel, Grzegorz Kluka, Andrzej Szymonik, "Fuzzy methods of driving genetic algorithms", *Fourth International Workshop on Robot Motion and Control*, June 17-20, 2004, pp. 339-343.
[105] Zhiming Liu, Jiliu Zhou and Su Lai, "New adaptive genetic algorithm based on ranking", *Proceeding of the Second International Conference on Machine Learning and Cybernetics*, Xi'an, 2-5 November 2003, pp. 1841-1844.
[106] Li-Min Su, Hong Zhang, Chao-Zhen Hou and Xiu-Qin Pan, "Research on an improved genetic algorithm based knowledge acquisition", *Proceeding of the First International Conference on Machine Learning and Cybernetics*, Beijing, 4-5 November 2002, pp. 455-458.
[107] Hanan A. Kamal, "Optimal control of fedbatch fermentation processes using adaptive genetic algorithm", *Twentieth National Radio Science Conference*, March 18-20, 2003, Cairo, Egypt, pp. 1-11.

[108] Xuefeng F. Yan, Dezhao Z. Chen, Shangxu X. Hu, "Chaos-genetic algorithms for optimizing the operating conditions based on RBF-PLS model", *Computer and Chemical Engineering*, Vol. 27, 2003, pp. 1393-1404.
[109] Gwo-Ching Liao and Ta-Peng Tsao, "Application of a fuzzy neural network combined with a chaos genetic algorithm and simulated annealing to short-term load forecasting", *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 3, June 2006, pp. 330-340.
[110] Qingzhang Lü, Guoli Shen, Ruqin Yu, "A chaotic approach to maintain the population diversity of genetic algorithm in network training", *Computational Biology and Chemistry*, Vol. 27, 2003, pp. 363-371.
[111] You Yong, Sheng Wanxing, Wang Sunam, "Study of chaos genetic algorithms and its application in neural networks", *TENCON'02, IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering*, Vol. 1, 2002, pp. 232-235.
[112] Kenny Q. Zhu, "A diversity-controlling adaptive genetic algorithm for the vehicle routing problem with time windows", *Proceeding of the 15th IEEE International Conference on Tools with Artificial Intelligence* (ICTAI'03), 3-5 Nov. 2003, pp. 176-183.
[113] W.W. Tan, F. Lu, A.P. Loh, K.C. Tan, "Modeling and control of a pilot pH plant using genetic algorithm", *Engineering Applications of Artificial Intelligence*, Vol. 18, 2005, pp. 485-494.
[114] Wei-Der Chang, "Nonlinear system identification and control using a real-coded genetic algorithm", *Applied Mathematical Modelling*, Vol. 31, 2007, pp. 541-550.
[115] M. A. Rodrigo, A. Seco, J. Ferrer, J. M. Penya-roja, J. L. Valverde, "Nonlinear control of an activated sludge aeration process : Use of fuzzy techniques for tuning PID controllers", *ISA Transactions*, Vol. 38, 1999, pp. 231-241.
[116] S.Kirkpatrick & M. P. Vecchi, "Optimization by simulated annealing", *Science*, Vol. 220, No. 4598, pp. 671-680.
[117] F. Glover, "Tabu search – Part I", *ORSA Journal on Computing*, Vol. 1, 1989, pp. 190-206.
[118] F. Glover, "Tabu search – Part II ", *ORSA Journal on Computing*, Vol. 2, 1990, pp. 4-32.
[119] Pierre Yves Glorennec, "*Algorithmes d'apprentissage pour systèmes d'inférence floue*", Editions Hermes, Paris, 1999.
[120] Serge Guillaume, "*Induction des règles floues interprétables*", Thèse de doctorat, Institut National des Sciences Appliquées de Toulouse, France, Novembre 2001.
[121] L. C. Bezdek, R. J. Hathaway, M. J. Sabin, W. T. Tucker, "Convergence theory for fuzzy c-means : Counterexamples and repairs'', *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 17, No. 5, 1987, pp. 873-877.
[122] Yan Shi, Masaharu Mizumoto, "An improvement of neuro-fuzzy learning algorithm for tuning fuzzy rules", *Fuzzy Sets and Systems*, Vol. 118, 2001, pp. 339-350.
[123] Jelena Godjevac, "*Neuro-fuzzy controllers : Design & application*", Presses polytechniques et universitaires Romandes, Lausanne, Collection META, 1997.
[124] Il Kim, Jae-Hyun lee, Eun-Oh Bang, "A new approach to adaptive membership function for fuzzy inference system", *1999 Third International Conference on Knowledge-Based Intelligent Information Engineering Systems*, 31st Aug-1st September 1999, Adelaide, Australia, pp.112-116.
[125] Hanli Wang, Sam Kwong, Yaochu Jin, WeiWei, K.F. Man, "Multi-objective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction", *Fuzzy Sets and Systems*, Vol. 149, 2005, pp. 149-186.
[126] Han-Xiong Li and Shaocheng Tong, "A Hybrid Adaptive Fuzzy Control for A Class of Nonlinear MIMO Systems", *IEEE Transactions on Fuzzy Systems*, Vol. 11, No. 1, February 2003, pp. 24-34.
[127] L. X. Wang and J. Mendel, "Generating fuzzy rules by learning from examples", *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 22, July 1992, pp. 1414-1427.
[128] N. Golea, A. Golea, K. Benmahammed, "Stable indirect fuzzy adaptive control", *Fuzzy Sets and Systems*, Vol. 137, 2003, pp. 353-366.
[129] Sung-Woo Kim, Ju-Jang Lee, "Design of a fuzzy controller with fuzzy sliding surface", *Fuzzy Sets and Systems*, Vol. 71, 1995, pp. 359-367.

