

## SUPERVISION NEURO-FLOUE A APPRENTISSAGE GENETIQUE D'UN PID ROBUSTE

Reçu le 29/04/2004 - Accepté le 10/05/2005

### Résumé

Dans cet article on présente de nouveaux outils pour la supervision et l'optimisation intelligente des systèmes hautement non linéaires à base de la logique floue. Les paramètres dominants caractérisant la base des connaissances floues (facteurs d'échelle des variables d'entrée/sortie, fonctions d'appartenance et règles d'inférence) sont optimisés par l'emploi des algorithmes génétiques (AG). Le conventionnel PID dans sa forme améliorée, où les coefficients de différentes actions  $K_p$ ,  $K_I$  et  $K_D$  sont non linéairement variables. Un système d'inférence floue à structure neuronale à apprentissage génétique joue le rôle de superviseur qui permet de donner des fonctions optimales à ces coefficients.

La structure du réseau superviseur est spécifiée par une combinaison entre le raisonnement flou mixte de Takagi-Sugeno et de Mamdani. Le codage optimal mixte, binaire-entier, est utilisé pour la construction des chromosomes définissant l'ensemble des paramètres nécessaires pour la conception du superviseur désiré. Ce nouveau superviseur se distingue lui même par un gain de dénormalisation (facteur d'échelle de sortie) qui varie linéairement avec les entrées floues. Sous certaines conditions, il devient similaire à un PID conventionnel à coefficients non-linéairement variables.

Les résultats de simulation sur la commande à temps optimal d'un modèle de référence prédéterminé pour le système de pendule inversé montrent bien l'efficacité et la robustesse de la méthode utilisée.

**Mots clés:** PID flou, algorithme génétique, réseau de neurones.

### Abstract

This article presents the application of a new generation of fuzzy logic supervisor (FLS) to the highly nonlinear systems. The dominant parameters characterizing the base of fuzzy knowledge: scaling factors of the Input/Output (I/O) variables, membership functions and the rule consequences are optimized by using the Genetic Algorithms (GA). The conventional PID in its improved form, where the coefficients of different actions  $K_p$ ,  $K_I$  and  $K_D$  are nonlinear variables. A fuzzy inference system with multilayer neural network structure with genetic training plays the role of supervisor who allows giving optimal functions to these coefficients.

The fuzzy structure is specified by a combination of the mixed Takagi-Sugeno's and Mamdani's fuzzy Reasoning TSM-FR. The mixed integer-binary optimal coding is utilized to construct the chromosomes, which define the same of necessary prevailing parameters for the conception of the desired supervisor. This new fuzzy supervisor stands out by a non standard gain (output scaling factor) which varies linearly with the fuzzy inputs. It becomes similar to the conventional PID controller with non-linearly variable coefficients. Computer simulation indicates that the designed fuzzy supervisor is satisfactory in PID control of a nonlinear system 'Inverted Pendulum'.

**Keywords:** Fuzzy PID, genetic algorithms, neural networks

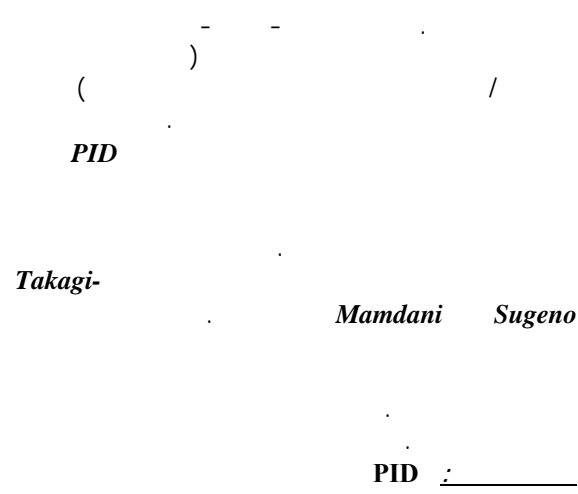
A. SOUKKOU\*  
A. KHELLAF\*\*  
S. LEULMI\*\*\*

\* Département d'Electronique  
Université de Jijel  
Algérie.

\*\* Institut d'Electronique,  
Université Ferhat Abbas-Sétif  
Algérie.

\*\*\* Institut d'Electrotechnique,  
Université de Skikda  
Algérie.

### ملخص



La commande des procédés industriels complexes, se heurte de plusieurs difficultés: *i)* Présence de non linéarités, *ii)* Les incertitudes et, *iii)* La complexité au niveau de la modélisation et d'élaboration des lois de commande convenables. Ce qui exige de nouvelles méthodes de contrôle plus sophistiqué ayant de grandes capacités d'apprentissage et de prise de décision dans des milieux incertains et perturbés. Malgré sa popularité dans l'industrie et sa facilité d'implantation, l'inconvénient du contrôleur PID (*Proportional-Integral-Derivative*) est le choix des coefficients de différentes actions  $K_p$ ,  $K_I$  et  $K_D$ . Yougho Lee [1] présente un bref historique sur les différentes méthodes de calcul des coefficients (gains)  $K_p$ ,  $K_I$  et  $K_D$ . Il a aussi proposé une nouvelle méthode de calcul basée sur le développement en série de Mc Laurin. En conséquence, beaucoup de techniques du réglage des paramètres basées sur différentes approches :



- Procédés empiriques, tel que méthodes Ziegler-Nichols [2].
- Méthodes analytiques, par exemple, le lieu géométrique des racines.
- Des méthodes d'optimisation, tel que Cian-Cone ou méthodes Lopez ... [3]. Ces techniques permettent d'obtenir les coefficients (gains) de différentes actions en se basant sur la minimisation d'un critère de performances désirées (IAE, ITAE...).

Les propriétés d'approximation et de traitement parallèle des systèmes à base de la logique floue [4] peuvent fournir des solutions plus efficaces aux problèmes de contrôle des systèmes complexes. L'avantage d'un tel système flou est que seules les connaissances du comportement de procédé à commander sont suffisantes pour la synthèse de la loi de commande. En termes plus spécifiques, la logique floue vise à modéliser les modes imprécis du raisonnement qui jouent un rôle très important dans la capacité humaine de prendre des décisions rationnelles dans un environnement incertain et imprécis. L'inconvénient majeur de l'approche traditionnelle pour la conception des systèmes flous réside au niveau de l'obtention des connaissances auprès d'un expert humain. Pour palier ce problème, la recherche s'est orientée vers l'élaboration des techniques optimales et systématiques pour la conception de ces systèmes.

Deux approches de conceptions les plus rencontrées dans la littérature:

- Approche connexionniste.
- Approche directe.

L'approche connexionniste consiste à combiner la théorie des réseaux de neurones avec celle de la logique floue afin de concevoir ce qu'on appelle un réseau neuronal flou (FNN: Fuzzy Neural Network). Deux manières pour construire un FNN [5]:

1. Soit en construisant un système d'inférence floue (SIF) sous forme d'un réseau de neurones multicouches dans lequel les poids du réseau correspondent aux paramètres du système flou (approche structurelle).
2. Soit en construisant un réseau neuronal qui peut incorporer le processus de raisonnement approximatif (approche fonctionnelle).

La rétropropagation est l'algorithme le plus utilisé pour l'apprentissage et l'ajustement des poids synaptiques du réseau caractérisant la base de connaissances floues. L'inconvénient majeur de cette méthode est le temps de calcul et le problème de convergence.

Dans l'approche directe le problème de conception et résolu par l'application des méthodes d'optimisation pour l'ajustement des paramètres du SIF,  $i$ ,  $e$ , sa base de connaissances.

Les algorithmes génétiques (AG), algorithmes itératifs inspirés des lois d'évolution des espèces et de la génétique naturelle, développés par J. Holland au début des années 70 [6], sont devenus des concurrents très puissants dans le domaine de l'optimisation des paramètres nécessaires à la conception des systèmes flous optimaux. Ainsi l'objectif est de combiner ces deux méthodes afin d'exploiter les richesses de chacune pour obtenir des systèmes aussi simples et efficaces que possibles.

