

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE ABDERRAHMANE MIRA DE BEJAIA
FACULTE DE LA TECHNOLOGIE

Département d'Electronique

MEMOIRE

Présenté en vu de l'obtention du Diplôme de Magister

en : Automatique et Traitement du Signal

Option : Systèmes

Par : Mouloud IDER

Ingénieur d'Etat en Electronique, année 2006

Thème :

**Etude des systèmes flous *type-2* : Application à
la commande des systèmes non linéaires.**

Jury :

K. MOKRANI,	Maitre de Conférences, Université de Béjaia	Président
B. MENDIL,	Professeur, Université de Béjaia	Rapporteur
R. ALKAMA,	Maitre de Conférences, Université de Béjaia	Examineur
S. BERRAH,	Maitre de Conférences, Université de Béjaia	Examineur

Remerciements

Ce présent travail est réalisé sous la direction du Pr. B. MENDIL, qu'il trouve ici ma profonde reconnaissance et mes sincères remerciements.

Je tiens à remercier le président du jury pour avoir accepté de présider le jury.

Je remercie aussi les examinateurs pour l'honneur qu'ils nous font en participant à l'évaluation de ce travail.

Sommaire

Introduction générale	1
------------------------------------	---

Chapitre 1 : Logique et systèmes flous

1. Introduction.....	3
2. La logique floue <i>type-1</i>	3
2.1. Principes et définition.....	3
2.2. Opérateurs et normes.....	4
2.2.1. <i>T</i> - norme : Intersection floue.....	5
2.2.2. <i>T</i> -conorme : Union floue.....	5
2.3. Logique floue.....	5
3. Modèles de contrôleurs flous <i>type-1</i>	7
3.1. Modèle flou linguistique.....	7
3.1.1. Fuzzification.....	7
3.1.2. Base de règles.....	8
3.1.3. Mécanisme d'inférence.....	9
3.1.4. Défuzzification.....	9
3.2. Modèle flou type Takagi- Sugeno-Kang (<i>TSK</i>).....	10
4. Concept des systèmes flous <i>type-2</i>	10

Chapitre 2 : Représentation des systèmes flous *type-2*

1. Introduction.....	14
2. Opérations sur les ensembles flous <i>type-2</i>	14
2.1. Opérations <i>meet</i> sous la t-norme <i>produit</i>	15
2.2. Opérations algébriques.....	17

3. Systèmes et contrôleurs flous <i>type-2</i>	18
3.1. Base de règles.....	18
3.2. Inférence floue.....	19
3.3. Traitement de sortie.....	19
4. Réduction de type des ensembles flous <i>type-2</i>	20
4.1. Réduction de type par la méthode du centroïde.....	22
4.2. Réduction de type par la méthode des centres d'ensembles.....	23
4.3. Interprétation de l'ensemble type réduit.....	24
5. Réduction de type des ensembles flous <i>type-2</i> intervalle.....	25
5.1. Réduction de type par la méthode du centroïde.....	27
5.2. Réduction de type par la méthode des centres d'ensembles.....	27
6. Conclusion.....	30

Chapitre 3: Application des systèmes flous *type-1* et *type-2* à la commande des systèmes non linéaires.

1. Introduction.....	32
2. Commande non linéaire floue.....	33
3. Etude des incertitudes numériques.....	33
3.1. Contexte et objectif.....	33
3.2. Critères de performance.....	34
3.3. Simulation et résultats.....	35
4. Etude des incertitudes linguistiques.....	39
4.1. Modélisation et objectif de la commande.....	40
4.2. Analyse de la stabilité et conception des FLC.....	41
4.3. Résultats de simulation.....	43

5. Conclusion.....	46
--------------------	----

**Chapitre 4: Optimisation des contrôleurs flous *type-1* et *type-2* par les
AG : Application à la commande d'un robot mobile.**

1. Introduction.....	47
2. La robotique mobile.....	48
2.1. Introduction.....	48
2.2. La logique floue en robotique mobile.....	48
2.3. Modélisation.....	49
3. Les algorithmes génétiques.....	51
4. Navigation à base des FLC <i>type-1</i> et <i>type-2</i> : réglage heuristique.....	56
5. Optimisation des FLC par les <i>AG</i>	60
6. Conclusion.....	67
Conclusion générale.....	68

Bibliographies.

Annexes.

Introduction générale

Introduction générale.

La logique, au sens commun du mot, est une conception des mécanismes de la pensée qui ne devrait jamais être floue, toujours rigoureuse et formelle. Mais les Mathématiciens se sont aperçus qu'il n'y avait pas, en réalité, une logique unique, mais autant de logiques qu'on veut ; tout étant dépendant de l'axiomatique choisie [1] [14].

La logique floue a pour objet d'études la représentation des connaissances imprécises et le raisonnement approché. On peut donc la situer à côté des heuristiques de résolutions de problèmes, des systèmes experts, de l'intelligence artificielle et même du traitement de la langue naturelle [2]. Cette théorie, introduite en 1965 par le professeur Lotfi Zadeh [1], permet de traiter des propositions ou des états par plusieurs niveaux de vérité.

Devant le délicat problème de modélisation des systèmes complexes, les méthodes de commande nécessitant une bonne connaissance du modèle mathématique s'avèrent incapables de répondre aux performances exigées. Afin de répondre à ces exigences toujours croissantes, les algorithmes de commande développés dans la littérature deviennent de plus en plus complexes. Alors, la commande floue s'est présentée, ces dernières années, comme l'une des alternatives.

La commande floue se base sur deux concepts essentiels à savoir : celui de la décomposition d'une plage de variation d'une variable sous formes de nuances linguistiques et des règles provenant de l'expertise de l'opérateur humain qui exprime sous forme linguistique comment doivent évoluer les commandes du système en fonction des variations de l'observation [14]. Un système décrit par des ensembles flous *type-1* est appelé système flou *type-1*. Il y a au moins trois sources d'incertitudes dans de tel système [4]:

- ✓ Le sens des mots utilisés dans les règles peut être incertain, c'est-à-dire qu'un mot peut prendre plusieurs sens.
- ✓ Les mesures activant un système flou *type-1* peuvent être bruitées et incertaines.
- ✓ Les données utilisées pour l'ajustement des paramètres d'un système flou *type-1* peuvent être aussi bruitées.

Nous devons distinguer alors entre deux types principaux d'incertitudes : aléatoires (numériques) et linguistiques [4][5]. La théorie des probabilités est utilisée pour modéliser l'incertitude aléatoire où la fonction de densité de probabilité (pdf) incarne la totalité des informations relative à cette dernière. Cependant, dans la plupart des applications pratiques, il est impossible de connaître ou de déterminer la pdf. Ainsi, nous serons obligés d'admettre le fait qu'une pdf sera complètement caractérisée en connaissant tous ses moments. L'utilisation du moment d'ordre 1, qui est la moyenne est insuffisant, parce que les incertitudes aléatoires exigent la connaissance de la dispersion autour de cette valeur moyenne, une information fournie par la variance [6]. Donc, un modèle probabiliste exige la connaissance d'au moins deux moments de la pdf.

Si nous projetons la même philosophie sur les systèmes flous, nous allons alors considérer la sortie d'un système flou *type-1* comme étant la valeur moyenne de la pdf, donc pour que le système flou prenne en compte les incertitudes, nous aurons alors besoin d'une certaine variance (dispersion) ; d'où la notion d'ensembles flous *type-2* où la variance est considérée comme étant le domaine d'appartenance secondaire de ces derniers. Les ensembles flous *type-2* ont donné naissance à la logique et aux systèmes flous *type-2*.

L'objectif de ce travail est l'étude des systèmes flous *type-2*, en vue d'application à la commande des systèmes non linéaires. Dans un premier temps, on montre l'avantage de cette approche par rapport à l'approche classique (logique floue *type-1*). Puis, nous allons procéder à l'optimisation des paramètres des contrôleurs flous *type-2* par un algorithme génétique. Le contrôleur hybride résultant sera appliqué à la commande d'un robot mobile et sa robustesse sera testée sous deux types de perturbations.

Le mémoire est composé de quatre chapitres. Le premier porte essentiellement sur la théorie et la structure des systèmes flous *type-1*. Le second expose les systèmes flous *type-2* (opérateurs, structure, réduction de type, ...). Une application des commandes floues *type-1 et type-2* sur des systèmes non linéaires, est présentée dans le chapitre 3.

Le dernier chapitre est consacré à l'optimisation des paramètres des contrôleurs flous étudiés par les algorithmes génétiques (AG). L'application est portée sur la robotique mobile.

Chapitre 1

Logique et systèmes flous.

1. Introduction

L'objectif principal de ce premier chapitre est l'introduction du concept de la logique floue *type-2* qui présente un intérêt particulier notamment dans le domaine de l'automatique et de la conduite de procédés. Cet outil, introduit pour la première fois par le professeur L. Zadeh en 1975 [7], et intensivement étudié par Mendel et ses collaborateurs ces dernières années [4] [5] [6] [8] [9][10][11][12][13] offre un degré de liberté supplémentaire qui permet l'incorporation des incertitudes.