[130] Lon-Chen Hung, Hung-Ping Lin, Hung-Yuan Chung, "Design of self-tuning fuzzy sliding mode control for TORA system", *Expert Systems with Applications*, Vol. 32, 2007, pp. 201-212.
[131] Tai-Zu Wu, Yau-Tarng Juang, "Design of variable structure control for fuzzy nonlinear systems", *Expert Systems with Applications*, Vol. xxx, 2008, pp. xxx-xxx.
[132] Chun-Fei Hsu, Chih-Min Lin, "Fuzzy-identification-based adaptive controller design via backstepping approach", *Fuzzy Sets and Systems*, Vol. 151, 2005, pp. 43-57.
[133] Yan-Jun Liu, Wei Wang, "Adaptive fuzzy control for a class of uncertain nonaffine nonlinear systems", Information Sciences, Vol. 177, 2007, pp. 3901-3917.
[134] Kazuo Tanaka and H. O. Wang, "*Fuzzy control systems, design and analysis: A Linear Matrix Inequality Approach*", JOHN WILEY & SONS, INC., 2001.
[135] M. Mahfouf, S. Kandiah, D.A. Linken, "Fuzzy model-based predictive control using an ARX structure with feedforward", *Fuzzy Sets and Systems*, Vol. 125, 2002, pp. 39-59.
[136] M. Mahfouf, S. Kandiah and D. A. Linkens, "Adaptive estimation for fuzzy TSK model-based predictive control", *Transactions of the Institute of Measurement and Control*, Vol. 23, No. 1, 2001, pp. 31-50.
[137] Mahdi Mahfouf, Maysam F. Abbod and Darek A. Linkens, "Online elicitation of Mamdani-type fuzzy rules via TSK-based generalised predictive control", *IEEE Transactions on Systems, Man and Cybernetics- Part B: Cybernetics*, Vol. 33, No. 3, June 2003, pp. 465-475.
[138] R. C. Eberhart & Y. Shi, "Comparison between genetic algorithms and particle swarm optimization", *in Proc. IEEE Int. Conf. Evol. Comput*., Auchorage, AK, May 1998, pp. 84-89.
[139] Cihan Karakuzu, "Fuzzy controller training using particle swarm optimization for nonlinear system control", *ISA Transactions*, Vol. 47, No. 2, April 2008, pp. 29-39.
[140] S. E. Papadakis, J. B. Theocharis, "A GA-based fuzzy modelling approach for generating TSK models", *Fuzzy Sets and Systems*, Vol. 131, 2002, pp. 121-152.
[141] Cheol W. Lee, Yung C. Shin, "Construction of fuzzy systems using least-squares method and genetic algorithm", *Fuzzy Sets and Systems*, Vol. 137, 2003, pp.297-323.
[142] Liang Wang, John Yen, "Extracting fuzzy rules for system modelling using a hybrid of genetic algorithms and Kalman filter", *Fuzzy Sets and Systems*, Vol. 101, 1999, pp. 353-362.
[143] Baolin Wu, Xinghuo Yu, "Fuzzy modelling and identification with genetic algorithm based learning", *Fuzzy Sets and Systems*, Vol. 113, 2000, pp. 351-365.
[144] Francesco Cupertino, Vincenzo Giordano, David Nasi, Biagio Turchiano, "A hybrid approach to adaptive fuzzy control based on genetic algorithms", *2004 IEEE International Conference on Systems, Man and Cybernetics*, 2004, pp. 3607-3612.
[145] Vincenzo Giordano, David Naso, and Biagio Turchiano, "Combining genetic algorithms and Lyapunov-based adaptation for online design of fuzzy controllers", *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 36, No. 5, October 2006, pp. 1118-1127.
[146] Zne-Jung Lee, "A novel hybrid algorithm for function approximation", *Expert Systems with Applications*, Vol. 34, No. 1, January 2008, pp. 384-390.
[147] Zong-Mu Yeh, "A systematic method for design of multivariable fuzzy logic control systems", *IEEE Transactions on Fuzzy Systems*, Vol. 7, No. 5, October 1999, pp. 741-752.
[148] C. L. Karr, "Design of an adaptive fuzzy logic controller using a genetic algorithm", *Proceeding of the Fourth International Conference on Genetic Algorithms*, San Mateo, Canada, 1991, pp. 450-457.
[149] C. L. Karr, "Genetic algorithm for fuzzy logic controller", *AI Expert 2*, 1991, pp. 26-33.
[150] C. L. Karr, E. J. Gentry, "Fuzzy control of pH using genetic algorithms", *IEEE Transactions on Fuzzy Systems*, Vol. 1, 1993, pp. 46-53.
[151] P. Thrift, "Fuzzy logic synthesis with genetic algorithms", *Proceeding of the Fourth International Conference on Genetic Algorithms*, San Mateo, Canada, 1991, pp. 509-513.
[152] P.T. Chan, W.F. Xie, A.B. Rad, "Tuning of fuzzy controller for an open-loop unstable system: a genetic approach", *Fuzzy Sets and Systems*, Vol. 111, 2000, pp. 137-152.

[153] H. J. Cho, "Automatic rule generation for fuzzy controllers using genetic algorithms: a study on representation scheme and mutation rate", *7thProceeding of the IEEE International Conference on Fuzzy Systems*, Anchorage, AK, 1998, pp. 1290-1295.
[154] D.T. Pham, D. Karaboga, "Optimum design of fuzzy logic controllers using genetic algorithms", *Journal of Systems Engineering*, Vol. 1, 1991, pp. 114-118.
[155] N. Xiong, L. Litz, "Reduction of fuzzy control rules by means of premise learning - method and case study", *Fuzzy Sets and Systems*, Vol. 132, 2002, pp. 217-231.
[156] A. Varsek, T. Urbancic, B. Filipic, "Genetic algorithm in controller design and tuning", IEEE Transactions on Systems, Man and Cybernetics, Vol. 5, 1993, pp. 1330-1339.
[157] A. Homaifar and Ed Mc Cormick, "Simultaneous design of membership functions and rules sets for fuzzy controllers using genetic algorithms", *IEEE Transactions on Fuzzy Systems*, Vol. 3, No. 2, May 1995, pp. 161-176.
[158] Jinwoo Kim, Yoonkeon Moon, and Bernard P. Zeigler, "Designing fuzzy net controllers using genetic algorithms", *IEEE control systems Magazine*, June 1995, pp. 66-72.
[159] Jinwoo Kim, and Bernard P. Zeigler, "Designing fuzzy logic controllers using a multiresolutional search paradigm", *IEEE Transaction on Fuzzy Systems*, Vol. 4, No. 3, August 1996, pp. 213-226.
[160] Yi-Sheng Zhou and Lin-Ying Lai, "Optimal design for fuzzy controllers by genetic algorithms", *IEEE Transactions on Industry Applications*, Vol. 36, No. 1, January 2000, pp. 93-97.
[161] K. Belarbi, F. Titel, W. Bourebia, K. Benmahammed, "Design of Mamdani fuzzy logic controllers with rule base minimisation using genetic algorithm", *Engineering Applications of Artificial Intelligence*, Vol. 18, No. 7, October 2005, pp. 875-880.
[162] Tandra Pal and Nikhil Pal, "SOGARG: A self-organized genetic algorithm-based rule generation scheme for fuzzy controllers", *IEEE Transactions on Evolutionary Computation*, Vol.7, No. 4, 2003, pp. 397-415.
[163] Chia-Feng Juang, Jiann-Yow Lin, and Chin-Teng Lin, "Genetic Reinforcement Learning through Symbiotic Evolution for Fuzzy Controller Design", *IEEE Transactions on Systems, Man and Cybernetics- Part B: Cybernetics,* Vol. 30, No. 2, April 2000, pp.290-302.
[164] David Moriarty, Risto Miikulainen, "Efficient reinforcement learning through symbiotic evolution", *Machine Learning*, Vol. 22, 1996, pp. 11-33.
[165] Chia-Feng Juang, "A TSK-Type Recurrent Fuzzy Network for Dynamic Systems Processing by Neural Network and Genetic Algorithms", *IEEE Transactions on Fuzzy Systems,* Vol. 10, No. 2, April 2002, pp. 155-170.
[166] M. Jamei, M. Mahfouf, D.A. Linkens, "Elicitation and fine-tuning of fuzzy control rules using symbiotic evolution", *Fuzzy Sets and Systems*, Vol. 147, 2004, pp. 57-74.
[167] Bernadette Bouchon-Meunir, "*La logique floue et ses applications*", Addison-Wasley, October 1995.
[168] Bart Kosko, "*Neural Networks and Fuzzy Systems: A dynamical systems approach to machine intelligence*", Prentice Hall, 1992.
[169] A. Braham, "*Adaptation of Fuzzy Inference System Using Neural Learning*", Studies in Fuzziness and Soft Computing, Vol. 181, Springer-Verlag Berlin Heidelberg 2005, pp. 53-83.
[170] Jyh-Shing R. Jang, "Self Learning Fuzzy Controllers Based on Temporal Back Propagation", *IEEE Transactions on Neural Networks*, Vol. 3, No. 5, September 1992, pp. 714-723.
[171] Jyh-Shing R. Jang, "ANFIS: Adaptive-Network-based Fuzzy Inference Systems", *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 23, No. 3, May 1993, pp. 665-685.
[172] Jyh-Shing Roger Jang and Chuen-Tsai Sun, "Neuro-Fuzzy Modeling and Control", *Proceeding of the IEEE*, Vol. 83, No. 3, March 1995, pp. 378-406.
[173] M.A. Denaı, F. Palis, A. Zeghbib, "Modeling and control of non-linear systems using soft computing techniques", *Applied Soft Computing*, Vol. 7, 2007, pp. 728-738.
[174] I. Turkmen, K. Guney, "Genetic tracker with adaptive neuro-fuzzy inference system for multiple target tracking", Expert Systems with Applications, Vol. xxx, 2007, pp. xxx-xxx.