L'objectif de ce travail est de développer un contrôleur *PID* dans une forme améliorée, où les coefficients de différentes actions  $K_P$ ,  $K_I$  et  $K_D$  sont non linéairement variables. Un superviseur flou à structure neuronale à apprentissage génétique permet de donner des fonctions optimales à ces coefficients. Ce manuscrit est divisé en deux parties principales :

- Une description de principe de fonctionnement des outils de conception : les SIF et les AG.
- Description de la méthodologie de conception de superviseur flou suivie par les résultats de simulation.

## LES ALGORITHMES GENETIQUES

Les AG commencent avec une population de créatures artificielles (chaînes de caractères) analogues aux chromosomes en nature qui évoluent par application de plusieurs opérateurs (modélisant le test de la survie naturelle) vers la solution optimale ou du moins vers une solution proche de celle-ci.

Une solution (un chromosome) résout ce problème, si elle satisfait une fonction de performances donnée au préalable. L'ensemble des solutions est appelé espace des solutions. L'objectif de la procédure de recherche est de localiser une solution optimale.

Une description abstraite de l'AG de base peut se traduire par la figure 1. En se basant sur le principe de fonctionnement des AG, on peut identifier quelques différences avec les méthodes classiques d'optimisation. Ces différences sont [7]:

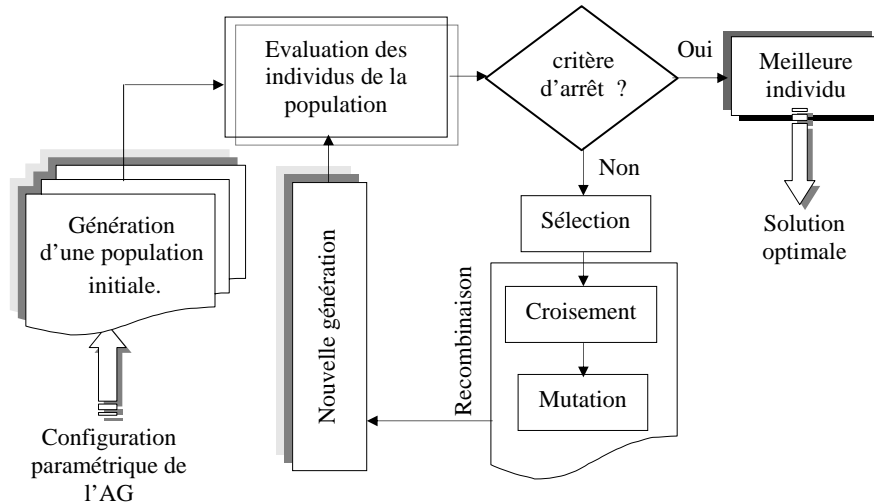
- Les AG possèdent une représentation codée et cherchent une représentation dans l'espace des solutions et non pas directement dans le domaine original.
- Les AG utilisent une population de solutions à la place d'une seule solution.
- Les AG utilisent des règles de transition probabilistes et non déterministes (pseudo- aléatoires) comme outils pour guider l'exploration à travers les régions de l'espace de recherche.
- Les AG n'utilisent que les valeurs de la fonction à optimiser, pas sa dérivée ou une autre connaissance auxiliaire.

### Mécanismes d'un AG standard

A partir d'une population initiale de chromosomes (chaînes de caractères) créée aléatoirement, l'AG génère de nouveaux chromosomes pour construire une nouvelle génération, en effectuant les opérations génétiques: *Sélection et reproduction*.

#### La sélection

Cette opération permet aux chromosomes codant de bonnes structures de se reproduire le plus souvent que ceux qu'ils ne le sont pas [7]. La plus utilisée est celle de la sélection proportionnelle (sélection par roue de fortune biaisée).



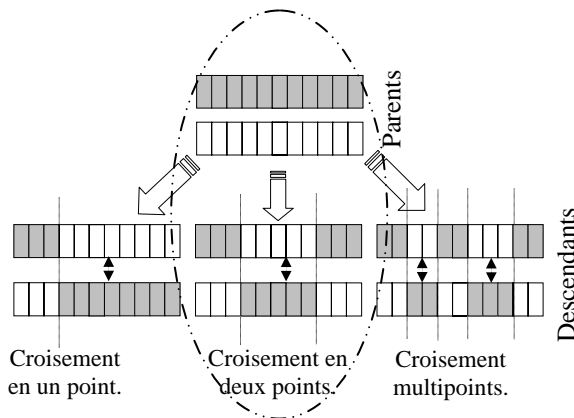
**Figure 1.** Le processus d'exécution de l'AG.

### La reproduction

Cette opération permet la construction d'une nouvelle génération, en appliquant les opérateurs génétiques à savoir le croisement et la mutation.

#### Croisement

L'opérateur de croisement consiste à échanger des gènes entre deux individus (parents) pour obtenir des descendants qui prennent les caractéristiques de leurs parents. Le site de croisement est choisi aléatoirement. La figure 2 schématise clairement les différents types de croisements.



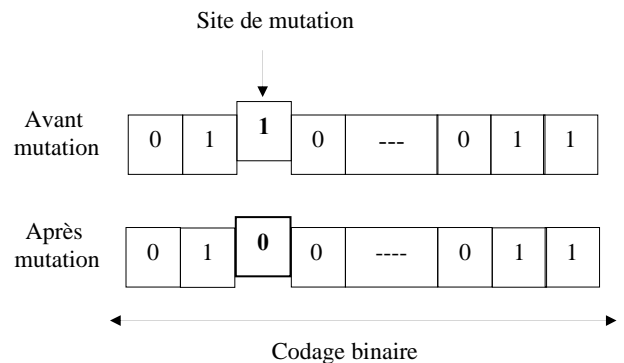
**Figure 2.** Croisement.

#### Mutation

L'opérateur de mutation consiste à altérer un ou plusieurs gènes choisis aléatoirement dans un chromosome. Cette altération va prendre des formes différentes selon le type de gènes :

- Une simple inversion pour un codage booléen (figure 3).
- Un tirage aléatoire dans un ensemble discret de valeurs de gènes (codage entier).
- L'ajout d'un bruit gaussien pour un codage réel.

Les opérateurs génétiques à savoir la sélection, la mutation et le croisement sont généralement suffisants pour accomplir la tâche d'optimisation. L'AG dans ce cas sera considéré comme une boîte noire d'optimisation de fonctions ne nécessitant aucune connaissance sur le domaine. Cependant, l'amélioration de l'AG standard et éventuellement la connaissance du domaine peuvent être introduites et exploitées pour améliorer les performances de l'AG.



**Figure 3.** Mutation.

### Notion du codage

La première étape dans le fonctionnement de l'algorithme génétique est le codage de l'ensemble des paramètres à optimiser en une chaîne de caractères de longueur finie. Pour une fonction (modèle d'un système) multivariable  $f(k_1, k_2, \dots, k_n)$ , le codage se déroule de la façon suivante :

- Coder chaque variable  $k_i$  ( $i = 1, \dots, n$ ) selon le choix de sa longueur.
- Construire les chaînes en concaténant les différents codes, c'est à dire juxtaposer ces derniers l'un après l'autre.

**Exemple:**

$$chaîne=(code\ 1)(code\ 2)\ \dots\ (code\ n). \tag{1}$$

Plusieurs types du codage sont utilisés [7]: Codage binaire, entier, réel et Codage Gray.