Dans un premier temps, nous donnerons brièvement un rappel sur les concepts de base, opérateurs et structures des modèles de systèmes flous classiques (dit *type-1*). En suite, nous allons montrer comment généraliser ces opérateurs pour les ensembles flous *type-2* moyennant le fameux principe d'extension de Zadeh (voir annexe 2).

2. La logique floue *type-1*

2.1. Principe et définition

La logique floue repose sur la théorie des ensembles flous développée par Zadeh en 1965 [14]. A côté d'un formalisme mathématique fort développé, nous préférons aborder la présentation de manière intuitive.

Les notions de température moyenne ou de courant faible sont relativement difficiles à spécifier de manière précise. En logique floue, Le concept de variable linguistique, comme son nom le suggère, joue un rôle très important. C'est une variable dont les valeurs sont des mots ou des phrases au lieu de nombres [15]. Par exemple, ' *Température* ' est une variable

linguistique et l'ensemble des valeurs qu'elle peut prendre est appelé l'ensemble de termes (nuances), soient par exemple : *Basse*, *Moyenne*, *Elevée*. D'autres termes peuvent être construits à l'aide de modificateurs linguistiques tels que (très, plus ou moins, un peu...etc.). Chaque valeur linguistique (e.g., faible) est caractérisée par sa fonction d'appartenance $\mu_{faible}(x)$ qui associe à chaque valeur numérique x de 'Température', un degré qui reflète la vérité du sens faible.

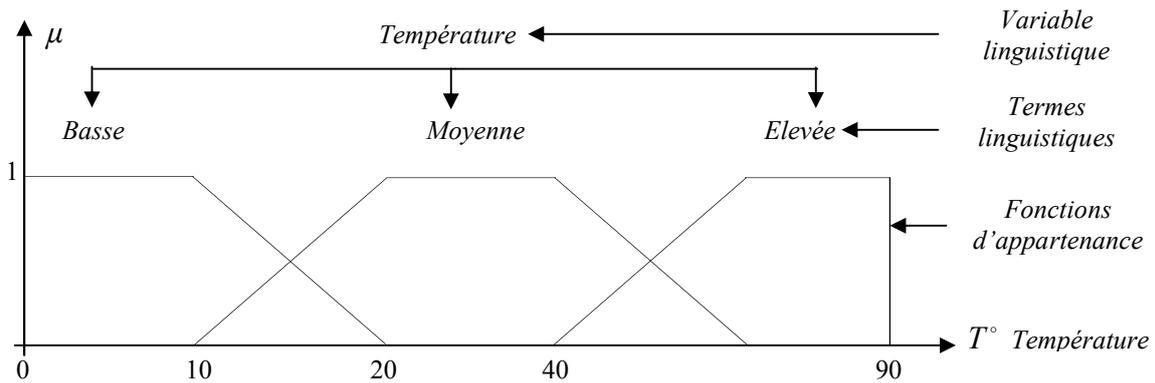


Fig.1.1. Exemple d'une variable linguistique 'Température' avec trois termes linguistiques.

2.2. Opérateurs et normes

Comme dans la théorie des ensembles classiques, on définit l'intersection (\cap), l'union (\cup) des ensembles flous ainsi que le complémentaire ($\bar{}$) d'un ensemble flou. Ces relations sont traduites par les opérateurs 'et', 'ou' et 'non'. De nouvelles fonctions d'appartenance liées à ces opérateurs sont établies [7] [8] [14] [16] [17]:

- L'intersection : $x \in A \cap B \Leftrightarrow x \in \mu_{A \cap B} \mid \mu_{A \cap B}(x) = \min [\mu_A(x), \mu_B(x)]$ (1.1)

- L'union : $x \in A \cup B \Leftrightarrow x \in \mu_{A \cup B} \mid \mu_{A \cup B}(x) = \max [\mu_A(x), \mu_B(x)]$ (1.2)

- Le complément (négation) : $\forall x \in X, \mu_{\bar{A}}(x) = 1 - \mu_A(x)$ (1.3)

L'intersection floue de deux ensembles flous peut être définie d'une façon plus générale par une opération binaire sur l'intervalle unitaire, c-à-d, une fonction de la forme $T: [0, 1] \times [0, 1] \rightarrow [0, 1]$. Pour que la fonction T soit qualifiée pour assurer une intersection floue, elle doit vérifier quelques propriétés. Des fonctions connues par le nom t-normes

(normes triangulaires) possèdent ces propriétés. Similairement, des fonctions appelées t-conormes peuvent être utilisées pour l'union floue [8] [17].

2.2.1. T-norme (intersection floue)

Une t-norme T est une opération binaire sur l'intervalle unitaire, qui vérifie au moins les axiomes suivants pour tout $a, b, c \in [0, 1]$ [8] [15] [17]:

- 1) $T(a, 1) = a$
- 2) $b \leq c \Rightarrow T(a, b) \leq T(a, c)$
- 3) $T(a, b) = T(b, a)$
- 4) $T(a, T(b, c)) = T(T(a, b), c)$

Les t-normes fréquemment utilisées sont [14]

- Intersection standard (Zadeh) : $T(a, b) = \min(a, b)$
- Produit algébrique (Intersection probabilistique) : $T(a, b) = ab$

2.2.2. T-conorme (union floue)

Une t-conorme S est une opération binaire sur l'intervalle unitaire, qui vérifie en plus des axiomes 2, 3, et 4 la propriété : $S(a, 0) = a$ [8] [14]:

Les t-conormes fréquemment utilisées sont [4] [11]

- Union standard (Zadeh) : $S(a, b) = \max(a, b)$
- Somme algébrique (Union probabilistique) : $S(a, b) = a + b - ab$

Une t-norme T et une t-conorme S , peuvent être reliées par la loi de De Morgan suivante [8]:

$$S(a, b) = 1 - (1 - a)T(1 - b) \quad (1.4)$$

2.3. Logique floue [8]

La logique floue commence en empruntant des notions de la logique ordinaire, comme pour les ensembles flous qui eux aussi empruntent des notions de la théorie des ensembles ordinaires. Cependant, ce dernier passage est inadéquat pour les applications d'ingénierie. Car en ingénierie, le principe cause-effet est la pierre angulaire de tous les modèles. Ce qui n'est pas le cas dans la logique classique. Pour contourner ce problème, Mamdani a défini un opérateur d'implication (Engineering implication operator).

Le principe d'extension des ensembles ordinaires vers des ensembles flous et l'extension de la logique classique vers la logique floue est réalisé en remplaçant des fonctions d'appartenance bivalentes de la logique classique par des fonctions d'appartenance floues à plusieurs degrés de vérité. Donc, la citation (règle) : *SI - ALORS* « *SI x est A, alors y est B* », $x \in X$ et $y \in Y$, appelée en logique classique *modus ponens*, sera caractérisée, dans le cas de la logique floue, par un ensemble $\mu_{A \rightarrow B}(x, y) \in [0, 1]$ et le *modus ponens* est étendu vers le *modus ponens généralisé*.

- *Modus ponens généralisé* [8]

Prémisse 1: 'x est A^* ' ;

Prémisse 2 : 'si x est A, alors y est B ;

Conséquence : y est B^* '.

L'ensemble flou A^* n'est pas nécessairement égal à l'antécédent A de la règle, et l'ensemble flou B^* n'est pas nécessairement la même conséquence B de la règle. On dit $A \neq A^*$ mais A^* est similaire à A ; et $B \neq B^*$ mais B^* est similaire à B.

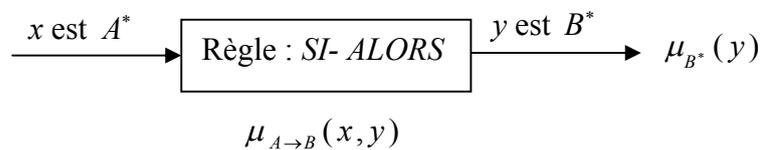


Fig.1.2. Système interprétant le modus ponens généralisé.

L'opérateur d'implication de contrôle qui a été défini par Mamdani possède la forme suivante :

$$\mu_{A \rightarrow B}(x, y) = \min[\mu_A(x), \mu_B(y)] \quad (1.5)$$

Les raisons de ce choix étaient la simplicité de programmation. Plus tard, Larsen a proposé l'implication *produit* suivante :

$$\mu_{A \rightarrow B}(x, y) = \mu_A(x) \cdot \mu_B(y) \quad (1.6)$$

Une fois de plus, la facilité dans la programmation a été la raison de ce choix. Aujourd'hui, les inférences *min* et *produit* sont largement utilisées dans les applications de la logique floue.

3. Modèles de contrôleurs flous *type-1*

En fonction de la forme des conséquences des règles, deux types de modèles flous peuvent être distingués [15] [18]:

- Modèle flou linguistique, où l'antécédent et la conséquence sont des propositions floues (on l'appelle aussi modèle flou de Mamdani).
- Modèle flou type Takagi-Sugeno-Kang (*TSK*), où la conséquence est une fonction ordinaire des variables d'entrée (antécédents).