[175] Shinn-Jang Ho, Li-Sun Shu, Shinn-Ying Ho, "Optimizing fuzzy neural networks for tuning PID controllers using an Orthogonal Simulated Annealing algorithm OSA", *IEEE Transactions on Fuzzy Systems*, Vol. 14, No. 3, June 2006, pp. 421-434.
[176] M. Aliyari Shoorehdeli, M. Teshnehlab, A. K. Sedigh, "A novel training in ANFIS structure", *Proceeding of the 2006 American Control Conference,* Minneapolis, Minnesota, USA, June 14-16, 2006, pp. 5059-5064.
[177] Robert Babuška, Henk Verbruggen, "Neuro-fuzzy methods for nonlinear system identification", *Annual Reviews in Control*, Vol. 27, 2003, pp. 73-85.
[178] Mahdi Jalili-Kharaajoo, "Nonlinear system identification using ANFIS based on emotional learning", *Advances in Artificial Intelligence*, *IBERAMIA 2004, LNAI 3315*, 2004, Springer-Verlag Berlin Heidelberg 2004, pp. 697–707.
[179] Hamid Berenji & Pratap Khedkar, "Learning and tuning fuzzy controllers through reinforcements", *IEEE Transactions on Neural Networks*, Vol. 3, No. 5, 1992, pp.724-740.
[180] Carlos Kavka, "*Evolutionary design of geometric-based fuzzy systems*", Thèse de doctorat, Université de Paris-Sud, France, Juillet 2006.
[181] Mohammad Hossein Fazel Zarandi, Javid Jouzdani, and Ismail Burhan Turksen, "Generalized reinforcement learning fuzzy control with vague states", *Analysis and Design of Intelligent Systems using Soft Computing Techniques*, ASC 41, Springer-Verlag Berlin Heidelberg 2007, pp. 811–820.
[182] C. Zhou, "Neuro-fuzzy gait synthesis with reinforcement learning for a biped walking robot", *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, Vol. 4, No. 4, December 2000, pp. 238-250.
[183] M. J. D. Powell, "Radial Basis Functions for multivariable interpolation: A review", *IMA Conference on Algorithms for the Approximation of Functions and Data*, RMCS Shrivenham, 1985, pp. 143-167.
[184] J. Moody, C. Darken, "Fast learning in networks of locally-tuned process units", *Neural Computation*, Vol. 1, 1989, pp. 281-294.
[185] V. David S'anchez A., "Searching for a solution to the automatic RBF network design problem", *Neurocomputing*, Vol. 42, 2002, pp. 147-170.
[186] J. S. R. Jang, C. T. Sun, "Functional equivalence between radial basis function network and fuzzy inference systems", *IEEE Transactions on Neural Networks*, Vol. 4, 1993, pp. 156-159.
[187] N. Steele & Jelena Godjevac, "Adaptive radial basis function neural networks and fuzzy systems", *Conference on Computational Engineering in Systems Applications, CESA'96*, Lille, France, July 1996, pp. 143-148.
[188] Kwang Bo Cho, Bo Hyeun Wang, "Radial basis function adaptive fuzzy systems and their applications to system identification and prediction", *Fuzzy Set and Systems*, Vol 83, 1996, pp. 325-339.
[189] C. Harpham, C. W. Dawson, M. R. Brown, "A review of genetic algorithmq applied to training radial basis function networks", *Neural Computing and Applications*, Vol. 13, 2004, pp. 193-201.
[190] Zhong-Qiu Zhao, De-Shuang Huang, "A mended hybrid learning algorithm for radial basis function neural networks to improve generalization capability", *Applied Mathematical Modelling*, Vol. 31, No. 7, July 2007, pp. 1271-1281.
[191] A. M. S. Barreto, H. J. C. Barbosab, N. F. F. Ebecken, "GOLS-Genetic orthogonal least squares algorithm for training RBF networks", *Neurocomputing*, Vol. 69, 2006, pp. 2041–2064.
[192] Roman Neruda, Petra Kudovà, "Learning methods for radial basis function networks", *Future Generation Computer Systems*, Vol. 21, 2005, pp. 1131-1142.
[193] Jeen-Shing Wang and C. S. George Lee, "Self-ddaptive neuro-fuzzy inference systems for classification applications", *IEEE Transactions on Fuzzy Systems*, Vol. 10, No. 6, 2002, pp. 790-802.
[194] Ching-Hung Lee and Ching-Cheng Teng, "Identification and control of dynamic systems using recurrent fuzzy neural networks", *IEEE Transactions on Fuzzy Systems,* Vol. 8, No. 4, August 2000, pp. 349-366.