**Notion du décodage**

L'opération du décodage est l'inverse du codage. Chaque chaîne doit être décodée pour pouvoir calculer la valeur de la fonction d'aptitude qui lui est associée. Parmi les types du décodage possibles, le décodage binaire est le plus utilisé:

$$k=U_{MIN}+decimal(101\dots101_2)\frac{U_{MAX}-U_{MIN}}{2^l-1} \tag{2}$$

Avec  $U_{MAX}$ ,  $U_{MIN}$  sont les bornes supérieure et inférieure du domaine de variation de la variable  $k$ , respectivement.  $decimal(101\dots101_2)$  sa valeur décimale convertie du binaire et  $l$  est la longueur de la chaîne correspondante au code de la variable  $k$ . Le choix de la longueur dépend essentiellement de la précision voulue sur la variable  $k$  qui est donnée par:

$$\Pi=\frac{U_{MAX}-U_{MIN}}{2^l-1} \tag{3}$$

Quand le domaine de variation des chaînes est inconnu, ainsi que dans le cas où la longueur de la chaîne dépasse 32 bits (selon la norme IEEE), il est conseillé d'utiliser le décodage réel, comme suit:

$$x=(sgn)2^{exp}(mant) \tag{4}$$

Où ( $sgn$ ) est le signe, ( $exp$ ): l'exposant (sur 08 bits, après un décodage de 08 bits en base 10, le résultat est retranché à 127, ce qui donne l'exposant) et ( $Mant$ ): est la mantisse (sur 23 bits).

**SYSTEME D'INFERENCE FLOUE**

La structure de base d'un SIF est schématisée par la figure 4. Un SIF peut être décrit par un ensemble de règles linguistiques de la forme :

**SI** (condition(s)) **ALORS** (action(s)).

Ces règles permettent de convertir la stratégie de contrôle d'un expert humain en une stratégie de contrôle automatique bien adaptée au monde réel. Un SIF est composé de quatre blocs :

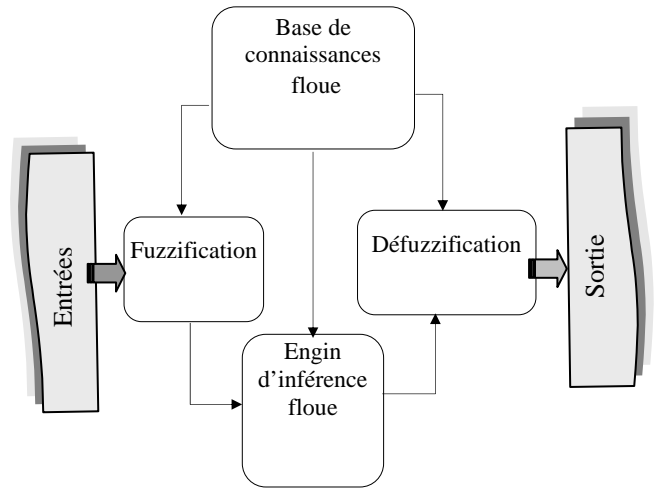
- Une base de connaissances.
- Une interface de fuzzification en entrée.
- L'engin d'inférence ou la logique de prise de décision.
- Une interface de défuzzification en sortie.

La *base de connaissances floue*, constitue le noyau du SIF. Elle contient les définitions des fonctions d'appartenance des variables d'Entrées/Sorties (gaussienne, trapézoïdale, etc.) ainsi que les règles d'inférence.

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

L'*engin d'inférence* permet de définir la stratégie de contrôle en utilisant les implications floues qui lient les différentes variables de chaque règle. Parmi les mécanismes d'inférence les plus utilisés: *MAX-MIN*, *MAX-PROD* et *SOM-PROD*.

L'opération de *défuzzification* consiste à convertir le résultat de combinaisons des règles en une forme numérique bien adaptée au milieu du processus à gérer. Plusieurs méthodes de défuzzification sont proposées dans la littérature [8]. La plus utilisée est celle du centre de gravité.



**Figure 4.** Structure de base d'un SIF.

Le problème de conception d'un SIF revient à la détermination de sa structure et aux choix de ces paramètres (phase d'acquisition). La conception d'un SIF nécessite le choix des paramètres suivants:

- Définition des variables d'Entrées/Sorties (E/S) du système.
- Choix de la partition floue, c'est à dire, association d'un ensemble de termes linguistiques caractérisés par des fonctions d'appartenance définies sur des univers de discours appropriés pour chaque variable d'E/S.
- La forme des fonctions d'appartenance de l'espace d'E/S (gaussienne, trapézoïdale, ...etc.).
- Les facteurs d'échelle (normalisation / dénormalisation)
- La base des règles.
- Le choix de la méthode d'inférence et de la stratégie de défuzzification.

Les systèmes flous ont connus un succès considérable dans la commande et l'identification des procédés industriels complexes, du fait de leurs caractères approximatif et qualitatif inspirés de la pensée humaine. Cependant, leurs performances sont liées à deux facteurs importants:

1. La disponibilité de l'expertise (le savoir-faire).
2. La validité des techniques d'acquisition de connaissances et la justesse des données acquises.

L'inconvénient majeur de l'approche traditionnelle pour la conception des SIF réside au niveau de l'obtention des connaissances auprès d'un expert humain. Pour palier ce problème, la recherche s'est orientée vers l'élaboration des techniques optimales et systématiques pour la conception de ces systèmes.

## CONCEPTION DE SUPERVISEUR FLOU

Les modèles les plus rencontrés dans la littérature des SIF, sont:

- Les modèles de Mamdani.
- Les modèles de Sugeno.

La différence, entre ces deux raisonnements, réside dans la forme d'évaluation des conséquences des règles floues. Dans la méthode de Mamdani [9-10], chaque règle est représentée par une conjonction (*MIN*) et l'agrégation des règles par une disjonction (*MAX*). Dans ce modèle, les conséquences sont indépendantes des entrées du SIF. Le lien entre les conditions et les actions est réalisé, uniquement, par des opérateurs.

Takagi-Sugeno et Kang [10] ont proposé un modèle pour développer une approche systématique pour générer les règles floues à partir de l'ensemble de données d'E/S. Une règle de ce type a la forme suivante:

$$R_i: SI (x_1 \text{ est } A_{1i}) ET \dots ET (x_m \text{ est } A_{mi}) ALORS \\ Y_i = F_i(x_1, x_2, \dots, x_m) \quad (4)$$

Les  $A_{mi}$  caractérisent les sous ensembles flous des entrées et les  $x_i$  représentent les variables d'entrées. Généralement  $F_i(-)$  possède une forme polynomiale de  $x_1, x_2, \dots$  et  $x_m$ .

$$F_i(x_1, \dots, x_m) = a_{0i} + a_{1i}x_1 + \dots + a_{mi}x_m \quad (5)$$

Cette forme est appelée le modèle de Sugeno d'ordre un. Dans certains cas,  $F_i(-)$  doit être constante (Sugeno d'ordre zéro). La détermination des coefficients des conséquences doit être faite par l'emploi des méthodes d'identification paramétriques [12].

Les méthodes de défuzzification adaptées à cette forme de règles sont basées sur l'approche : *Défuzzifier Sans Combiner (DSC)*. La plus utilisée est celle des moyennes pondérées "weighted-average defuzzification", donnée par:

$$y = \frac{\sum_{i=1}^M \mu_i F_i(x_1, \dots, x_m)}{\sum_{i=1}^M \mu_i} \quad (6)$$

Où  $M$  est le nombre des règles et  $\mu_i$  est le degré d'activation de la  $i^{\text{ème}}$  règle calculé soit par la conjonction *MIN* ou le produit arithmétique:

$$\mu_i = \prod_{j=1}^m \mu_i(x_j) \quad \text{Or} \\ \mu_i = \text{MIN}(\mu_i(x_1), \dots, \mu_i(x_m)) \quad (7)$$

La simplicité et le temps d'exécution réduit présentent deux avantages de l'approche de Sugeno. Plusieurs chercheurs [13-14] ont essayé de développer de nouvelles

méthodes permettant de réduire le nombre de paramètres des conséquences qui alourdissent l'approche de Sugeno.