3.1. Modèle flou linguistique

La forme générale d'une règle dans ce modèle est [6] [8] [19] [20] :

$$R^{(l)} : \text{Si } x_1 \text{ est } A_1^l \text{ et } x_2 \text{ est } A_2^l \text{ et } \dots \text{ et } x_p \text{ est } A_p^l \text{ alors } y \text{ est } B^l \quad (1.7)$$

où $A_i^l \subseteq X \subset \mathfrak{R}$ est la variable antécédent représentant la variable d'entrée du système flou. Similairement, $B^l \subseteq Y \subset \mathfrak{R}$ est la variable conséquence représentant la variable de sortie du système flou. A_i^l et B^l sont des termes linguistiques définis par les ensembles multidimensionnelles $\mu_{A_i^l}(x) : X \rightarrow [0 \ 1]$ et $\mu_{B^l}(x) : Y \rightarrow [0 \ 1]$, respectivement. Si tous les ensembles des conséquences possèdent seulement une seule valeur non nulle, dans ce cas, on parle d'un modèle linguistique *singleton*, dans le cas contraire, le modèle est dit *non singleton*.

La relation entrée/sortie du système flou est réalisée par le mécanisme d'inférence floue. Etant donné la base de règles et la valeur de l'entrée, ce dernier délivre une valeur de sortie correspondante. Comme les valeurs entrée/sortie sont numériques, alors le système flou doit être doté de deux interfaces de conversion appelées fuzzificateur et défuzzificateur, d'où la structure générale d'un contrôleur flou *type-1* de la figure (1.3) [19] [21] [22].

3.1.1. Fuzzification

Elle transforme toute variable numérique d'entrée en variable floue. Cependant, dans notre travail, on utilise que la fuzzification singleton (i.e, la fonction d'appartenance est égale à 1 quant son argument correspond à l'entrée numérique et nulle ailleurs).

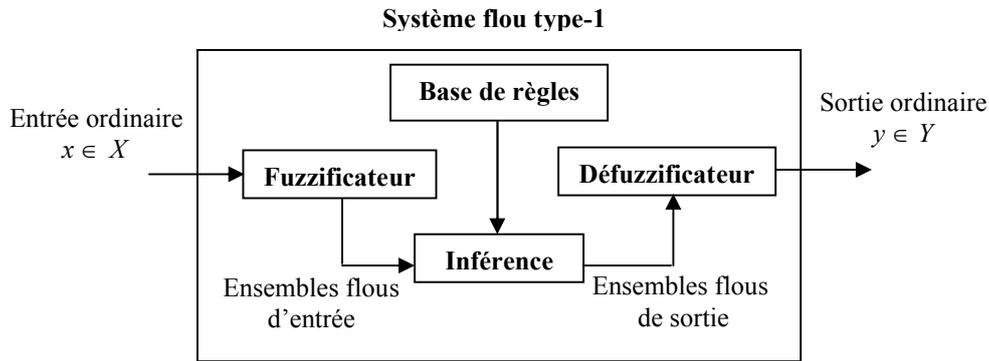


Fig.1.3. Structure générale d'un contrôleur flou *type-1*.

3.1.2. Base de règles

Les règles floues permettent d'exprimer sous forme linguistique les variables d'entrée aux variables de commande du système.

La littérature nous donne quelques méthodes qui peuvent être mises en œuvre pour recueillir ces règles [3]. Il s'avère que bon nombre de contrôleurs flous exploités repose sur une base de règles anti diagonale, dite base de Mac Vicar-Whelan. Les règles sont souvent arrangées dans une table appelée table de décision. Le tableau (1.1) représente un exemple de table de décision pour les deux variables linguistiques d'entrée e et de et la variable floue de sortie u . Quelque soit la dimension de la table, soit pleine ou creuse, elle doit toujours respecter certaines propriétés : la cohérence, la continuité et la complétude [3].

$e \backslash de$		u		
		N	Z	P
N	PG	P	Z	
Z	P	Z	N	
P	Z	N	NG	

- **NG** : négatif grand.
- **N** : négatif.
- **Z** : zéro.
- **P** : positif.
- **PG** : positif grand.

Tab 1.1. Exemple de table de décision.

3.1.3. Mécanisme d'inférence

Consiste à calculer les valeurs de vérité des différentes règles et à associer à chacune d'elle une valeur de sortie qui dépend de sa partie conclusion.

L'utilisation des opérateurs *et/ou* et l'implication dans les règles floues, tels qu'ils sont définis par les fonctions élémentaires (*min, max, produit...*), et la combinaison de ces différentes fonctions conduit à plusieurs méthodes d'inférence.

3.1.4. Défuzzification

La défuzzification consiste à calculer, à partir de l'ensemble flou de sortie, la valeur numérique la plus représentative de ce dernier. Plusieurs méthodes sont proposées dans la littérature [8] [19] [22]. Dans notre cas, on s'est limité à deux méthodes: le centre de gravité (*CG*) et celle des centres d'ensembles (*COS*).

- Méthode du centre de gravité

C'est la méthode de défuzzification la plus utilisée. Elle est exprimée par :

$$y_{CG} = \frac{\sum_{i=1}^N y_i \mu_B(y_i)}{\sum_{i=1}^N \mu_B(y_i)} \quad (1.8)$$

où B est l'ensemble de sortie discrétisé en N points (y_1, \dots, y_N) .

- Méthode des centres d'ensembles

Chaque ensemble de sortie d'une règle est remplacé par un singleton représentant son centroïde. La sortie numérique est donnée par la relation suivante:

$$y_{COS} = \frac{\sum_{l=1}^M c^l \mathfrak{F}_{i=1}^p \mu_{A_i^l}(y_i)}{\sum_{i=1}^N \mathfrak{F}_{i=1}^p \mu_{A_i^l}(y_i)} \quad (1.9)$$

où \mathfrak{F} indique la t-norme choisie, et c^l est le centroïde du $l^{\text{ième}}$ ensemble flou de sortie.

3.2. Modèle flou type Takagi-Sugeno-Kang (TSK)

Ce type de modèles flous a été introduit en 1985 par Takagi-Sugeno-Kang [18]. Les conséquences des règles sont des fonctions ordinaires des entrées du modèle [8] [18] [22]:

$$R^{(l)} : \text{Si } x_1 \text{ est } A_1^l \text{ et } \dots \text{ et } x_p \text{ est } A_p^l \text{ alors } y^l = f(x_1, \dots, x_p) \quad (1.10)$$

Les fonctions conséquences f^l sont choisies typiquement de formes paramétriques. Si toutes les conséquences du modèle deviennent des constantes, nous obtenons alors un modèle TSK d'ordre zéro qui peut être vu comme un modèle flou linguistique de type singleton [18].

4. Concept des systèmes flous *type -2*

Le concept des systèmes flous *type-2* est basé sur les ensembles flous *type-2*. Ces derniers ont été introduits par Zadeh en 1975 comme extension du concept des ensembles flous ordinaires appelés ensembles flous *type-1* [7]. Un ensemble flou *type-2* est caractérisé par une fonction d'appartenance floue, c'est-à-dire, le *degré d'appartenance* de chaque élément de l'ensemble est un ensemble flou dans $[0, 1]$. De tels ensembles peuvent être utilisés dans les situations où nous avons de l'incertitude sur les valeurs d'appartenance elles mêmes. L'incertitude peut être soit dans la forme de la fonction d'appartenance ou dans l'un de ses paramètres.

Considérons la transition des ensembles ordinaires vers les ensembles flous. Lorsque nous ne pouvons pas déterminer le degré d'appartenance d'un élément à un ensemble par 0 ou 1, on utilise les ensembles flous *type-1*. Similairement, lorsque nous ne pouvons pas déterminer les fonctions d'appartenance floues par des nombres réels dans $[0, 1]$, on utilise alors les ensembles flous *type-2*. Ainsi, les ensembles flous *type-1* peuvent être considérés comme une approximation du premier ordre de l'incertitude. Alors que les ensembles flous *type-2* seront considérés comme approximation du deuxième ordre [6].

- **Exemple 1**

Soit un ensemble flou caractérisé par une fonction d'appartenance gaussienne de moyenne m et d'un écart-type σ qui prend des valeurs dans l'intervalle $[\sigma_1, \sigma_2]$, tel que [5] [20]

$$\mu(x) = \exp[-1/2(x - m)^2 / \sigma^2]; \sigma \in [\sigma_1, \sigma_2] \quad (1.11)$$

A chaque valeur de σ , nous allons obtenir une courbe d'appartenance différente (voir figure.1.4.a). La valeur d'appartenance de n'importe quel x (excepté $x = m$) possède plusieurs valeurs (dépendant de σ). Ce qui veut dire que le degré d'appartenance n'est pas un nombre ordinaire, mais un ensemble flou. La figure (1.4.a) montre le domaine de l'ensemble flou associé à $x = 0.7$; mais ne montre pas la fonction d'appartenance associée à ce domaine.