[195] R. C. Lin, W. D. Weng and C. T. Hsueh, "Design of an SCRFNN-based nonlinear channel Equaliser", *IEE Proc.-Commun*., Vol. 152, No. 6, December 2005, pp. 771-779.
[196] Faa-Jeng Lin, Po-Kai Huang and Wen-Der Chou, "Recurrent-fuzzy-neural-network-controlled linear induction motor servo drive using genetic algorithms", *IEEE Transactions on Industrial Electronics*, Vol. 54, No. 3, June 2007, pp. 1449-1461.
[197] Chih-Min Lin, Chun-Fei Hsu, "Supervisory recurrent fuzzy neural network control of wing rock for slender delta wings", *IEEE Transactions on Fuzzy Systems*, Vol. 12, No. 5, October 2004, pp. 733-742.
[198] Wei Sun, Yaonan Wang, "A recurrent fuzzy neural network based adaptive control and its application on robotic tracking control", Neural Information Processing - Letters and Reviews, Vol. 5, No. 1, October 2004, pp. 19-26.
[199] Faa-Jeng Lin, Rong-Jong Wai and Chun-Ming Hong, "Hybrid supervisory control using recurrent fuzzy neural network for tracking periodic inputs", *IEEE Transactions on Neural Networks*, Vol. 12, No. 1, January 2001, pp. 68-90.
[200] Faa-Jeng Lin and Rong-Jong Wai, "Hybrid control using recurrent fuzzy neural network for linear-induction motor servo drive", *IEEE Transactions on Fuzzy Systems*, Vol. 9, No. 1, February 2001, pp. 102-115.
[201] Ying-Chung Wang, Chiang-Ju Chien and Ching-Cheng Teng, "Takagi-Sugeno recurrent fuzzy neural networks for identification and control of dynamic systems", *2001 IEEE International Fuzzy Systems Conference*, pp. 537-540.
[202] Paris A. Mastorocostas and John B. Theocharis, "A recurrent fuzzy-neural model for dynamic system identification", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 32, No. 2, 2002, pp.176-190.
[203] Meng Joo Er, Wai Mun Wong and Shiqian Wu, "A comparative study of different methods for realizing DFNN algorithm", *Proceedings of the 38th Conference on Decision & Control*, Phoenix, Arizona USA, December 1999, pp. 2641-2642.
[204] T.G. Barbounis, J.B. Theocharis, "A locally recurrent fuzzy neural network with application to the wind speed prediction using spatial correlation", *Neurocomputing*, Vol. 70, 2007, pp. 1525-1542.
[205] Chia-Feng Juang, Jung-Shing Chen, "A recurrent fuzzy-network-based inverse modeling method for a temperature system control", *IEEE Transactions on Systems, Man and Cybernetics- Part C: Applications and Reviews,* Vol. 37, No. 3, May 2007, pp. 410-417.
[206] R. J. Williams and D. Zipser, "A learning algorithm for continually running recurrent neural networks", *Neural Computer*, Vol. 1, No. 2, 1989, pp. 270-280.
[207] Chia-Feng Juang, Yuan-Chang Liou, "TSK-type recurrent fuzzy network design by The hybrid of genetic algorithm and particle swarm optimization", *2004 IEEE International Conference on Systems, Man and Cybernetics,* 2004, pp. 2314-2318.
[208] Chia-Feng Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design", *IEEE Transactions on Systems, Man and Cybernetics- Part B: Cybernetics,* Vol. 34, No. 2, April 2004, pp. 997-1006.
[209] Chia-Feng Juang, I-Fang Chung, "Recurrent fuzzy network design using hybrid evolutionary learning algorithms", *Neurocomputing*, Vol. 44, 2007, pp. 1–10.
[210] Jinzhu Peng, Yaonan Wang and Wei Sun, "Trajectory-tracking control for mobile robot using recurrent fuzzy cerebellar model articulation controller", *Neural Information Processing - Letters and Review*s, Vol. 11, No. 1, January 2007, pp. 15-23.
[211] Floriberto Ortiz, Wen Yu, Marco Moreno-Armendariz, and Xiaoou Li, "Recurrent fuzzy CMAC for nonlinear system modeling", *Advances in Neural Networks – ISNN 2007, LNCS 4491*, 2007, pp. 487-495.
[212] Chi-Yung Lee, Cheng-Jian Lin, Huei-Jen Chen, "A self-constructing fuzzy CMAC model and its applications", *Information Sciences*, Vol. 177, 2007, pp. 264-280.
[213] Cheng-Jian Lin, Jia-Hong Lee, Chi-Yung Lee, "A novel hybrid learning algorithm for parametric fuzzy CMAC networks and its classification applications", *Expert Systems with Applications*, Vol. xxx, 2007, pp. xxx–xxx.
[214] J. Kim, N. Kasabov, "HyFIS: Adaptive neuro-fuzzy inference", *Neural Computer*, Vol. 1, No. 2, 1998, pp. 270-280.

[215] Z. J. Zhou, Z. Y. Mao and Peter K. S. Tam, "On designing an optimal Fuzzy Neural Network Controller using Genetic Algorithms", *Proceeding of the 3rd World Congress on Intelligent Control and Automation*, June 28- July 2, 2000, Hafei, P.R. China, pp. 391-395.

[216] K. Belarbi and F. Titel, "Genetic algorithm for the design of a class of fuzzy controllers: An Alternative Approach", *IEEE Transactions on Fuzzy Systems*, Vol. 8, No. 4, August 2000, pp. 398-405.

[217] Liang Hsuan Chen and Cheng-Hsiung Chiang, "New approach to intelligent control systems with self-exploring process", *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics,* Vol. 33, No. 1, February 2003, pp. 56-66.

[218] Detlef Nauck, Rudolf Kruse, "Neuro-fuzzy systems for function approximation", *Fuzzy Sets and Systems*, Vol. 101, 1999, pp. 261-271.

[219] A. Nurnberger, D. Nauck, R. Kruse, "Neuro-fuzzy control based on the NEFCON-model: recent developments", *Soft Computing, Springer-Verlag*, Vol. 2, 1999, pp. 168-182.

[220] Detlef D. Nauck, "Fuzzy data analysis with NEFCLASS", *International Journal of Approximate Reasoning*, Vol. 32, 2 003, pp. 103-130.

[221] Chia-Feng Juang and Chin-Teng Lin, "A recurrent self-organizing neural fuzzy inference network", *IEEE Transactions on Neural Networks,* Vol. 10, No. 4, July 1999, pp. 828-845.

[222] Cheng-Jian Lin; Chin-Teng Lin, "An ART-based fuzzy adaptive learning control network"**,** *IEEE Transactions on Fuzzy Systems*, Vol. 5, No. 4, 1997, pp. 477-496.

[223] I-Fang Chung, Cheng-Jian Lin_, Chin-Teng Lin, "A GA-based fuzzy adaptive learning control network", *Fuzzy Sets and Systems*, Vol. 112, 2000, pp. 65-84.

[224] Haisheng Lin, X. Z. Gao, Xianlin Huang et Zhuoyue Song, "A Self-organizing Fuzzy Neural Networks", *Soft Computing in Industrial Applications*, ASC 39, Springer-Verlag Berlin Heidelberg 2007, pp. 200-210.

[225] K.H. Quah, C. Quek, G. Leedham, "Reinforcement learning combined with a fuzzy adaptive learning control network (FALCON-R) for pattern classification", *Pattern Recognition*, Vol. 38, 2005, pp. 513-526.

[226] T.Z. Tan, C. Quek, G.S. Ng, E.Y.K. Ng, "A novel cognitive interpretation of breast cancer thermography with complementary learning fuzzy neural memory structure", *Expert Systems with Applications*, Vol. 33, 2007, pp. 652-666.

[227] Rahmat Shoureshi, Zhi Hu, "Tsukamoto-type neural fuzzy inference network", *Proceeding of the American Control Conference,* Chicago, Illinois, USA, June 2000, pp. 2463-2467.

[228] Ibanez-Guzman, J., Ketata, R. "Fuzzy logic control experiments on a single-link manipulator"**,** *IEEE International Symposium on Industrial Electronics, ISIE '94,* pp. 420-425.

[229] J.C. Bezdek, "Cluster validity with fuzzy sets", *J. Cybern*., Vol. 3, No.3, 1974, pp. 58-71.

[230] Min-You Chen, D.A. Linkens, "Rule-base self-generation and simplification for data-driven fuzzy models", *Fuzzy Sets and Systems*, Vol. 142, 2004, pp. 243-265.

[231] Magne Setnes, Robert Babuška, "Rule base reduction: Some comments on the use of orthogonal transforms", *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications & Reviews*, Vol. 31, No. 2, May 2001, pp. 199-206.

[232] J. Johensohn and J. Mendel, "Two pass orthogonal least-squares algorithm to train and reduce fuzzy logic systems", *Proceedings of the Third IEEE World Congress on Computational Intelligence*, 1994, pp. 696-700.

[233] Zaheeruddin, M.J. Anwer, "A simple technique for generation and minimization of fuzzy rules", *IEEE International Conference on Fuzzy Systems*, Nevada, May 2005, pp. 489-494.

[234] Jonh Y. Hung et. al. "Variable Structure Control: A Survey", *IEEE Transaction on Industrial Electronics,* Vol. 40, No.1, 1993, pp. 2-21.

[235] Kazuo Tanaka, Tadannri Taniguchi and Hua 0. Wang, "Generalized Takagi-Sugeno fuzzy systems: Rule reduction and robust control", *The Ninth IEEE International Conference on Fuzzy Systems,* Vol. 2, 7-10 May 2000, pp. 688-693.

[236] Tadannri Taniguchi, Kazuo Tanaka, Hiroshi Ohtake and Hua 0. Wang, "Model construction, rule reduction, and robust compensation for generalized form of Takagi-Sugeno fuzzy systems", *IEEE Transactions on Fuzzy Systems*, Vol. 9, No. 4, August 2001, pp. 525-538.