Hao Ying [15] a proposé une méthode analytique de traitement de ce type de SIF. Les polynômes de sorties sont proportionnels l'un de l'autre. En introduisant certaines modifications au niveau des formes des règles, on obtient:

$$R_i : SI (x_1 \text{ est } A_{1i}) ET \dots ET (x_m \text{ est } A_{im}) ALORS \\ S_i = R_i F(x_1, \dots, x_m) \quad (8)$$

Avec,

$$F(x_1, \dots, x_m) = a_0 + a_1 x_1 + \dots + a_m x_m \quad (9)$$

En utilisant la méthode de défuzzification des moyennes pondérées, on obtient la sortie numérique  $S$  pour un ensemble de variables d'entrées  $x_1, x_2, \dots$  et  $x_m$ :

$$S = \frac{\sum_{i=1}^M \mu_i F_i}{\sum_{i=1}^M \mu_i} = \frac{\sum_{i=1}^M \mu_i R_i}{\sum_{i=1}^M \mu_i} \cdot F(x_1, \dots, x_m) \quad (10)$$

Les  $R_i$  sont les centres de gravité des ensembles flous de sortie associés à chaque règle. Les floues de base sont construites de telle sorte que  $\sum_{i=1}^M \mu_i \neq 0$ .

L'expression (10) devient:

$$S = G(x, \mu) \cdot F(x) \quad (11)$$

avec,

$$G(x, \mu) = \frac{\sum_{i=1}^M \mu_i R_i}{\sum_{i=1}^M \mu_i} \quad (12)$$

La relation  $a_0 + a_1 x_1 + \dots + a_m x_m$  présente une association d'un ensemble de contrôleurs linéaires, y compris le *PID (Proportional-Derivative-Integral)* comme cas particulier (avec  $a_0 = 0$ ). Les trois entrées utilisées par le *PID* dans sa forme discrète à l'instant d'échantillonnage  $n$  sont données par:

$$x_1(n) = y_r(n) - y(n) \quad ; \quad x_2(n) = x_1(n) - x_1(n-1) \quad ; \\ x_3(n) = x_2(n) - x_2(n-1)$$

Où  $y_r(n)$  et  $y(n)$  sont le signal de référence et l'état actuel du système à commander à l'instant d'échantillonnage  $n$ , respectivement.

Selon les données du problème à traiter, la variable de sortie peut être l'action directe ou sa variation. Il est parfois judicieux de formuler les règles d'inférence de sorte que le signal de sortie intervient comme incrément du signal d'action entre les instants  $n$  et  $n-1$ :

$$S(n) = S(n-1) + \Delta S(n) \quad (13)$$

Avec bien sûr évitement des régions de saturation:

$$S_{\min} \leq S(n) \leq S_{\max} \quad (14)$$

Dans le reste de travail, on note  $S(n)$  et  $\Delta S(n)$ , l'action directe et sa variation à l'instant d'échantillonnage  $n$ , respectivement.

La forme de sortie du SIF sera donnée par:

$$S(n) = G(x, \mu)F(x) = G(x, \mu)(K_{OP}x_1(n) + K_{OD}x_2(n)) \\ = (G(-)K_{OP})x_1(n) + (G(-)K_{OD})x_2(n) \quad (15)$$

La loi de commande par *PD* dans sa forme discrète est donnée par:

$$S_{PD}(n) = \overline{K_P}x_1(n) + \overline{K_D}x_2(n) \quad (16)$$

Où  $\overline{K_P}$  et  $\overline{K_D}$  sont les gains (coefficients) de proportionnalité et de dérivation, respectivement. Ceci rend le SIF équivalent à un *PD* avec des gains  $\overline{K_P}$  et  $\overline{K_D}$  variables et non linéaires, puisque la sortie d'un SIF est généralement non linéaire par rapport à ces entrées.

$$\overline{K_P} \rightarrow G(-)K_{OP} ; \quad \overline{K_D} \rightarrow G(-)K_{OD}$$

Dans le cas de la variation de l'action, la sortie du SIF sera donc:

$$\Delta S(n) = G(-)(a_1x_1(n) + a_2x_2(n) + a_3x_3(n)) \quad (17)$$

Par analogie avec un *PI* et un *PID* dans la forme discrète, la variation de sortie est:

$$\Delta S_{PI}(n) = \overline{K_P}x_2(n) + \overline{K_I}x_1(n) \quad (18)$$

$$\Delta S_{PID}(n) = \overline{K_P}x_2(n) + \overline{K_I}x_1(n) + \overline{K_D}x_3(n) \quad (19)$$

La relation (17) peut s'écrire comme suit:

$$\Delta S(n) = (a_1G(-))x_1(n) + (a_2G(-))x_2(n) + (a_3G(-))x_3(n) \quad (20)$$

Cette dernière est similaire à celle de l'expression (19), seulement les gains constants  $\overline{K_P}$ ,  $\overline{K_I}$  et  $\overline{K_D}$  sont remplacés par des facteurs non linéaires:  $(a_2G(-))$ ,  $(a_1G(-))$  et  $(a_3G(-))$ , respectivement.

Généralement, la commande des systèmes complexes mal définis par un *PID* présente une tâche délicate. La solution du problème revient à choisir ou à développer une méthode efficace pour la détermination des gains de différentes actions, ou la combinaison du *PID* avec d'autres méthodes plus sophistiquées bien adaptées à la complexité et à la nature des procédés industriels.

Le superviseur ainsi développé représente une façon de cette hybridation, dont le but est de profiter d'une part de la précision du modèle mathématique du *PID*, d'autre part, de la souplesse des éléments manipulés par la logique floue.

## STRUCTURE DU RESEAU SUPERVISEUR

La structure d'un réseau de neurones adaptée à la forme des règles de l'expression (8) est schématisée par la figure 5. Elle présente un SIF à deux entrées  $x_1$ ,  $x_2$ , une sortie  $S$ , trois sous ensembles flous pour chacune et neuf règles. Il s'agit donc d'un contrôleur du type 333 selon Hamaifar [16] et Jim Woo Kim [17-18].

Si  $F(x)$  de l'expression (9) est constante,  $F(x) = a_0$ , le superviseur exploite efficacement le raisonnement approximatif de Sugeno d'ordre zéro. Dans le cas où  $F(x)$  dépend des entrées du SIF (Sugeno d'ordre un modifié), celle-ci n'est qu'une hybridation *FLOU-PID* où les

paramètres du système (base de connaissances floues) sont injectés sur un réseau de neurones multicouches.

Le réseau utilisé possède 06 couches :

- Une couche de normalisation des grandeurs d'entrées.
- Une couche d'entrée réalise le transfert direct des signaux d'entrées.
- Une couche de fuzzification réalise la conversion des données numériques en une forme symbolique bien adaptée au raisonnement approximatif. La fonction d'activation associée à chaque neurone de cette couche est la fonction triangulaire (figure 5). La sortie de chacun est caractérisée par l'expression (21).

$$Tri = \begin{cases} (x - x_{i-1}) / (x_i - x_{i-1}), & x \in [x_{i-1}, x_i], i = 1, 2, \dots, n \\ (x_{i+1} - x) / (x_{i+1} - x_i), & x \in [x_i, x_{i+1}], i = 1, 2, \dots, n-1 \\ 1, & x < x_1 \text{ or } x > x_n \end{cases} \quad (21)$$

Les poids entre la couche d'entrée et celle de fuzzification représentent les largeurs des sous ensembles flous associés à chaque variable linguistique.

- Une couche d'inférence où chaque neurone calcule le degré d'activation de chaque règle (utilisant un opérateur de produit arithmétique). La sortie d'un neurone de la couche d'inférence est donnée par:

$$O_i = (\mu_i(x_1) \times \mu_i(x_2)) \quad (22)$$

Les poids entre la couche d'inférence et celle de fuzzification sont fixés à 1.

- Une couche de défuzzification permettant de générer la sortie numérique. La méthode utilisée est celle des moyennes pondérées.

Les poids entre cette couche et celle d'inférence représentent les conséquences des règles, qui sont les centres des répartitions floues (singleton) de sortie de chaque règle.

- Une couche de dénormalisation permet de d'amplifier ou d'atténuer le signal d'action.