- **Exemple 2**

Soit un ensemble flou avec une fonction d'appartenance gaussienne ayant un écart-type σ fixe, mais une valeur moyenne m incertaine dont les valeurs appartiennent à l'intervalle $[m_1, m_2]$, tel que [5] [20]

$$\mu(x) = \exp[-1/2(x - m)^2 / \sigma^2]; m \in [m_1, m_2] \quad (1.12)$$

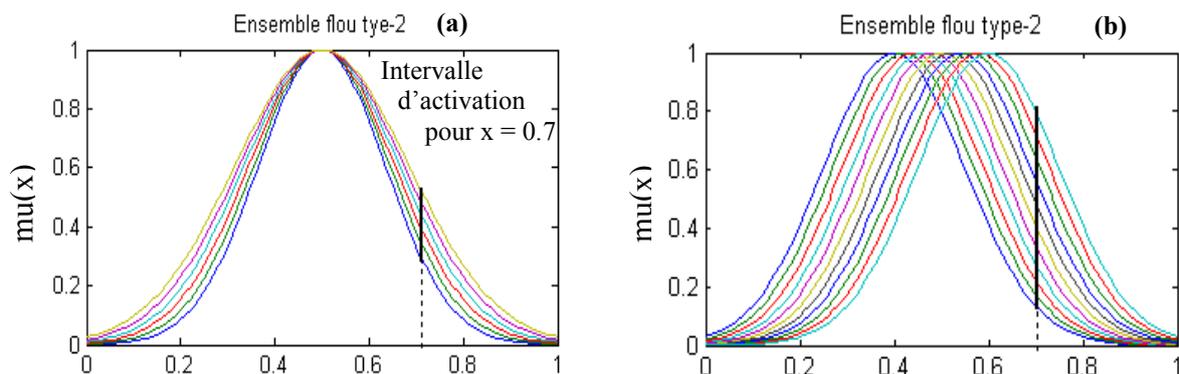


Fig.1.4. Ensemble floue *type-2* obtenu à partir d'un ensemble flou *type-1* :

(a) variance incertaine, (b) moyenne incertaine.

La figure (1.4.b) illustre l'allure d'un tel ensemble flou. Ici, comme dans (a), il n'est pas possible de déduire la fonction d'appartenance associée à l'appartenance floue de n'importe quel x .

- **Exemple 3**

Soit $m(x)$ un ensemble flou *type-1* caractérisé par une fonction d'appartenance gaussienne à moyenne M et un écart-type σ_x . Cet ensemble a une valeur réelle unique pour chaque $x \in X$ tel que [20]

$$m(x) = \exp \left[-1/2 (x - M)^2 / \sigma_x^2 \right] \quad (1.13)$$

La figure (1.5) représente une telle fonction d'appartenance. Imaginons maintenant que cette appartenance de x est un ensemble flou. Appelant les éléments du domaine de cet ensemble 'appartenance primaire de x ' (noté par μ_1) et les valeurs d'appartenance de ces appartenances primaires 'appartenances secondaires de x ' (notées $\mu_2(x, \mu_1)$). Ainsi, pour un certain x donné, nous avons un ensemble flou *type-1* dont les éléments de son domaine sont les appartenances primaires de x , et les degrés d'appartenance correspondants sont les appartenances secondaires de x .

Si nous supposons que les appartenances secondaires sont des gaussiennes de moyenne $m(x)$ et d'un écart type σ_m , comme le représente la figure (1.5), on peut alors représenter la fonction d'appartenance secondaire pour chaque x par

$$\mu_2(x, \mu_1) = \exp \left[-1/2 (\mu_1 - m(x))^2 / \sigma_m^2 \right] \quad (1.14)$$

où $\mu_1 \in [0, 1]$ et m est comme dans (1.13). Les équations (1.13) et (1.14) peuvent être combinées comme suit :

$$\mu_2(x, \mu_1) = \exp \left[-1/2 \left(\mu_1 - e^{-\frac{1}{2} \left(\frac{x-M}{\sigma_x} \right)^2} \right)^2 / \sigma_m^2 \right] \quad (1.15)$$

Donc, de l'équation (1.15) nous pouvons considérer qu'une fonction d'appartenance secondaire comme fonction à deux variables x et μ_1 . Le degré d'appartenance $\mu(x)$ de chaque x , représentant toutes les appartenances primaires ainsi que leurs appartenances secondaires correspondantes, peut être écrit sous la forme (1.16) (voir figure 1.5). La figure (1.6) est une autre représentation, dite, tridimensionnelle d'un ensemble flou *type-2*.

$$\mu(x) = \int_{\mu_1 \in [0, 1]} \mu_2(x, \mu_1) \mid \mu_1 ; \quad x \in X \tag{1.16}$$

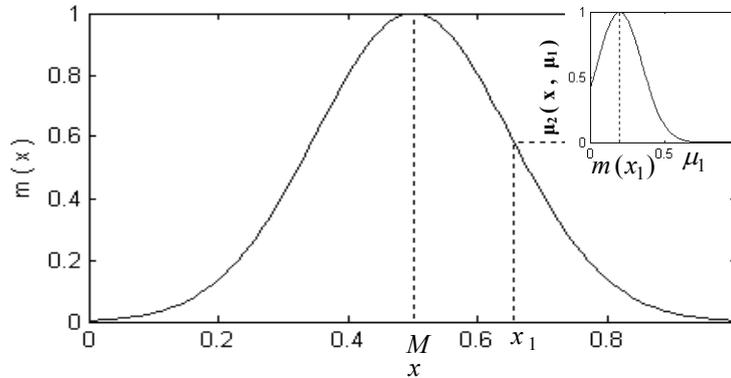


Fig.1.5. Le graphe de la gaussienne $m(x)$ et le degré d'appartenance de type gaussien correspondant à $x=x_1$.

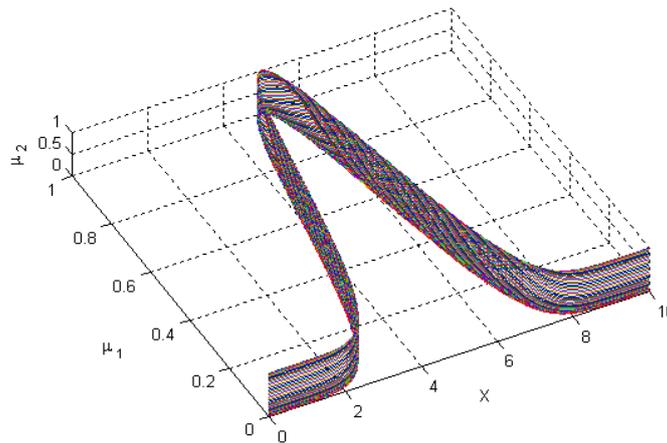


Fig.1.6. Représentation tridimensionnelle d'un ensemble flou *type-2* gaussien.

5. Conclusion

Dans ce chapitre, nous avons présenté les éléments de base des systèmes flous *type-1*. Ce qui va nous permettre d'aborder les systèmes flous *type-2*, objet de notre étude, avec moins de difficultés. Ces derniers ont été introduits, à la fin du chapitre, à travers un bref exposé.

Chapitre 2

Représentation des systèmes flous *type-2*.

1. Introduction

Dans ce chapitre, nous allons procéder à l'extension des systèmes flous *type-2*. Les opérations et les modèles de ces systèmes seront abordés. Une partie consistante de ce chapitre sera consacrée à l'étude du bloc de traitement de sortie qui présente une difficulté liée essentiellement à la nécessité de la *réduction de type* des ensembles résultants de l'inférence floue. A cet effet, une procédure algorithmique de réduction de type des ensembles flous *type-2* intervalle, relativement facile à celle de réduction de type d'un système flou *type-2* général, sera présentée. Enfin, deux exemples d'application de ladite procédure terminent le chapitre.

2. Opérations sur les ensembles flous *type-2*

Nous rappelons que le degré d'appartenance de deux ensembles flous *type-2* $\tilde{A} \in X$ et $\tilde{B} \in X$ sont des ensembles flous *type-1*. Donc pour calculer l'union, l'intersection et le complément de \tilde{A} et \tilde{B} , nous aurons besoin d'étendre (généraliser) les opérations *min*, *max* et l'opération de *négation*, qui manipulent initialement des valeurs ordinaires, vers des ensembles flous [10]. Pour cela, nous allons utiliser le fameux principe d'extension de Zadeh (voir annexe 2) [7]. Soient $\mu_{\tilde{A}}(x)$ et $\mu_{\tilde{B}}(x)$ deux appartenances floues des deux ensembles précédents, tel que [5]:

$$\mu_{\tilde{A}}(x) = \sum_i f_x(u_i) / u_i \quad ; \quad u_i \in J \quad (2.1)$$

$$\mu_{\tilde{B}}(x) = \sum_j g_x(w_j) / w_j \quad ; \quad w_i \in J \quad (2.2)$$

En utilisant ledit principe, une opération binaire réelle $*$ peut être étendue vers les appartenances floues $\mu_{\tilde{A}}(x)$ et $\mu_{\tilde{B}}(x)$ [4] [10]:

$$\mu_{\tilde{A}}(x) * \mu_{\tilde{B}}(x) = \sum_{i,j} (f_x(u_i) \wedge g_x(w_j)) / u_i * w_j \quad (2.3)$$

où \wedge indique la t-norme choisie (*min* ou *produit*). Donc, en appliquant ce principe aux ensembles \tilde{A} et \tilde{B} , les fonctions d'appartenance pour l'intersection, l'union et la complémentation deviennent :

- L'intersection : $\tilde{A} \cap \tilde{B} = \sum_{i,j} (f_x(u_i) \wedge g_x(w_j)) / u_i \wedge w_j \quad (2.4)$

- L'union : $\tilde{A} \cup \tilde{B} = \sum_{i,j} (f_x(u_i) \wedge g_x(w_j)) / u_i \vee w_j \quad (2.5)$

- Le complément : $\tilde{A}^- = \sum_i f_x(u_i) / (1 - u_i) ; \quad \tilde{B}^- = \sum_i g_x(w_j) / (1 - w_j) \quad (2.6)$

où \vee est la t-conorme. Dans ce cas, et comme dans [5] [10] [13] [20], nous allons appeler les opérations \cup , \cap et $^-$: *join*, *meet*, et *négation*, respectivement. Les mêmes notations seront utilisées dans le cas continu.