[237] Orsolya Takács and Annamária R. Várkonyi-Kóczy, "SVD-based complexity reduction of rule-bases with nonlinear antecedent fuzzy sets", *IEEE Transactions on Instrumentation and Measurement*, Vol. 51, No. 1, April 2002, pp. 217-221.
[238] Yehui Chen, Jiwen Dong and Bo Yang, "Automatic design of hierarchical TS-FS model using ant programming and PSO algorithm", *The Eleventh International Conference on Artificial Intelligence: Methodology, Systems, and Applications* (AIMSA), Varna, Bulgaria, September 2-4, 2004, Vol. 3192, 2004, pp. 285-294.
[239] J. Zhao, R. Gorez, V. Wertz, "Genetic algorithms for the elimination of redundancy and/or rule contribution assessment in fuzzy models", *Mathematics and Computers in Simulation*, Vol. 41, 1996, pp. 139-148.
[240] Rafael Alcalá, Jesús Alcalá-Fdez, Jorge Casillas, Oscar Cordéon, Francisco Herrera, "Hybrid learning models to get the interpretability–accuracy trade-off in fuzzy modelling", *Soft Computing*, Vol. 10, 2006, pp. 717-734.
[241] Pietari Pulkkinen, Hannu Koivisto, "Identification of interpretable and accurate fuzzy classifiers and function estimators with hybrid methods", *Applied Soft Computing*, Vol. 7, 2007, pp. 520-533.
[242] Sofiane Achiche, Luc Baron, Marek Balazinski, "Real/binary-like coded versus binary coded genetic algorithms to automatically generate fuzzy knowledge bases: A comparative study", *Engineering Applications of Artificial Intelligence*, Vol. 17, 2004, pp. 313-325.
[243] T. Pal, N. R. Pal, M. Pal, "Learning fuzzy rules for controllers with genetic algorithms", *International Journal of Intelligent Systems*, Vol. 18, 2003, pp. 569-592.
[244] Hai-Xiang Guo, Ke-Jun Zhu, Si-Wei Gao, Yue Li, Jing-Jing Zhou, "Extracting fuzzy rules based on fusion of soft computing in oil exploration management", *Expert Systems with Applications*, Vol. xxx, 2008, pp. xxx-xxx.
[245] Rafael Alcalá, Jesús Alcalá-Fdez, Francisco Herrera, José Otero, "Genetic learning of accurate and compact fuzzy rule based systems based on the 2-tuples linguistic representation", *International Journal of Approximate Reasoning*, Vol. 44, 2007, pp. 45-64.
[246] **A. Soukkou, A. Khellaf, S. Leulmi**, "Multiobjective Optimization of a Fuzzy PID controller"*, Archives of Control Sciences* (ACS), Vol. 16(LII), No. 4, 2006, pp. 445-461.
[247] Lina HNG, Hiroyuki INOUE, Kenji MIYASAKA, Mitsuru TSUKAMOTO, "Automatic generation of fuzzy classification systems using hyper-cone membership functions", *Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, July 16-20,2003, Kobe, Japan, pp. 658-663.
[248] Man Gyun Na, Sun Ho Shin, Dong Won Jung, Soong Pyung Kim, Ji Hwan Jeong Byung Chul Lee, "Estimation of break location and size for loss of coolant accidents using neural networks", *Nuclear Engineering and Design*, Vol. 232, 2004, pp. 289-300.
[249] V. Matellan, C. Fernandez, J. M. Molina, "Genetic Learning of fuzzy reactive controllers", *Robotics and Autonomous Systems*, Vol. 25, 1998, pp. 33-41.
[250] Feng Wan, Huilan Shang, Li-Xin Wang and You-Xian Sun, "How to determine the minimum number of fuzzy rules to achieve given accuracy: a computational geometric approach to SISO case", *Fuzzy Sets and Systems*, Vol. 150, No. 2, March 2004, pp. 199-209.
[251] Francesco Cupertino, Vincenzo Giordano, David Nasi, Biagio Turchiano, "A hybrid approach to adaptive fuzzy control based on genetic algorithms", *IEEE International Conference on Systems, Man and Cybernetics*, 2004, pp. 3607-3612.
[252] Vincenzo Giordano, David Naso, Biagio Turchiano, "Combining genetic algorithms and Lyapunov-based adaptation for online design of fuzzy controllers**",** *IEEE Transactions on Systems, Man and Cybernetics- Part B: Cybernetics*, Vol. 36, No. 5, October 2006, pp. 1118-1127.
[253] Chi-Ho Lee, Ming Yuchi and Jong-Hwan Kim, "Two phase optimization of fuzzy controller by evolutionary programming", *The 2003 Congress on Evolutionary Computation, CEC'03,* 8-12 Dec. 2003, Vol. 3, pp. 1949-1956.
[254] G. V. S. Raju, J. Zhou and R. A. Kisner, "Hierarchical fuzzy control", *International Journal of Control*, Vol. 54, No**.** 5, 1991, pp. 1201-1216.
[255] M. Jamshidi, "*Fuzzy control systems",* Springer-Verlag, Chapter of Soft Computing, 1997, pp. 42-56.

[256] Yulia Nikolaevna Ledeneva, Carlos Alberto Reyes García1, and José Antonio Calderón Martínez, "Automatic estimation of the fusion method parameters to reduce rule base of fuzzy control complex systems", *MICAI : Advances in Artificial Intelligence 2006, LNAI 4293*, Springer-Verlag Berlin Heidelberg 2006, Vol. 3789, pp. 146-155.

[257] Moon G. Joo and Jin S. Lee, "A class of hierarchical fuzzy systems with constraints on the fuzzy rules", *IEEE Transactions on Fuzzy Systems*, Vol. 13, No. 2, April 2005, pp. 194-203.

[258] G. V. S. Raju, J. Zhou, "Adaptive hierarchical fuzzy controller", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, No. 4, July/August 1993, pp. 973-980.

[259] Zong-Mu Yeh and Huang-Pin Chen, "A novel approach for multistage inference fuzzy control", *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics,* Vol. 28, No. 6, December 1998, pp. 935-946.

[260] Ming-Ling Lee, Hung-Yuan Chung, Fang-Ming Yu, "Modelling of hierarchical fuzzy systems", *Fuzzy Sets and Systems*, Vol. 138, 2003, pp. 343-361.

[261] J. Abonyi, R. Babuska, F. Szeifert, "Modified Gath–Geva fuzzy clustering for identification of Takagi–Sugeno fuzzy models", *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics*, Vol. 32, No. 5, 2002, pp. 612-621.

[262] Hisao Ishibuchi, Takashi Yamamoto, "Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining", *Fuzzy Sets and Systems*, Vol. 141, 2004, pp. 59-88.

[263] M. Zolghadri Jahromi, M. Taheri, "A proposed method for learning rule weights in fuzzy rule-based classification systems", *Fuzzy Sets and Systems*, Vol. 159, 2008, pp. 449- 459.

[264] Xi-Zhao Wang, Chun-Ru Dong, Tie-Gang Fan, "Training T-S norm neural networks to refine weights for fuzzy if–then rules", *Neurocomputing*, Vol. 70, 2007, pp. 2581-2587.

[265] Sung-Kwun Oh*,* Witold Pedrycz, Ho-Sung Park, ''Genetically optimized fuzzy polynomial neural networks'', *IEEE Transactions on Fuzzy Systems*, Vol. 14, No. 1, February 2006, pp. 125-144.