Les paramètres ou poids du réseau de neurones, ainsi que les coefficients  $K_{OP}$ ,  $K_{Ob}$ ,  $K_{OD}$  associés au gain de dénormalisation et les facteurs d'échelle en entrée  $K_{X1}$  et  $K_{X2}$  sont optimisés par l'AG.

## STRUCTURE DE LA COMMANDE

La structure de commande de la figure 7 contient quatre blocs:

- Un bloc d'optimisation simulé par l'AG.
- Un bloc de supervision représentant le SIF.
- Le système de commande (PID).
- Le système à commander (procédé).

L'organigramme de la figure 8 exprime clairement l'interaction entre ces blocs.

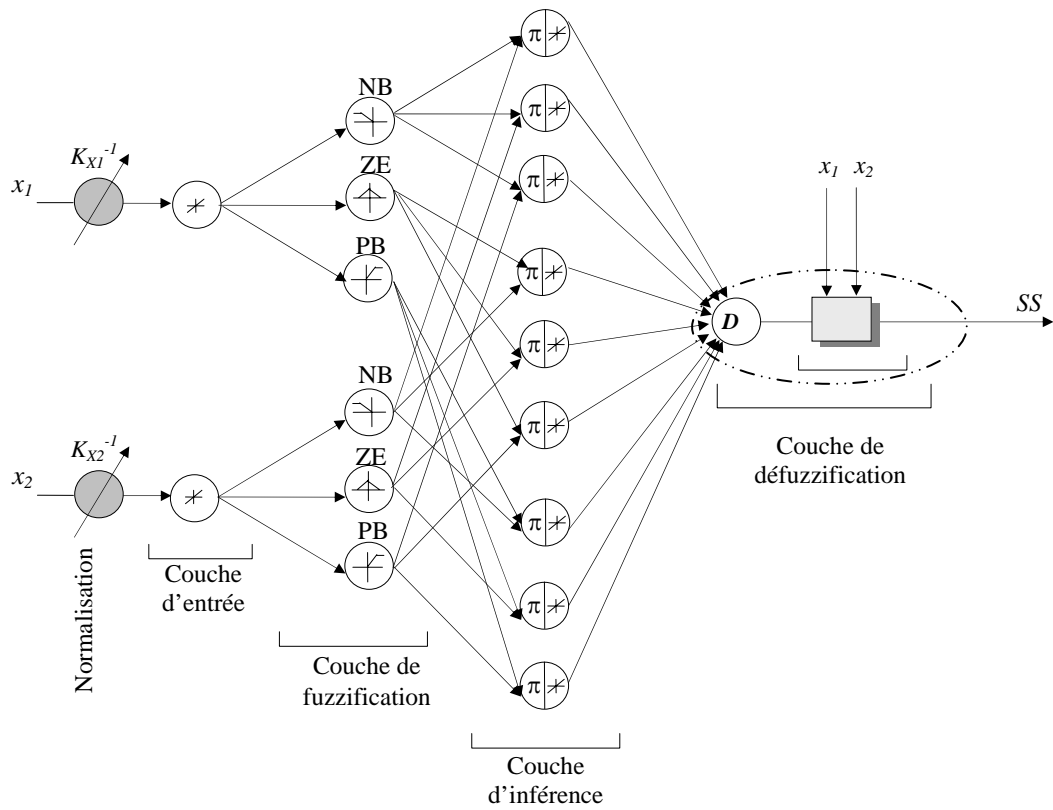


Figure 5. Structure du réseau superviseur.

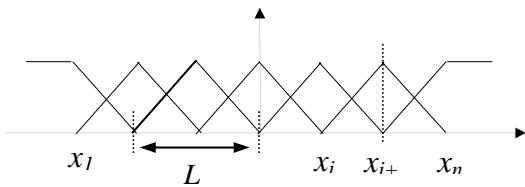


Figure 6. Fonctions d'appartenance triangulaires.

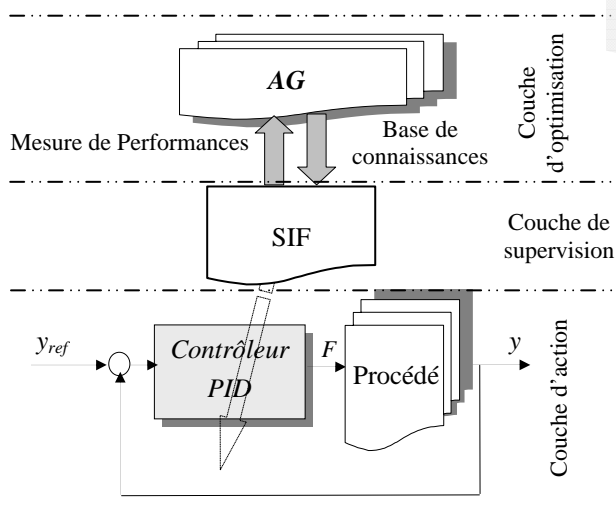


Figure 7. Structure de contrôle et d'apprentissage.

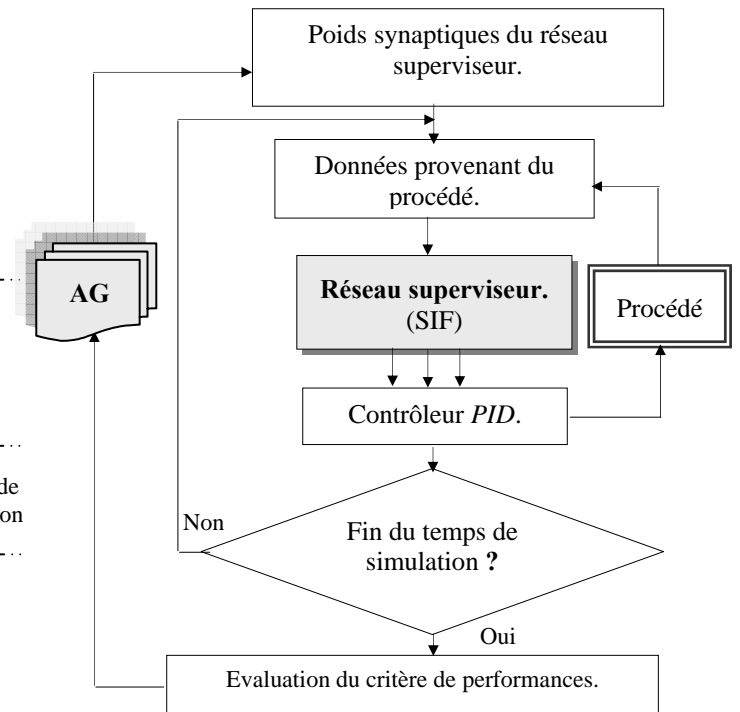


Figure 8. Interaction entre les différents blocs de la structure de commande.



**Remarques:**

- ☞ La mise en échelle (Normalisation / dénormalisation) des grandeurs d'E/S permet d'adapter le traitement des signaux d'E/S. Par convention, la plage de variation des variables d'E/S est comprise entre -1 et +1. Ces facteurs d'échelles sont adaptés, soit empiriquement, soit par l'emploi des méthodes de contrôle adaptatif.
- ☞ Les variables d'E/S du SIF sont utilisées avec normalisation, c'est à dire, utilisant le domaine de discours normalisé [-1, +1].
- ☞ Les fonctions d'appartenance utilisées sont de type triangulaires symétriques équidistantes avec un chevauchement de 25%.

**CODAGE DES PARAMETRES**

La différence entre les AG utilisés dans la littérature réside niveau du codage (binaire, entier ou réel) et les opérateurs génétiques adaptés à chaque mode du codage [19].

Le codage mixte binaire-entier multiparamétré et concaténé où plusieurs paramètres sont codés et juxtaposés l'un après l'autre, pour construire le chromosome de l'AG, est utilisé dans cette conception. Les paramètres à optimiser sont:

- Les largeurs des sous ensembles flous associés aux espaces d'E/S qui sont codées en binaire.
- Les règles ou centres des sous ensembles flous de sortie sont codées en entier. Chaque centre (singleton) est codé par un numéro allant de 1 jusqu'au nombre maximal N des sous-ensembles flous de l'espace de sortie.
- Les coefficients  $K_{X1}$ ,  $K_{X2}$ ,  $K_{OP}$  et  $K_{OD}$  sont les facteurs d'échelle des variables d'E/S codés en binaire.