2.1. Opération *meet* sous la t-norme *produit*

Considérant deux ensembles flous *type-1* réels A et B caractérisés par des fonctions d'appartenance f et g , respectivement. L'opération *meet* entre A et B sous une t-norme *produit* peut être définie par la relation (2.7) ci-après. Cette opération sera utilisée dans nos applications pour exprimer l'inférence floue dans les systèmes flous *type-2*. D'autres opérations sont illustrées par la figure (2.1).

$$A \cap B = \int_{v \in R} \int_{w \in R} [f(v)g(w)] / (vw) \quad (2.7)$$

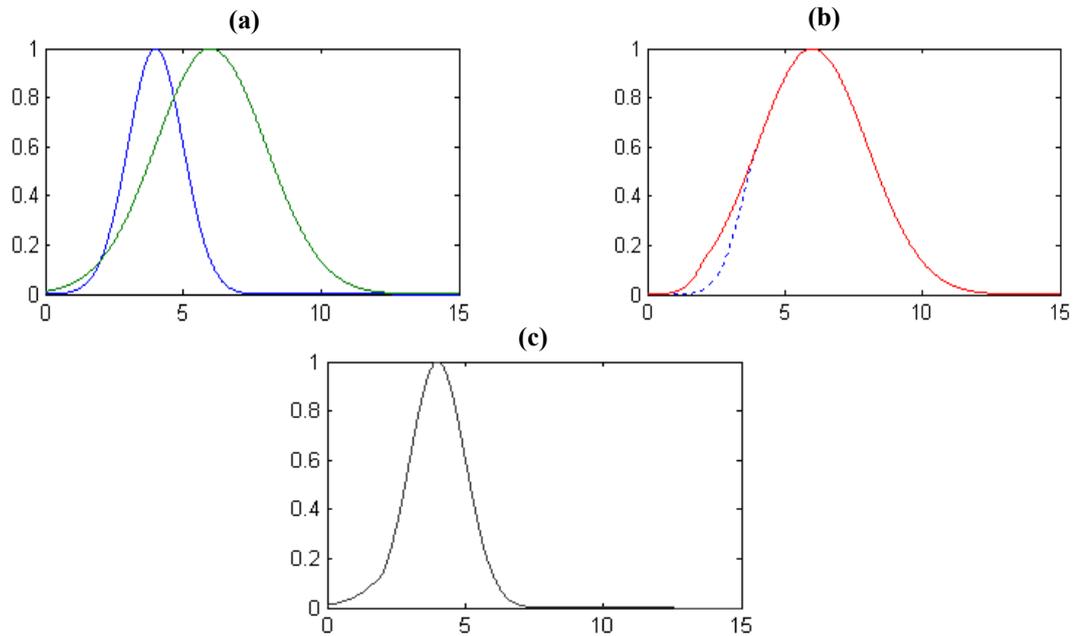


Fig.2.1 Opérations *join* et *meet* entre deux ensembles flous *type-1* gaussiens : (a) : Opérandes, (b) : En discontinu *join* sous t-norme *produit*, en continu *join* sous t-norme *min*, (c) : *meet* sous t-norme *min*.

Lorsque les ensembles A et B sont des ensembles flous *type-1* intervalle, l'équation (2.7) sera considérablement simplifiée. Si par exemple les domaines de A et B sont $[l_f, r_f]$ et $[l_g, r_g]$, respectivement, alors, la relation (2.7) devient

$$A \cap B = \int_{v \in R} \int_{w \in R} [1 \times 1] / (vw) \tag{2.8}$$

Chaque terme dans $A \cap B$ est égal au produit vw de $v \in A$ et $w \in B$, avec le plus petit terme étant $l_f l_g$ est le plus grand terme étant $r_f r_g$. Puisque A et B sont continu, alors $A \cap B$ est aussi continu, par conséquent, $A \cap B$ est un ensemble *type-1* intervalle défini par le domaine $[l_f l_g, r_f r_g]$ tel que

$$F \cap G = \int_{v \in [l_f l_g, r_f r_g]} 1/v \tag{2.9}$$

D'une manière similaire, le *meet* $\prod_{i=1}^n A_i$ de n ensembles *type-1* intervalles A_1, \dots, A_n , ayant comme domaines $[l_1, r_1], \dots, [l_n, r_n]$, respectivement, est un ensemble intervalle sur le domaine $\left[\prod_{i=1}^n l_i, \prod_{i=1}^n r_i \right]$.

2.2. Opérations algébriques

Les deux opérations les plus intéressantes sont celles de la multiplication et de l'addition [8] [10].

- Multiplication : $A \times B = \int_v \int_w [f(v) \wedge g(w)] / (v \times w)$ (2.10)

- Addition : $A + B = \int_v \int_w [f(v) \wedge g(w)] / (v + w)$ (2.11)

où \wedge indique la t-norme utilisée. Observons de (2.7) et (2.10) que la multiplication algébrique de A et B est le même que le *meet* de A et B , c-a-d, $A \times B = A \cap B$, par conséquent, l'opération *meet* sous une t-norme *produit* sera utilisée pour multiplier deux ensembles flous.

Lorsque A et B sont des ensembles *type-1* intervalle, (2.11) devient :

$$A + B = \int_v \int_w [1 \wedge 1] / (v + w) \quad (2.12)$$

Notons que chaque terme dans $A + B$ est égal à la somme $(v + w)$ de $v \in A$ et $w \in B$, avec le plus petit terme étant $l_f + l_g$ et le plus grand terme étant $r_f + r_g$. Puisque A et B sont continus, alors $A + B$ est aussi continu, par conséquent, $A + B$ est un ensemble *type-1* intervalle défini par le domaine $[l_f + l_g, r_f + r_g]$. D'une manière similaire, la somme de n ensembles *type-1* intervalle A_1, \dots, A_n , ayant comme domaines $[l_1, r_1], \dots, [l_n, r_n]$, respectivement, est un ensemble intervalle sur le domaine $\left[\sum_{i=1}^n l_i, \sum_{i=1}^n r_i \right]$ [10] [13].

3. Systèmes et contrôleurs flous type-2

Un système flou est dit de *type-2*, s'il inclut au moins un ensemble flou *type-2* dans les fonctions d'appartenance définissant ses prémisses ou ses conséquences [6] [15]. L'inférence floue, dans ce type de système, conduit à l'obtention d'ensembles flous de sortie de *type-2*.

La structure d'un système flou *type-2* est très similaire à celle d'un système flou *type-1*. Elle comporte un bloc de fuzzification, une base de règles et un mécanisme d'inférence. La seule différence est au niveau de la sortie. Dans le cas du *type-2*, le bloc de défuzzification est précédé d'un bloc *réduction de type* (figure 2.2) [6] [9] [20].

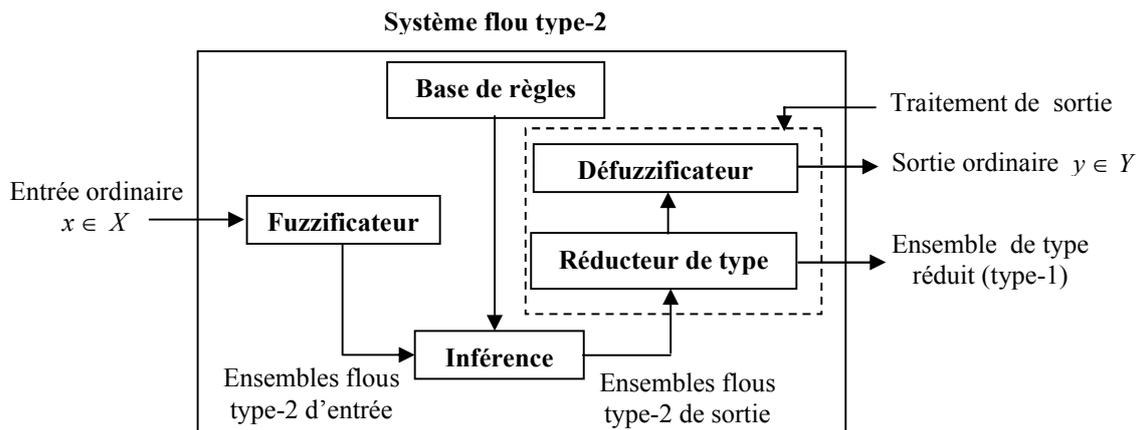


Fig .2.2. Structure générale d'un contrôleur flou *type -2*.