[266] Sung-Kwun Oh*,* Witold Pedrycz, Ho-Sung Park, ''Multilayer hybrid fuzzy neural networks: Synthesis via technologies of advanced computational intelligence'', *IEEE Transactions on Circuits and Systems*, Vol. 53, No. 3, Mars 2006, pp. 688-703.

[267] Hao Ying, "Constructing nonlinear variable gain controllers via the Takagi-Sugeno fuzzy control", *IEEE Transactions on Fuzzy Systems*, Vol. 6, No. 2, May 1998, pp. 226-234.

[268] Hao Ying, "Theory and application of a novel fuzzy PID controller using a simplified Takagi-Sugeno rule scheme", *Information Sciences*, Vol. 123, 2000, pp. 281-293.

[269] **A. Soukkou**, "*Conception d'un contrôleur neuronal flou par les algorithmes génétiques*", Mémoire de Magister, Département d'Electronique, Université de Sétif, 1999.

[270] **A. Soukkou, A. Khellaf & S. Leulmi**, "Systematic Design Procedure of TS-type Fuzzy Controllers"*, International Journal of Computational Intelligence and Applications (IJCIA),* Vol. 6, No. 4, December 2006, pp. 531-549.

[271] **A. Soukkou, S. Leulmi, A. Khellaf**, "Systematic design and hybrid learning of robust fuzzy neural network controller with reduced rule base"*, International Journal of Hybrid and Intelligent Systems* (IJHIS), Vol. 4, No. 2, June 2007, pp. 63-88.

[272] **A. Soukkou, A. Khellaf, S. Leulmi**, "Multiobjective optimization of robust Takagi-Sugeno fuzzy neural controller with hybrid learning algorithm"*, International Journal of Modelling, Identification and Control (IJMIC)*, Vol. 2, No. 4, 2007, pp. 332-346.

[273] Yibin Song, Peijin Wang and Kaili Li, "Complex model identification based on RBF neural network", *Advances in Neural Networks, LNCS Vol. 3174*, Springer-Verlag Berlin Heidelberg 2004, pp. 224-229.

[274] K. J. Aström and T. Hägglund, "*PID Controllers: Theory, Design, and Tuning",* Instrument Society of America, 1995.

[275] Rajani K. Mudi, Chanchal Dey, Tsu-Tian Lee, "An improved auto-tuning scheme for PI controllers", *ISA Transactions*, Vol. 47, 2008, pp. 45-52.

[276] Antonio Visioli, "*Practical PID control (Advances in industrial control)*", Springer, 2006.

[277] T. E. Marlin, *"Process control: Designing processes and control systems for dynamic performance",* Mc Graw-Hill, 1995.

[278] Yongho Lee and Sunwon Park, "PID controllers tunning for desired closed-loop responses for SI/SO systems", *Aiche Journal*, Vol. 44, N°. 4, January 1998, pp. 106-115.
[279] Wei-Der Chang, Rey-Chue Hwang, Jer-Guang Hsieh, "A multivariable on-line adaptive PID controller using auto-tuning neurons", *Engineering Applications of Artificial Intelligence*, Vol. 16, 2003, pp. 57-63.
[280] TU Diep Cong Thanh, Kyoung Kwan Ahn, "Nonlinear PID control to improve the control performance of 2 axes pneumatic artificial muscle manipulator using neural network", *Mechatronics*, Vol. 16, 2006, pp. 577-587.
[281] Sylvie Galichet & Laurent Foulloy, "Fuzzy controllers: Synthesis and equivalences", *IEEE Transactions on Fueey Systems*, Vol. 3, No. 2, 1995, pp. 140-148.
[282] **A. Soukkou, S. Leulmi, A. Khellaf**, "How to optimize the TS-fuzzy knowledge base to achieve a desired performances: Accuracy and robustness"*, International Journal of Optimal Control Applications and Methods (OCAM)*, Vol. 29, No. 1, 2008, pp. 19-40.
[283] V. Mukherjee, S.P. Ghoshal, "Intelligent particle swarm optimized fuzzy PID controller for AVR system", *Electric Power Systems Research*, Vol. 77, 2007, pp. 1689-1698.
[284] **A. Soukkou, A. Khellaf, S. Leulmi**, "Supervision neuro-floue a apprentissage génétique d'un PID robuste"*, Revue Sciences et Technologies*, Université Mentouri, Constantine, No. 23, Juin 2005, pp. 95-106.
[285] P. J. Escamilla-Ambrosio and N. Mort, "A novel design and tuning procedure for PID type fuzzy logic controllers", *2002 First International IEEE Symposium 'Intelligent Systems'*, September 2002, pp. 36-41.
[286] Han-Xiong Li, Lei Zhang, Kai-Yuan Cai, and Guanrong Chen, "An improved robust fuzzy-PID controller with optimal fuzzy reasoning", *IEEE Transactions on Systems, Man, and Cybernetics—Part B : Cybernetics*, Vol. 35, No. 6, December 2005, pp. 1283-1294.
[287] Bao-Gang Hu, George K. I. Mann, and Raymond G. Gosine, "A systematic study of fuzzy PID controllers-function-based evaluation approach", *IEEE Transactions on Fuzzy Systems*, Vol. 9, No. 5, October 2001, pp. 699-712.
[288] Zhihong Xiu, Guang Ren, "Optimisation design of TS-PID controllers based on genetic algorithms", *Proceeding of the 5th World Congress on Intelligent Control and Automation*, June 15-19, 2004, Hangzhou, P. R. China, pp. 2476-2480.
[289] Ya Lei Sun and Meng Joo Er, "Optimal hybrid control of linear and nonlinear systems", *Proceeding of the 40th IEEE Conference on Decision and Control*, Orlando, Florida USA, December 2001, pp. 1669-1674.
[290] Dongqing Feng, Lingjiao Dong, Minrui Fei, and Tiejun Chen, "Genetic algorithm based neuro-fuzzy network adaptive PID control and its applications", *CIS* 2004, Springer-Verlag Berlin Heidelberg 2004, LNCS 3314, pp. 330-335.
[291] James Carvajal, Guanrong Chen, Haluk Ogmen, "Fuzzy PID controller: Design, performance evaluation, and stability analysis", *Information Sciences*, Vol. 123, 2000, pp. 249-270.
[292] Jian-Xin Xu, Chang-Chieh Hang, Chen Liu, "Parallel structure and tuning of a fuzzy PID controller", *Automatica*, Vol. 36, 2000, pp. 673-684.
[293] D. Driankov, H. Hellendoorn, "*An introduction to fuzzy Control*", Springer Edition, 1993.
[294] H. X. Li, H. B. Gatland, "Conventional fuzzy control and its enhancement", *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 26, No. 5, 1996, pp. 791-797.
[295] Kuo-Ming Chang, "Adaptive control for a class of chaotic systems with nonlinear inputs and disturbances", *Chaos, Solitons and Fractals*, Vol. 36, 2008, pp. 460-468.
[296] R. Boukezzoula, "Commande floue d'une classe de systèmes non linéaires : Application au problème de suivi de trajectoire", Thèse de docteur de l'université de Savoie, France, 2000.
[297] Min Han, Yannan Sun, Yingnan Fan, "An improved fuzzy neural network based on TS model", *Expert Systems with Applications*, Vol. 34, 2008, pp. 2905-2920.
[298] Wen Li & Yoichi Hori, "An algorithm for extracting fuzzy rules based on RBF neural networks", *IEEE Transactions on Industrial Electronics*, Vol. 53, No. 4, August 2006, pp. 1269-1276.