Chaque chromosome peut représenter une solution du problème, c'est à dire, une base de connaissances floues optimale. Sa structure est représentée par la figure 9.  $L_{ZE\ X1}$ ,  $L_{ZE\ X2}$  et  $L_{ZE\ S}$  sont les codes des largeurs des univers de discours ou des sous ensembles flous (ZE) des entrées  $x_1, x_2$  et de la sortie  $S$ , respectivement. Les  $R_i$  sont les codes des conclusions caractérisant les centres des sous ensembles flous de sortie,  $K_{X1}$ ,  $K_{X2}$ ,  $K_{OP}$  et  $K_{OD}$  sont les codes des coefficients des facteurs d'échelle des variables d'E/S.

**METHODE D'OPTIMISATION**

Les grandeurs ajustables sont les largeurs des sous-ensembles flous, les règles et les facteurs d'échelle des variables d'E/S. On commence par l'optimisation de  $L_{ZE}$ , c'est à dire, déterminer les bornes de variations supérieure (UB) et inférieure (LB) de ZE. Après l'opération de décodage, on aura:

$$L_{ZE} = LB + decimal(01101.....01_2) \frac{UB-LB}{2^l-1} \quad (23)$$

Où  $l$  représente la longueur ou la taille du code de ZE. Les largeurs des autres sous-ensembles flous sont obtenues par un décalage avec un chevauchement de 25%. De même pour les autres paramètres codés en binaire.

**MECANISME DE L'AG**

La sélection par roue de fortune pour la reproduction, l'opérateur de croisement en deux points et la mutation sont les étapes essentielles de l'AG utilisé dans ce travail.

Si la mutation, a lieu dans la partie codée en binaire, l'opérateur de mutation a le même principe que celui de la mutation standard. Si la mutation a lieu dans la partie des entiers, au lieu de changer un 1 en 0 ou un 0 en 1, on change l'allèle (l'entier) par une valeur aléatoire de 1 à N différente de sa valeur courante. N représente le nombre des sous ensembles flous de sortie. La figure 10 schématise clairement le processus de mutation.

Enfin, pour mettre une limitation à la divergence génétique, une des stratégies d'élitisme a été introduite. Cette dernière, basée sur la technique de sélection à l'état permanent "steady-state selection" permet de construire une nouvelle génération plus performante que la précédente.

De plus, et 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 [20] ( $N_{gen}$ : le numéro de génération). La figure 11 indique clairement le processus de l'évolution des opérateurs génétiques, croisement et mutation.

$$P_c = \exp\left(\frac{-N_{gen}}{Maxgen}\right) \quad (24)$$

$$P_m = \exp\left(\frac{0.05 * N_{gen}}{Maxgen}\right) - 0.8 \quad (25)$$

Le critère d'arrêt de l'AG est fixé par un nombre bien déterminé de générations ( $Maxgen$ ).

La validité de la méthodologie de supervision proposée, est testée, sur le système du pendule inversé considéré comme un système, non linéaire, mono-entrée/multi-sorties (SIMO).

**SIMULATION**

La figure 12 montre le diagramme schématique du système de pendule inversé, ainsi que sa structure.

Les dynamiques du système sont caractérisées par:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{g \sin(x_1) + \cos(x_1) \left( \frac{-U - m_p l x_2^2 \sin(x_1)}{m_c + m_p} \right)}{\left( \frac{4}{3} \frac{m_p \cos^2(x_1)}{m_c + m_p} \right)} \end{aligned} \quad (26)$$

Où,  $l$ : demi longueur du segment,  $g$ : accélération gravitationnelle,  $m_c$ : masse du chariot,  $m_p$ : masse du segment,  $x_1 = \theta$  l'angle du segment par rapport à l'axe vertical et  $x_2 = \dot{\theta}$  la vitesse angulaire du segment. Les valeurs numériques utilisées dans la simulation sont:  $l = 0.5m$ ,  $m_c = 1kg$ ,  $m_p = 0.1kg$ ,  $g = 9.8m/s^2$ .

L'objectif du contrôle est de générer une séquence d'actions  $F$  (en Newton) pour stabiliser le pendule dans sa position d'équilibre (verticale) correspondante à un angle et

une vitesse angulaire nulles. Le véhicule se déplace librement à droite ou à gauche avec l'hypothèse que les frottements sont négligeables. Ce système est simulé par la méthode de Runge-Kutta avec une période d'échantillonnage égale à 10 ms.

Le but de l'AG est de générer une base de connaissances optimale du superviseur flou en minimisant une fonction coût donnée. Dans cette application la fonction objective responsable de classement des chromosomes dans la population est:

$$F_{it} = \left( 1 + \sum_{k=1}^{400} (10^{-3}|\theta(k)| + 10^{-4}|F(k)|) \right)^{-1} \quad (27)$$

Le superviseur ainsi développé pour cette application se caractérise par: Deux SIF à deux bases de connaissances différentes, une pour le gain de l'action de proportionnalité  $K_{PPD}$  et l'autre pour l'action de dérivation  $K_{DPD}$ , comme montre la figure 13.

Chaque SIF est caractérisé par :

- 05 sous ensembles flous pour l'angle,
- 05 sous ensembles flous pour la vitesse angulaire,
- 05 sous ensembles flous pour la commande (force).

Ce qui donne au total 25x2 règles.

Chaque chaîne de la population contiendra 120 gènes (10 bits pour chacune des largeurs des univers de discours, 25x2 entiers pour les règles et 10 bits pour chaque coefficient des gains).

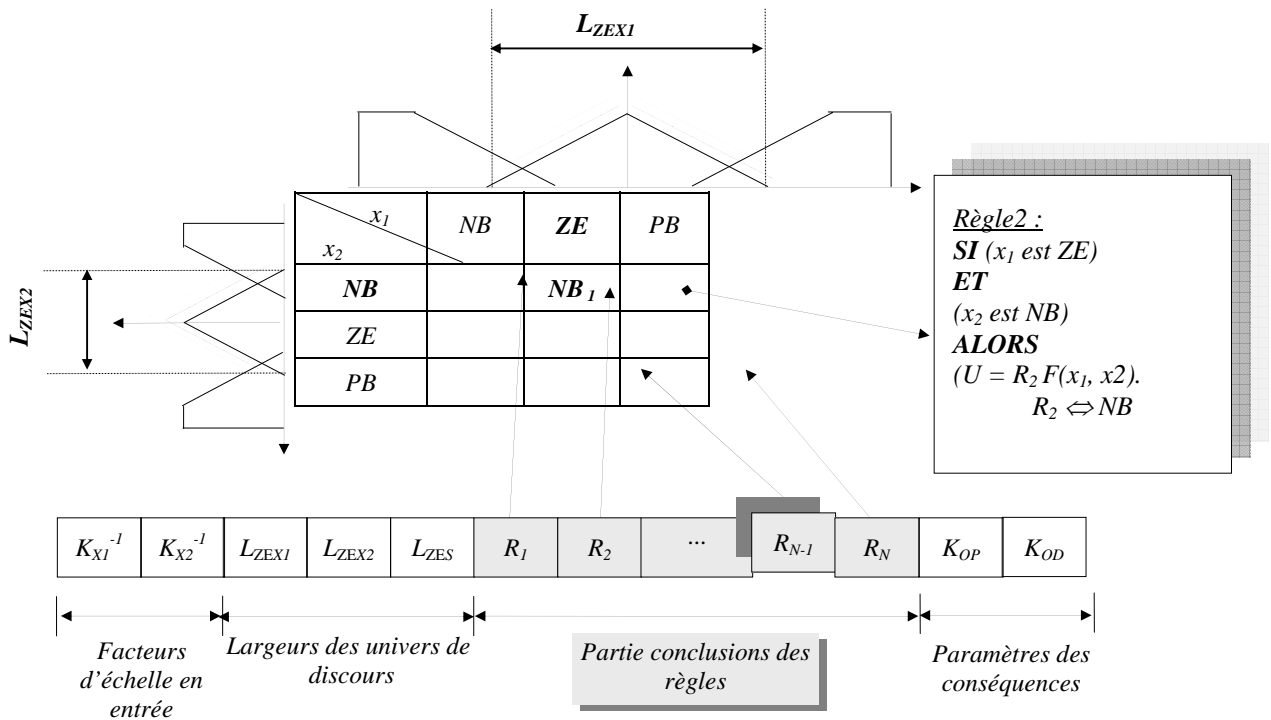


Figure 9. Structure de chromosome.