3.1. Base de règles

Comme nous l'avons déjà dit, la différence entre un système flou *type-1* et celui du *type-2* est due au type des fonctions d'appartenance des prémisses et des conséquences. La structure des règles dans le cas d'un système flou *type-2* alors reste inchangée [6][9] [19] [20]:

$$R^{(l)}: \text{SI } x_1 \text{ est } \tilde{A}_1^l \text{ et } x_2 \text{ est } \tilde{A}_2^l \text{ et } \dots \text{ et } x_p \text{ est } \tilde{A}_p^l \text{ ALORS } y \text{ est } \tilde{B}^l \quad (2.13)$$

3.2. Inférence floue

Nous considérons un système flou *type-2* avec p entrées et une seule sortie $y \in Y$. On suppose que les M règles constituant le système sont de la forme (2.13). Notons la fonction d'appartenance de cette relation floue *type-2* par $\mu_{\tilde{A}'_1 \times \dots \times \tilde{A}'_p \rightarrow \tilde{B}'_l}(x, y)$ où $\tilde{A}'_1 \times \dots \times \tilde{A}'_p$ dénote le produit cartésien de $\tilde{A}'_1, \dots, \tilde{A}'_p$ et $x = (x_1, \dots, x_p)^T$. Lorsqu'une entrée x' est appliquée, la composition de l'ensemble flou \tilde{X}' , à qui x' appartient, et la règle $R^{(l)}$, est calculée par la composition *sup-star* suivante [6] [20]:

$$\mu_{\tilde{X}' \circ \tilde{A}'_1 \times \dots \times \tilde{A}'_p \rightarrow \tilde{B}'_l}(y) = \bigcup_{x \in \tilde{X}'} [\mu_{\tilde{X}'}(x) \cap \mu_{\tilde{A}'_1 \times \dots \times \tilde{A}'_p \rightarrow \tilde{B}'_l}(x, y)] \quad (2.14)$$

Puisque nous utilisons une fuzzification de type singleton, l'ensemble flou \tilde{X}' est tel qu'il possède un degré d'appartenance unitaire à $x = x'$ et nul ailleurs. Par conséquent, (2.14) se réduit à :

$$\mu_{\tilde{X}' \circ \tilde{A}'_1 \times \dots \times \tilde{A}'_p \rightarrow \tilde{B}'_l}(y) = \mu_{\tilde{A}'_1 \times \dots \times \tilde{A}'_p \rightarrow \tilde{B}'_l}(x', y) \quad (2.15)$$

Notons l'ensemble de sortie correspondant à la $l^{\text{ième}}$ règle $\tilde{X}' \circ \tilde{A}'_1 \times \dots \times \tilde{A}'_p \rightarrow \tilde{B}'_l$ par \tilde{G}'_l . Le second membre de (2.15) est calculé en utilisant l'implication minimum ou produit (qui correspond à l'opération *meet* avec t-norme *min* ou *produit* dans le cas du *type-2*) :

$$\mu_{\tilde{G}'_l}(y) = \mu_{\tilde{A}'_1 \times \dots \times \tilde{A}'_p}(x') \cap \mu_{\tilde{B}'_l}(y) \quad (2.16)$$

qui est équivalent à :

$$\mu_{\tilde{G}'_l}(y) = \mu_{\tilde{A}'_1}(x_1) \cap \mu_{\tilde{A}'_2}(x_2) \cap \dots \cap \mu_{\tilde{A}'_p}(x_p) \cap \mu_{\tilde{B}'_l}(y) = \mu_{\tilde{B}'_l}(y) \cap \left[\prod_{i=1}^p \mu_{\tilde{A}'_i}(x_i) \right] \quad (2.17)$$

où \cap dénote l'opération *meet* basée sur la t-norme choisie.

3.3. Traitement de la sortie

Les ensembles de sortie issus de chacune des règles exprimées par (2.13) sont de *type-2*. Cela est dû à l'utilisation de fonctions d'appartenance de ce type dans les prémisses et dans

les conséquences. Donc, pour calculer la sortie defuzzifiée dans ce cas, nous devons utiliser des versions étendues, moyennant le principe d'extension de Zadeh, des méthodes de defuzzification *type-1*.

L'opération de defuzzification *type-1* permet d'obtenir une valeur numérique (ordinaire). Par contre, dans le cas des systèmes flous *type-2*, l'opération de defuzzification étendue permet d'obtenir, dans un premier temps, un ensemble flou de *type-1*. Cette opération qui transforme un ensemble flou *type-2* à un ensemble flou *type-1* s'appelle *réduction de type*. On appelle l'ensemble flou *type-1* résultant : *ensemble de type réduit*. Enfin, l'ensemble flou de *type réduit* doit être defuzzifié pour obtenir un nombre ordinaire [6] [11] [12][20].

4. Réduction de type des ensembles flous *type-2*

Dans cette section, nous allons généraliser le concept du centroïde d'un ensemble flou *type-1* vers celui du *type-2*. Le centroïde d'un ensemble flou A de *type-1* défini sur un domaine discret de N points, est donné par [6] [10] [19][22]

$$C_A = \frac{\sum_{i=1}^N x_i \mu_A(x_i)}{\sum_{i=1}^N \mu_A(x_i)} \quad (2.18)$$

Similairement, le centroïde d'un ensemble flou \tilde{A} de *type-2*, défini sur un domaine discret de N points, peut être étendu, en utilisant le principe d'extension, vers [14] [10] [12] :

$$C_{\tilde{A}} = \int_{\theta_1} \dots \int_{\theta_N} [\mu_{D_1}(\theta_1) \wedge \dots \wedge \mu_{D_N}(\theta_N)] \left/ \frac{\sum_{i=1}^N x_i \theta_i}{\sum_{i=1}^N \theta_i} \right. , \theta_i \in D_i \quad (2.19)$$

où : $D_i = \mu_{\tilde{A}}(x_i)$ et \wedge : la t-norme choisie (*min* ou *produit*). L'équation (2.19) signifie que : chaque point x_i de \tilde{A} possède un degré d'appartenance flou de *type-1* noté $D_i = \mu_{\tilde{A}}(x_i)$. Pour trouver le centroïde, nous devons considérer toutes les combinaisons $\{\theta_1, \dots, \theta_N\}$ tel que $\theta_i \in D_i$. Pour chacune de telles combinaisons, nous calculons un centroïde de *type-1* en utilisant l'équation (2.18) dans laquelle nous remplaçons chaque $\mu_{\tilde{A}}(x_i)$ par θ_i . A chaque

point centroïde, nous associons un degré d'appartenance égal à la t -norme des degrés d'appartenance des θ_i aux D_i . Si plus d'une combinaison des θ_i nous donnent le même point centroïde, nous conservons celui possédant le plus grand degré d'appartenance [6]. Si nous posons $\sum_{i=1}^N x_i \theta_i / \sum_{i=1}^N \theta_i = x$, alors, (2.19) devient :

$$C_{\tilde{A}} = \int_x \sup_{\{\theta_1, \dots, \theta_N\}} [\mu_{D_1}(\theta_1) \wedge \dots \wedge \mu_{D_N}(\theta_N)] / x \quad (2.20)$$

Un ensemble flou \tilde{A} type-2 peut être vu comme une collection d'ensembles flous de type-1 appelés : ensembles flous type-1 encastés dans \tilde{A} [6] [10].

Dans la représentation bidimensionnelle d'un ensemble flou type-2, un ensemble type-1 encasté est l'un des fonctions d'appartenance résidentes à l'intérieur de la zone hachurée (figure 2.3.a). Cette figure montre un exemple d'un ensemble type-1 encasté dans un ensemble type-2. Pour l'ensemble \tilde{A} type-2 de cette figure, chaque combinaison $\{\theta_1, \dots, \theta_N\}$ tel que $\theta_i \in D_i$ correspond à une fonction d'appartenance d'un ensemble type-1 encasté. Par conséquent, le centroïde $C_{\tilde{A}}$ de \tilde{A} peut être considéré comme un ensemble type-1 tel que les éléments de son domaine sont les centroïdes de tous les ensembles type-1 encastés dans \tilde{A} . Le degré d'appartenance d'un centroïde dans $C_{\tilde{A}}$ d'un ensemble encasté dans \tilde{A} est calculé par la t -norme de toutes les appartenances secondaires correspondantes aux $\{\theta_1, \dots, \theta_N\}$.

Remarquons que lorsque le domaine de \tilde{A} (ou de $\mu_{\tilde{A}}(x)$) est continu, le domaine de $C_{\tilde{A}}$ est aussi continu, par conséquent, le nombre de tous les ensembles type-1 encastés dans \tilde{A} est infini. Pour cela, le domaine de \tilde{A} (ou de $\mu_{\tilde{A}}(x)$) doit être discrétisé pour pouvoir calculer $C_{\tilde{A}}$ (figure 2.3.b). Notons aussi que si le domaine de chaque D_i est discrétisé en M points, le nombre de combinaisons possibles $\{\theta_1, \dots, \theta_N\}$ est M^N s'avère très grand même si M et N sont petits.