[299] Stanley E. Woodard and Devendra P. Garg, "A numerical optimization approach for tuning fuzzy logic controllers", *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, Vol. 29, No. 4, August 1999, pp. 565-569.
[300] Mu-Song Chen and Redean Liou, "An efficient learning method of fuzzy inference system", *1999 IEEE International Fuzzy Systems Conference Proceedings*, August 22-25. 1999, Seoul, Korea, pp. 635-638.
[301] K. -L. Du and M. N. S. Swamy, "*Neural networks in a soft computing framework*", Springer Edition, 2006.
[302] Oscar Montiel, Oscar Castillo, Patricia Melin, Roberto Sepulveda, "The evolutionary learning rule for system identification", *Applied Soft Computing*, Vol. 3, 2003, pp. 343-352.
[303] F. Cuesta and A. Ollero, "*Intelligent mobile robot navigation*", STAR 16, Springer-Verlag Berlin Heidelberg 20005, pp. 7-49.
[304] Lianfei ZHAI, Tianyou CHAI, "Nonlinear decoupling PID control using neural networks and multiple models", *Journal of Control Theory and Applications*, Vol. 1, 2006, pp. 62-69.
[305] Cheng-Jian Lin, ''A GA-based neural fuzzy system for temperature control'', *Fuzzy Sets and Systems*, Vol. 143, 2004, pp. 311-333.
[306] C. Treesatayapun, S. Uatrongjit, "Adaptive controller with fuzzy rules emulated structure and its applications", *Engineering Applications of Artificial Intelligence*, Vol. 18, 2005, pp. 603-615.
[307] S. Labiod, M.S. Boucherit, and T.M. Guerra, "Commande adaptative floue d'une classe de systèmes non linéaires monovariables", *LFA'03*, Tours, France, 2003, pp. 169-176.
[308] Mehrhad Hojati and Saeed Gazor, "Hybrid adaptive fuzzy identification and control of nonlinear systems", *IEEE Transactions on Fuzzy Systems*, Vol. 10, No. 2, June 2002, pp. 198-210.
[309] Wei- Yen Wang, Chih- Yuan Cheng and Yih-Guang Leu, "An online GA-based output-feedback direct adaptive fuzzy-neural controller for uncertain nonlinear systems", *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, 2003, pp. 1-12.
[310] T. Fukuda and N. Kubota, "An intelligent robotic system based on a fuzzy approach", *Proc. IEEE*, Vol. 87, September 1999, pp. 1448-1470.
[311] J. Slotine and W. Li, "*Applied nonlinear control",* Englewood Cliffs, New Jersey: Prentice Hall, 1991.
[312] Kuu-Young Young and Shaw-Ji Shan, "An approach to enlarge learning space coverage for robot learning control", *IEEE Transactions on Fuzzy Systems*, Vol. 5, No. 4, November 1997, pp.150-159.
[313] Mario E. Magãna and Shinsuke Tagani, "An improved trajectory tracking decentralized adaptive controller for robot manipulator", *IEEE Transactions on Industrial Electronics*, Vol. 41, No. 5, October 1994, pp. 477-482.
[314] L. X. Wang, "A supervisory controller for fuzzy control systems that guarantees stability", *IEEE Transactions on Automatic and Control*, Vol. 39, No. 9, 1994, pp. 1845-1847.
[315] Abdelkader AKHENAK, "*Conception d'observateurs non linéaires par approche multimodèle : Application au diagnostic*", Thèse de docteur, Institut National Polytechnique de Lorraine, France, 2004.
[316] Rosario Toscano, "*Commande et diagnostic des systèmes dynamiques : Modélisation, Analyse, Commande par PID et par Retour d'état, Diagnostic*", Ellipses Edition Marketing S. A., 2005.
[317] Luc Jaulin, "*Représentation d'état pour la modélisation et la commande des systèmes*", Hermès Science Publications, Lavoisier, 2005.
[318] A. Kandel, Y. Luo and Y. Q. Zhang, "Stability analysis of fuzzy control systems", F*uzzy Sets and Systems*, Vol. 105, No. 1, July 1999, pp. 33-48.
[319] A. Ollero, J. Aracil, A. Garcia-Cerezo, "Robust design of rule-based fuzzy controllers", *Fuzzy Sets and Systems*, Vol. 70, 1995, pp. 249-273.
[320] Pierre-Frédéric TOULOTTE, "*Attelage virtuel pour véhicules automatisés*", Thèse de docteur, Université de Valenciennes et du Hainaut Cambrésis, France, 2006.

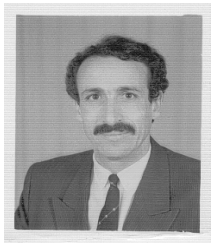
[321] Yann Morere, "*Mise en oeuvre de lois de commande pour les modèles flous de type Takagi-Sugeno*", Thèse de docteur, Université de Valenciennes et du Hainaut Cambrésis, France, 2001.
[322] Julio Concha, Aldo Cipriano, René Vidal, "Design of fuzzy controllers based on stability analysis", *Fuzzy Sets and Systems*, Vol. 121, 2001, pp. 25-38.
[323] Qu Sun, Renhu Li, Ping'an Zhang, "Stable adaptive fuzzy control of complex systems using fuzzy dynamicmodel", *Fuzzy Sets and Systems*, Vol. 133, 2003, pp. 1-17.
[324] Jooyoung Park, Jinsung Kim, Daihee Park, "LMI-based design of stabilizing fuzzy controllers for nonlinear systems described by Takagi-Sugeno fueey model", *Fuzzy Sets and Systems*, Vol. 122, 2001, pp. 73-82.
[325] C. M. Lin, Y. J. Mon, "A fuzzy PDC-based control for robotic systems", *Information Sciences*, Vol. 137, 2001, pp. 135-155.
[326] Cheng-Wu Chen, "Stability conditions of fuzzy systems and its application to structural and mechanical systems", *Advances in Engineering Software*, Vol. 37, 2006, pp. 624-629.
[327] Zhi-Hong Xiu, Guang Ren, "Stability analysis and systematic design of Takagi-Sugeno fuzzy control systems", *Fuzzy Sets and Systems*, Vol. 151, 2001, pp. 119-138.
[328] Kazuo Tanaka, Hiroshi Ohtake, Hua O. Wang, "A practical design approach to stabilization of a 3-DOF RC Helicopter", *IEEE Transactions on Control Systems Technology*, Vol. 12, No. 2, March 2004, pp. 315-325.
[329] M. J. Er, D. H. Lin, "A new approach for stabilising nonlinear systems with time delay", *International Journal of Intelligent Systems*, Vol. 17, 2002, pp. 289-302.
[330] **Soukkou, A. Khellaf, S. Leulmi, K. Bougeghdegh,** "Optimal control of a CSTR process", *Brazilian Journal of Chemical Engineering*, Vol. 25, No. 04, October-December 2008, pp. 809-822.
[331] Zhong Li, Wolfgang A. Halang, Guanrong Chen, "*Integration of fuzzy logic and chaos theory*", Studies in Fuzziness and Soft Computing, Volume 187, Springer-Verlag Berlin Heidelberg 2006.
[332] Thierry Marie Guerra, Laurent Vermeiren, "Control laws for Takagi-Sugeno fuzzy model", *Fuzzy Sets and Systems*, Vol. 120, 2001, pp. 95-108.
[333] M. Kannou, M. Benrejeb, S. Hajri-Gabouj et P. Borne, "Stabilité et commande multimodèle floue de systèmes complexes", *Conférence Internationale Francophone d'automatique CIFA 2004*, Douz, Tunisie, 22-24 Novembre 2004.
[334] D. Jenkins, K.M. Passino, "An introduction to nonlinear analysis of fuzzy control systems", *Journal of Intelligent and Fuzzy Systems*, Vol. 7, No. 1, 1999, pp. 75-103.
[335] Mehran Mesbahi, "Robustness analysis via the running time of the interior point methods", *Systems & Control Letters*, Vol. 44, 2001, pp. 355-361.
[336] Wook Chang, Jin Bae Park, Young Hoon Joo, "GA-based intelligent digital redesign of fuzzy-model-based controllers", *IEEE Transactions on Fuzzy Systems*, Vol. 11, No. 1, February 2003, pp. 35-44.
[337] Frank H. F. Leung, H. K. Lam, S. H. Ling, Peter K. S. Tam, "Optimal and stable fuzzy controllers for nonlinear systems based on an improved genetic algorithm", *IEEE Transactions on Industrial Electronics*, Vol. 51, No. 1, February 2004, pp. 172-182.
[338] Lanka Udawatta, Keigo Watanabe, Kazuo Kiguchi, Kiyotaka Izumi "Solution of global stability of fuzzy regulators via evolutionary computation", *Applied Soft Computing*, Vol. 4, 2004, pp. 25-34.
[339] Alain Berro, "*Optimisation multiobjectif et stratégies d'évolution en environnement dynamique*", Thèse de docteur en Informatique, Université des sciences sociales- Toulouse I, France, 2001.
[340] Yann Collette, Patrick Siarry, "*Optimisation multiobjectif*", Editions Eyrolles, 2002.
[341] Karima Turki & Olfa Boubaker, "Vers une modélisation multimodèle incertaine et une commande robuste séquencée des bioprocédés", *Conférence Internationale Francophone d'automatique CIFA 2004*, Douz, Tunisie, 22-24 Novembre 2004.
[342] W. C. Chen, Ni-Bin Chang, Jeng-Chung Chen, "Rough set-based hybrid fuzzy-neural controller design for industrial wastewater treatment", *Water Research*, Vol. 37, 2003, pp. 95-107.