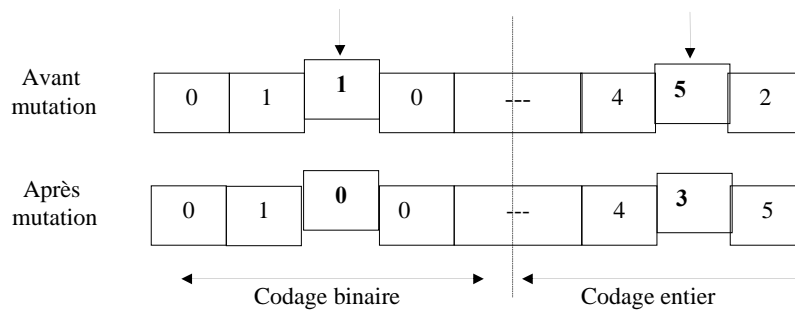
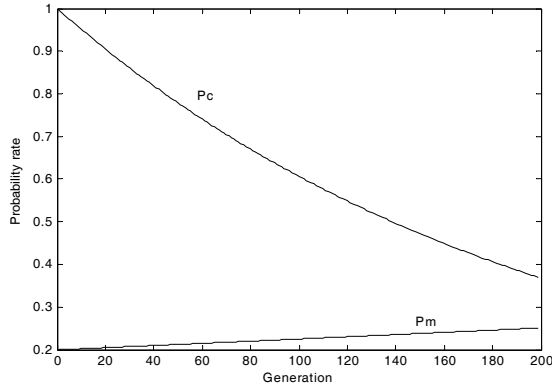


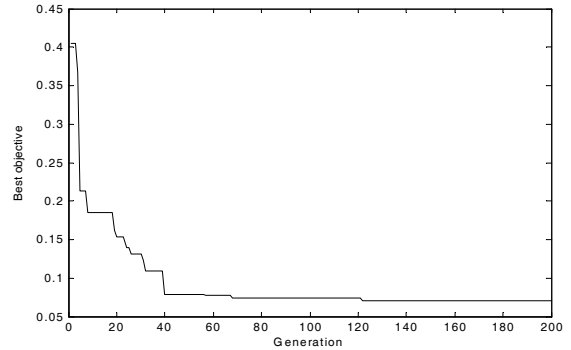
Figure 10. Mutation.



**Figure 11.** Evolution des probabilités Pm et Pc.

Les conditions initiales utilisées dans cette simulation sont:  $\theta_i=20^0$  et  $\dot{\theta}_i=0^0/sec$ , avec la contrainte:  $|U| \leq 125[N]$ .

La figure 14 indique l'évolution de la fonction objective de génération en génération.



**Figure 14.** Evolution de la fonction objective.

Les paramètres de l'AG sont présentés dans le tableau 1.

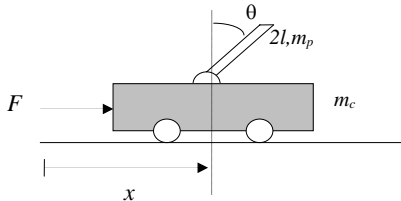
**Tableau 1.** Paramètres de l'AG.

Paramètre	Valeur
La taille de la population.	40
Les largeurs $L_{ZEX1}$ , $L_{ZEX2}$ et $L_{ZEU}$	[-0.5, 0.5]
Les paramètres $K_{X1}$ et $K_{X2}$	[10, 80]
Les paramètres $K_{OP}$ et $K_{OD}$	[1, 150]
Le nombre maximal de génération.	200

Le tableau 2 représente les bases de règles générées après 200 générations. La figure 15 montre, bien, la convergence de la trajectoire vers l'état d'équilibre ( $0^0$ ,  $0^0/s$ ) à partir du l'état initial ( $20^0$ ,  $0^0/s$ ).

**Tableau 2.** Tables des règles.

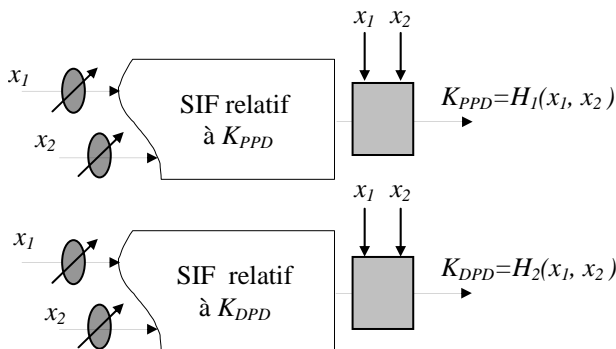
Base de règles	
<b>SIF 1 :</b>	<b>2525251344135523412112345</b>
<b>SIF 2 :</b>	<b>5343124233323234211242513</b>



**Figure 12.** Structure de pendule inversé

La robustesse du contrôleur est justifiée par sa capacité de bien réagir lorsqu'on s'écarte des conditions normales d'utilisation. Pour cela, ce dernier a été testé pour plusieurs conditions initiales, comme le montre la figure 16. Les conditions initiales testées sont les suivantes :

- (+10°,0°/s)(+20°,0°/s)(+30°,0°/s)(+40°,0°/s)
- (+ 50°,0°/s)(+ 60°,0°/s)(+ 70°,0°/s)(+ 80°,0°/s)



**Figure 13.** Structure de superviseur.

**CONCLUSION**

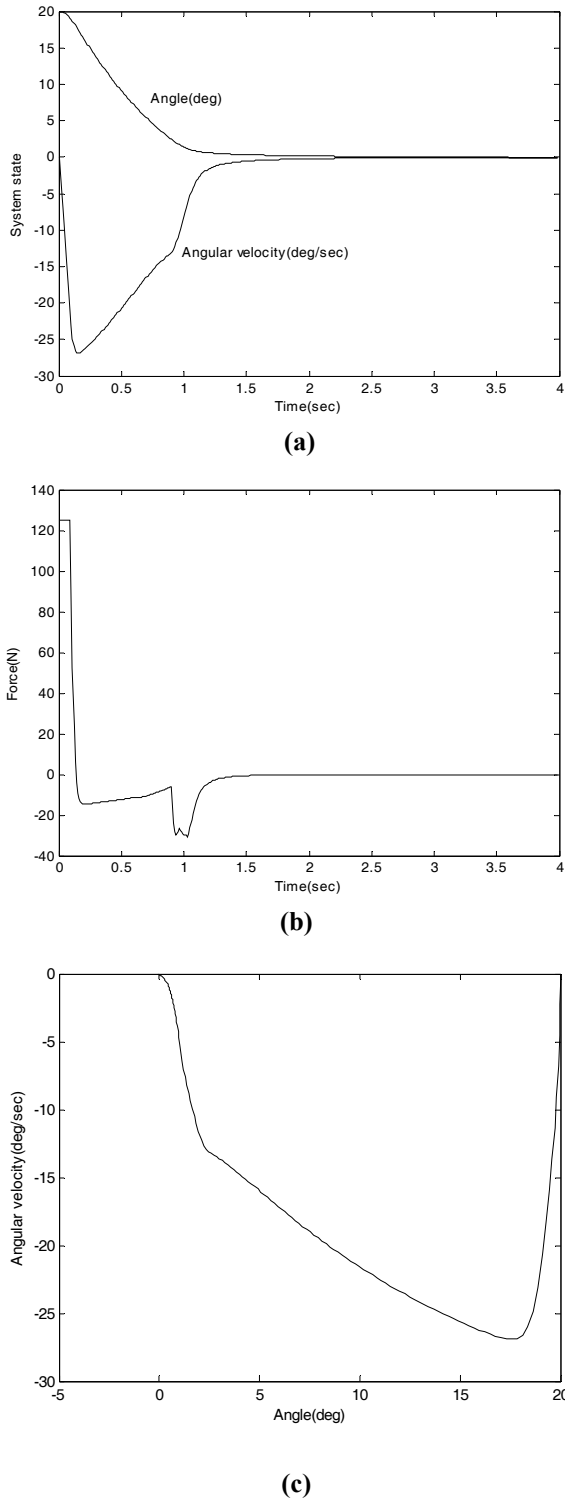
Dans ce travail, le contrôleur PD à supervision floue proposé a prouvé son efficacité. Sa robustesse est justifiée par sa capacité de réagir dans un sens positif (réalisation du but visé par la commande) vis à vis au changement des conditions de fonctionnement.