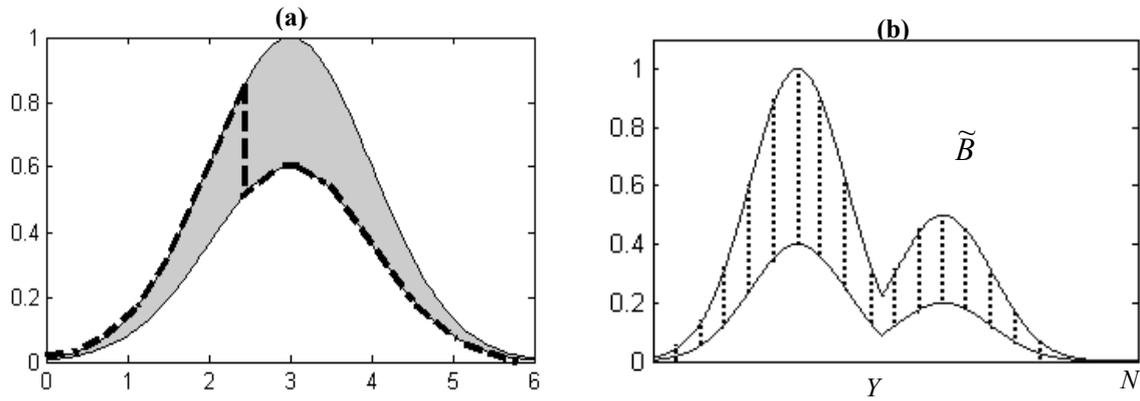


Fig. 2.3. (a) : Ensemble flou *type-1* (représenté en gras discontinu) encastré dans un ensemble flou *type-2*.
 (b) : Ensemble flou *type-2* discrétisé pour des raisons de réduction de type.

4.1. Réduction de type par la méthode du centroïde [14]

Le réducteur de type par la méthode du centroïde combine tous les ensembles *type-2* des sorties des règles en utilisant l'union. Le degré d'appartenance de $y \in Y$ devient alors

$$\mu_{\tilde{B}}(y) = \bigcup_{l=1}^M \mu_{\tilde{B}^l}(y) \quad (2.21)$$

où $\mu_{\tilde{B}}(y)$ est défini comme dans (2.17). Le réducteur de type calcule, par cette méthode, le centroïde de \tilde{B} , noté $Y_c(x)$, et cela en utilisant les relations (2.19) et (2.20)

$$Y_c(x) = \int_{\theta_1} \dots \int_{\theta_N} [\mu_{D_1}(\theta_1) \wedge \dots \wedge \mu_{D_N}(\theta_N)] \frac{\sum_{i=1}^N y_i \theta_i}{\sum_{i=1}^N \theta_i} \quad (2.22)$$

où $D_i = \mu_{\tilde{B}}(y_i)$ et $\theta_i \in \mu_{\tilde{B}}(y_i)$ ($i=1, \dots, N$). L'équation (2.22) peut être aussi écrite sous la forme suivante :

$$Y_c(x) = \int_y \sup_{\{\theta_1, \dots, \theta_N\}} [\mu_{D_1}(\theta_1) \wedge \dots \wedge \mu_{D_N}(\theta_N)] / y \quad (2.23)$$

où $\{\theta_1, \dots, \theta_N\}$ sont tel que $\sum_{i=1}^N y_i \theta_i / \sum_{i=1}^N \theta_i = y$.

Pour calculer $Y_c(x)$, nous devons suivre les étapes suivantes :

- 1) Calculer la sortie combinée en utilisant (2.21).
- 2) Discrétiser l'espace de sortie Y en N points y_1, \dots, y_N .
- 3) Discrétiser le domaine de chaque $\mu_{\tilde{B}}(y_i)$ en un nombre convenable de points.
- 4) Enumérer tous les ensembles encastés. Si par exemple chaque $\mu_{\tilde{B}}(y_i)$ est discrétisé en M points, nous allons avoir M^N ensembles flous *type-1* encastés, et si $\mu_{\tilde{B}}(y_i)$ est discrétisé en M_i points, nous allons alors obtenir $\prod_{j=1}^N M_j$ ensembles encastés.
- 5) Calculer l'ensemble centroïde type réduit en utilisant (2.22), cela revient à calculer le centroïde de chaque ensemble énuméré (encasté) et de lui associer un degré d'appartenance égal à la t-norme des appartenances secondaires correspondantes aux ensembles énumérés.

4.2. Réduction de type par la méthode des centres d'ensembles [6] [10]

La réduction de type par la méthode des centres d'ensembles consiste à remplacer chaque ensemble de sortie par son centroïde qui est lui-même un ensemble flou *type-1* si l'ensemble de sortie est de *type-2*, puis, calculer la moyenne pondérée de ces centroïdes, où le poids associé au $l^{i\text{ème}}$ centroïde sera le degré d'activation de la $l^{i\text{ème}}$ règle $\prod_{i=1}^p \mu_{\tilde{A}_i}(x_i)$. L'expression du réducteur de type, dans ce cas, est donnée par l'expression (2.24) obtenue par extension de la relation (1.9)

$$Y_{\text{cos}}(x) = \int_{d_1} \dots \int_{d_M} \int_{e_1} \dots \int_{e_M} \mathfrak{S}_{l=1}^M \mu_{C_l}(d_l) \wedge \mathfrak{S}_{l=1}^M \mu_{E_l}(e_l) \left/ \frac{\sum_{l=1}^M d_l e_l}{\sum_{l=1}^M e_l} \right. \quad (2.24)$$

où : \mathfrak{S} et \wedge indique les t-normes choisies ; $d_l \in C_l = C_{\tilde{B}_l}$, le centroïde du $l^{i\text{ème}}$ ensemble de sortie ; et $e_l \in E_l \prod_{i=1}^p \mu_{\tilde{A}_i}(x_i)$, le degré (intervalle) d'activation associé au $l^{i\text{ème}}$ ensemble de sortie, pour $l=1, \dots, M$. Donc pour calculer $y_{\text{cos}}(x)$, nous devons suivre les étapes suivantes :

- 1) Discrétiser l'espace de sortie Y en un nombre convenable de points, puis calculer les centroïdes C_l de chaque ensemble de sortie en utilisant (2.19). Les résultats doivent être mémorisés pour une future utilisation.

- 2) Calculer les degrés d'activation $E_l \prod_{i=1}^p \mu_{\tilde{A}_i'}(x_i)$ pour chaque règle.
- 3) Discrétiser le domaine de chaque C_l (C_l est le centroïde du $l^{i\text{ème}}$ ensemble de sortie) en un nombre convenable de points, soit M_l ($l=1, \dots, M$).
- 4) Discrétiser le domaine de chaque E_l en un nombre convenable de points, soit N_l ($l=1, \dots, M$).
- 5) Enumérer toutes les combinaisons possibles $\{d_1, \dots, d_m, e_1, \dots, e_m\}$ tel que $d_l \in C_l$ et $e_l \in E_l$. Le nombre total des combinaisons sera $\prod_{j=1}^M M_j N_j$.
- 6) Calculer l'ensemble de type réduit $Y_{COS}(x)$ en utilisant (2.24).

4.3. Interprétation de l'ensemble type réduit [6] [15] [20]

Comme nous l'avons vu, l'ensemble de type réduit est un ensemble flou *type-1* sur Y . Si nous considérons l'équation (2.22), l'ensemble de type réduit est le centroïde de l'ensemble flou *type-2* de la sortie du système flou. Par conséquent, chaque élément de l'ensemble de type réduit est le centroïde d'un certain ensemble *type-1* encastré dans l'ensemble de sortie du système flou *type-2*. Chacun de ces ensembles encastrés peut être considéré comme la sortie d'un certain système flou *type-1*. Ce qui nous mène à dire que le système flou *type-2* peut être vu comme une collection de plusieurs systèmes flous *type-1* encastrés dans le système flou *type-2*. Donc, l'ensemble de type réduit est une collection des sorties de tous les systèmes flous *type-1* encastrés dans le système flou *type-2* (figure 2.4).

Le degré d'appartenance d'un point $y_1 \in Y$ dans l'ensemble de type réduit indique le niveau d'incertitude associé au système flou *type-1* dont la sortie est y_1 . Ce niveau d'incertitude est calculé à partir des appartenances secondaires associées à l'ensemble *type-1* encastré dans la sortie du *type-2*. Si plusieurs systèmes flous *type-1* encastrés délivrent la même sortie y_1 , l'appartenance correspondante au point en question est prise comme le supremum des niveaux d'incertitudes associés à tous les systèmes flous encastrés concernés.

Si toutes les incertitudes d'un système flou *type-2* se réduisent à des incertitudes d'un système flou *type-1*, c-à-d, toutes les fonctions d'appartenance *type-2* se réduisent à leurs fonctions d'appartenance principales, les degrés d'appartenance des prémisses et des conséquences de chaque point seront réduits alors à des points (valeurs ordinaires). Cela va engendrer, bien sûr, la réduction de l'ensemble *type-2* de sortie à sa fonction d'appartenance principale, et la réduction de l'ensemble de type réduit vers un point singulier qui représente

le centroïde de la fonction d'appartenance principale de l'ensemble de sortie. Un système flou type-2 peut alors être vu comme une version perturbée d'un système flou type-1 à cause des incertitudes présentes dans les fonctions d'appartenance.