[343] Ludovic Mailleret, Olivier Bernard, Jean-Philipe Steyer, "Nonlinear adaptive control for bioreactors with unknown kinetics", *Automatica*, Vol. 40, 2004, pp. 1379-1385.
[344] Denis Douchain, "*Automatique des bioprocédés*", Hermes Science Europe Ltd, 2001.
[345] G. Boari, I.M Mancini, E. Trulli, "Technologies for water and wastewater treatment", *Séminaires Méditerranéens CIHEAM - Options Mediterraneennes*, *Sér. A /n037, 1997.*
[346] B. Dahhou, G. Roux, G. Chamilothoris, "Modelling and adaptive predictive control of a continuous fermentation process", *Applied Mathematical Modelling*, Vol.16, 1992, pp.545-552.
[347] B. Dahhou, M. Lakrori, I. Queinnec, E. Ferret, A. Cheruy, "Control of a continuous fermentation process", *Journal of Process Control*, Vol. 2, No. 2, 1992, pp.103-111.
[348] B. Dahhou, G. Roux, A. Vasilache, "Neural networks for biomass growth rate modelling and model changement detection in a yeast fermentation process", *4th IFAC Conference: System Structure and Control*, Bucharest, Roumanie, 1997, pp. 469-474.
[349] Jeng-Chung Chen, Ni-Bin Chang, "Mining the fuzzy control rules of aeration in a Submerged Biofilm Wastewater Treatment Process", *Engineering Applications of Artificial Intelligence*, Vol. 20m No. 7, 2007, pp. 959-969.
[350] C. Ben Youssef and B. Dahhou, "Multivariable adaptive predictive control of an aerated lagoon for a wastewater treatment process", *Journal of Process Control,* Vol. 6, No. 5, 1996, pp. 265-275.
[351] F. Nejjari, G. Roux, B. Dahhou, A. Benhammou, "Estimation and optimal control design of a biological wastewater treatment process", *Mathematics and Computers in Simulation*, Vol. 48, 1999, pp. 269-280.
[352] Meka Srinivasarao, Sachin C. Patwardhan, Ravindra D. Gudi, "Nonlinear predictive control of irregularly sampled multirate systems using black box observers", *Journal of Process Control*, Vol. 17, No. 1m 2007, pp. 17-35.
[353] Victor-Hugo GRISALES PALACIO, "*Modélisation et commande floues de type Takagi-Sugeno Appliquées à un bioprocédé de traitement des eaux*", Thèse de doctorat de l'Université Paul Sabatier-Toulouse III, France, 2007.
[354] Cherif Ben Youssef and Jesús I. Torres Zuñiga, "Stable model-reference adaptive control and estimation of a wastewater treatment benchmark", *2002 IEEE International Conference on Systems, Man and Cybernetics*, 2002, Vol. 6, pp. 1-6.

Biographies

Ammar Soukkou was born in Djimla-Jijel, Algeria. He received his Diploma in Engineering from the Electronics department, University of Sétif, Algeria in 1996, the Diplôme d'Etudes Approfondies in Automatic Control Electronics department, University of Sétif in 1997, the Magister degree from the Laboratoire de l'intelligence artificielle et de l'Automatique from the Electronics department, University of Sétif in 1999. Since 2000 to 2005, he held different positions involved in industrial field and education. He is currently working towards her PhD in Electrical Engineering in the Sétif University. Since 2005, he has been an Assistant Professor at the Electronics department, University of Jijel, Algeria. His current interests include optimal and robust control, nonlinear systems, fuzzy logic, neural networks and evolutionary algorithms.

A. Soukkou
Department of Electronics, University of Jijel, BP. 98,
Ouled Aissa, Jijel 18000, Algeria.
Soukkkou_a@yahoo.fr



Abdelhafid Khellaf received the es_Science Licence degree in physics department from the University of Constantine, Algeria in 1975, the Diplôme des Etudes Approfondies DEA in Traitement de signal et Systèmes aléatoires, from the University of Renne (France) in 1975, the Doctorat 3eme cycle in Electronique et Télécommunication from the University of Renne (France) in 1979 and a PhD degree in Electrical Engineering from the University of Constantine, Algeria in 1990. He is a full Professor in the faculty of Engineering, Sétif University, Algeria. His current research interests are intelligent control of biotechnological processes, optimal control, fuzzy logic, neural networks and evolutionary algorithms.

A. Khellaf
Department of Electronics,
University of Ferhat Abbas-Setif 19000, Algeria.
ah_khellaf@yahoo.fr

Salah Leulmi received a State Engineer Degree in electric power systems from the National Polytechnique School of Algiers in Algeria (1976), a Master Degree of Engineering from RPI, Troy, NY, USA in electric power system engineering (1978) and a PhD in electrical engineering from ISU, Ames, Iowa, USA (1983). He is the author of more than twenty publications in journals & proceedings. He was the Head "Director" of the University of Skikda, Algeria. Currently, he is a professor & president of the scientific council of the Science & Engineering Faculty "School" at the University of Skikda. He is, also, a referee of four Algerian Journals & some proceedings & one overseas society "WSEAS" for proceedings & journals. He is, also actually, the President of the National State Commission of the Equivalency Degrees.

S. Leulmi
Department of Electrotechnics,
University of Skikda 21000, Algeria.
leulmi_salah@yahoo.fr