La mise en œuvre de régulateur PID souffre de problèmes de réglage des gains de différentes actions afin d'obtenir de bonne performances.

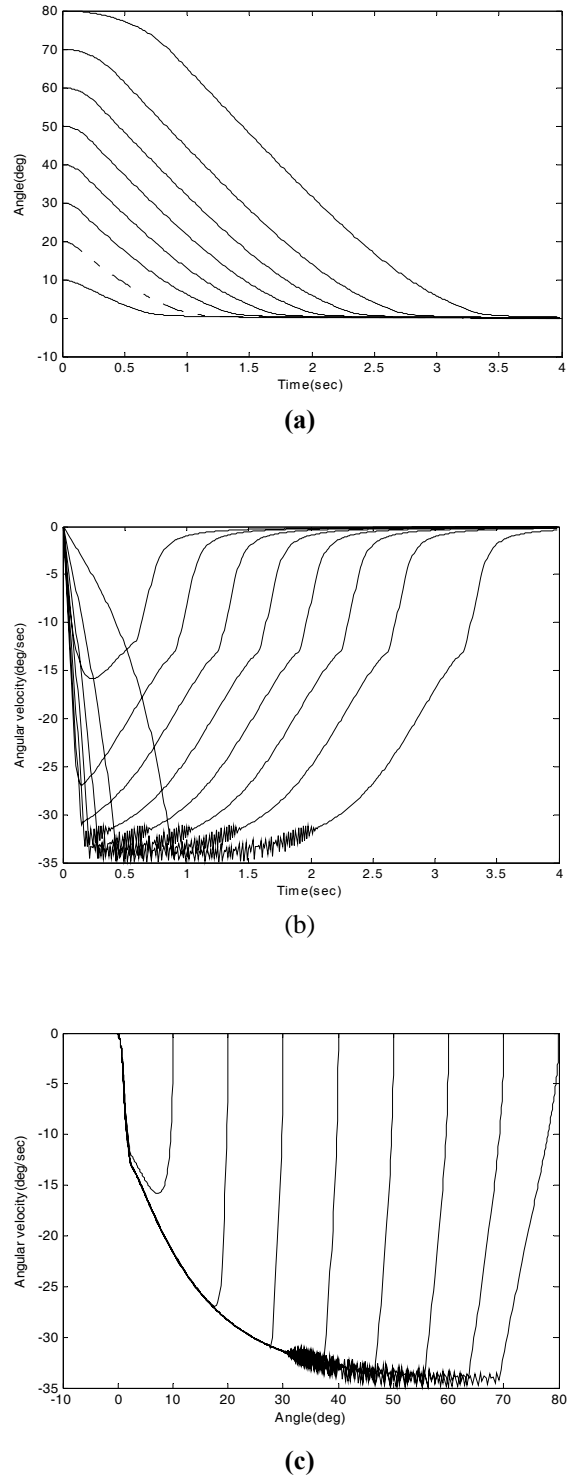
L'exploitation d'une supervision basée sur la logique floue à optimisation génétique en vue de l'adaptation des paramètres de synthèse d'un régulateur de type PID permet d'offrir une alternative fort intéressante dont l'objectif est de satisfaire les critères de performances désirées. La combinaison des techniques conventionnelles de l'automatique avec les techniques dites 'intelligentes' permet de tirer profit, d'une part des performances offertes par les algorithmes conventionnelles, et d'autre part de la

souplesse des éléments manipulés par les techniques modernes qui se traduit par leurs progressivités dans la prise de décisions.

On remarque que le contrôleur arrive toujours à réaliser l'objectif.



**Figure 15.** Réaction du pendule : (a) Etat du système. (b) Commande générée. (c) Plan de phase.



**Figure 16.** Réponses aux différentes conditions initiales. (a) Angles. (b) Vitesses angulaires. (c) plan de phase.

## REFERENCES

- [1] Yongho Lee and Sunwon Park, "PID controllers tuning for desired closed-loop responses for SI/SO systems", *Aiche journal*, Vol. 44, N° 4, January 1998, 106-115.
- [2] Aström, K.J. and T. Häggglund, "*PID Controllers: Theory, Design, and Tuning*", Instrument Society of America, 1995.

- [3] Marlin, T. E, "Process Control. Designing Processes and Control Systems for Dynamic Performance", Mc Graw-Hill, 1995.
- [4] L. A. Zadeh, "Fuzzy sets", *Information and Control*, Vol. 8, 1965, pp. 338-353.
- [5] Jhy-Shing Roger Jang and Chuen-Tsai Sun, "Neuro fuzzy modeling and control", *Proceedings of the IEEE*, Vol. 83, No. 3, March 1995, pp. 378-406.
- [6] David E. Goldberg, "Algorithmes génétiques: exploration, optimisation et apprentissage automatique", Addison-Wesley, 1994.
- [7] J. Holland, "Adaptation in natural and artificial systems", University of Michigan Press, 1975.
- [8] Yigang Shi and P.C. Sen, "A New Defuzzification Method for Fuzzy Control of Power Converters", *IEEE proceeding on Fuzzy Systems*, 2000.
- [9] Mamdani, E. H., "Application of fuzzy algorithms for simple dynamic plant", *Proc. IEE*, D-121, 1974, pp. 1558-1588.
- [10] Mamdani, E. H., and Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller", *International Journal of Man- Machine Studies*, Vol. 7, 1975, pp. 1-13.
- [11] M. Sugeno and G. T. Kang, "Structure identification of fuzzy model", *Fuzzy sets and systems*, Vol. 28, 1988, pp.15-33.
- [12] Jhy-Shing Roger Jang and Chuen-Tsai Sun, "Neuro fuzzy modeling and control", *Proceedings of the IEEE*, Vol. 83, No. 3, March 1995, pp. 378-406.
- [13] X. W. Yan, Z. D. Deng and Z. Q. Sun, "Genetic Takagi-Sugeno-fuzzy reinforcement learning", *Proceeding of the 2001 IEEE, International Symposium On Intelligent Control*, September 5-7, 2001, Mexico city, Mexico, pp. 67-72.
- [14] Chia-Feng Juang and Yuan-Chang Liou, "A TSK-type recurrent fuzzy network for dynamic systems processing via supervised and reinforcement learning", *2001 IEEE International Fuzzy Systems Conference*, pp. 240-243.
- [15] 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.
- [16] 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.
- [17] Jinwoo Kim, Yoonkeon Moon, and Bernard P. Zeigler, "Designing fuzzy net controllers using genetic algorithms", *IEEE control systems*, June 1995, pp. 66-72.
- [18] 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.
- [19] Sanjay Kumar Sharma and George W. Irwin, "Fuzzy Coding of Genetic Algorithms", *IEEE Transactions on Evolutionary Computation*, Vol. 7, No. 4, August 2003, pp 344-355.
- [20] Teo Lian Seng et al., "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.