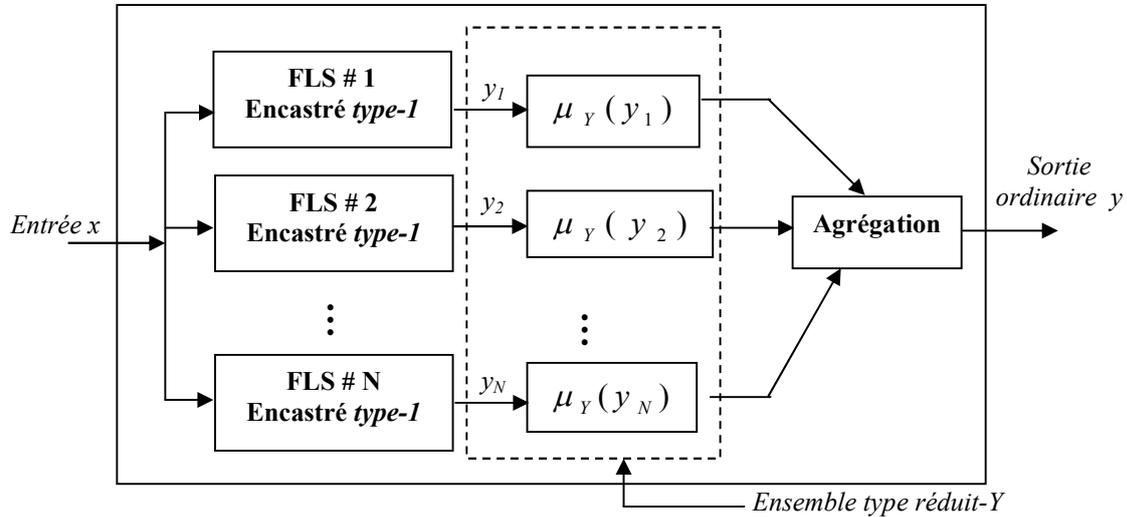


Fig. 2.4. Système flou type-2 vu comme collection de N systèmes type-1, avec Y la collection des sorties de ces systèmes encastrés.

5. Réduction de type des ensembles flous type-2 intervalle [6]

Un système flou type-2 est très compliqué à cause de la réduction de type. La formule générale du calcul de cet ensemble de type réduit est la suivante :

$$Y(Z_1, \dots, Z_M, W_1, \dots, W_M) = \int_{z_1} \dots \int_{z_M} \int_{w_1} \dots \int_{w_M} \mathfrak{S}_{l=1}^M \mu_{z_l}(z_l) \wedge \mathfrak{S}_{l=1}^M \mu_{w_l}(w_l) \left/ \frac{\sum_{l=1}^M w_l z_l}{\sum_{l=1}^M w_l} \right. \quad (2.25)$$

où \mathfrak{S} et \wedge indiquent les *t-normes* (*produit* ou *minimum*), $w_l \in W_l$ et $z_l \in Z_l$ pour $l=1, 2, \dots, M$. Les choses se simplifient beaucoup lorsque les fonctions d'appartenance secondaires sont de type intervalle. Dans notre travail, on s'est intéressé à la réduction de type des systèmes flous type-2 intervalle. Dans ce qui suit, nous présentons une procédure algorithmique de calcul de l'ensemble type réduit de ce genre de systèmes où on remarquera que les combinaisons de z_l et w_l ne seront pas toutes considérées.

Pour un système *type-2* intervalle, chaque z_l et w_l ($l=1,2,\dots,M$) dans (2.25) est un ensemble *type-1* intervalle, par conséquent nous avons $\mu_{z_l}(z_l)=\mu_{w_l}(w_l)=1$, donc (2.25) devient :

$$Y(Z_1,\dots,Z_M, W_1,\dots,W_M) = \int_{z_1} \dots \int_{z_M} \int_{w_1} \dots \int_{w_M} 1 \left/ \frac{\sum_{l=1}^M w_l z_l}{\sum_{l=1}^M w_l} \right. = [y_l, y_r] \quad (2.26)$$

Cela signifie que le calcul de l'ensemble type réduit se limite juste au calcul du quotient :

$\frac{\sum_{l=1}^M w_l z_l}{\sum_{l=1}^M w_l}$. Comme tous les degrés d'appartenance dans un ensemble *type-1* intervalle sont

unitaires, nous allons représenter ces ensembles par leurs limites gauches et droites $[l, r]$ ou par leurs centres et largeurs : $c = \frac{l+r}{2}$ et $s = \frac{r-l}{2}$ respectivement.

Dans un système flou *type-2*, chaque z_l dans (2.26) est un ensemble *type-1* intervalle de centre c_l et de largeur s_l ($s_l \geq 0$) et chaque w_l et un ensemble *type-1* intervalle de centre h_l ($l=1,\dots,M$). Y est aussi un ensemble *type-1* intervalle. Donc, nous avons seulement besoin de calculer les deux points extrêmes y_l et y_r . Comme il est montré dans [5], y_l dépend seulement de $(c_l - s_l)$ et sur l'un des deux points extrêmes de W_l . y_r dépend seulement $c_l + s_l$ et sur l'un des deux points extrêmes de W_l . Dans ce qui suit, nous allons citer une procédure itérative pour calculer y_l et y_r .

$$\text{Posons : } S(W_1, \dots, W_M) \cong \frac{\sum_{l=1}^M z_l w_l}{\sum_{l=1}^M w_l} \quad (2.27)$$

où $w_l \in [h_l - \Delta_l, h_l + \Delta_l]$ et $h_l \geq \Delta_l$ pour $(l=1, 2, \dots, M)$ et $z_l \in [c_l - s_l, c_l + s_l]$. Le maximum de S, y_r est obtenu comme suit. On pose $z_l = c_l + s_l$ ($l=1, 2, \dots, M$) et en supposant que les z_l sont arrangés dans l'ordre croissant, c-a-d, $z_1 \leq z_2 \leq \dots \leq z_M$. Alors :

1. posons $w_l = h_l$ pour $(l=1, 2, \dots, M)$ et calculons $S' = S(h_1, \dots, h_M)$ en utilisant (2.27).
2. trouver K ($1 \leq K \leq (M-1)$) tel que $z_k \leq S' \leq z_{k+1}$
3. poser $w_l = h_l - \Delta_l$ pour $l \leq K$ et $w_l = h_l + \Delta_l$ pour $l \geq K+1$ et calculer $S'' = S(h_1 - \Delta_1, \dots, h_k - \Delta_k, h_{k+1} + \Delta_{k+1}, \dots, h_M + \Delta_M)$ en utilisant (2.27).
4. vérifier si $S'' = S'$; si oui, arrêter. S'' est la valeur maximale de $S(W_1, \dots, W_M)$; si non, aller à l'étape 5.
5. poser $S'' = S'$; aller à l'étape 2.

Il a été démontré dans [5] que cette procédure itérative converge dans M itérations au maximum, où une itération est le passage de l'étape 2 à 5. [6].

Le minimum de $S(W_1, \dots, W_M)$ peut être obtenu en utilisant la même procédure juste citée. Seulement deux changements doivent être faits :

- ✓ nous devons mettre $z_l = c_l - s_l$ pour $l=1, 2, \dots, M$.
- ✓ dans l'étape 3 nous devons poser $w_l = h_l + \Delta_l$ pour $l \leq K$ et $w_l = h_l - \Delta_l$ pour $l \geq K+1$ et calculer $S''' = S(h_1 + \Delta_1, \dots, h_k + \Delta_k, h_{k+1} - \Delta_{k+1}, \dots, h_M - \Delta_M)$.

5.1. Réduction de type par la méthode du centroïde

Nous discrétisons l'ensemble de sortie caractérisé par $\mu_{\tilde{B}}(y)$ en N points, y_1, \dots, y_N . Soit $[L_i, R_i]$ le domaine de $\mu_{\tilde{B}}(y_i)$. Pour utiliser la procédure itérative, notons que la somme dans (2.27) est prise de 1 à N au lieu de 1 à M ; les y_i dans (2.19) jouent le rôle des c_l ; $s_l = 0 \forall l$, puisque les y_i sont ordinaires ; $(L_i + R_i)/2 = h_i$ et $(R_i - L_i)/2 = \Delta_i$.

5.2. Réduction de type par la méthode des centres d'ensembles

Dans cette méthode, la procédure itérative s'applique en deux étapes. Dans la première étape, nous calculons les centroïdes C_l des ensembles *type-2* intervalle des

conséquences. Dans la deuxième, nous calculons l'ensemble de type réduit, C_i va jouer le rôle des Z_i dans (2.26). Si le domaine de C_i est l'intervalle $[L_i^c, R_i^c]$, alors $c_i = (L_i^c + R_i^c)/2$ et $s_i = (R_i^c - L_i^c)/2$. Le degré d'activation E_i joue le rôle de W_i . Si le domaine E_i est l'intervalle $[L_i, R_i]$, alors $h_i = (L_i + R_i)/2$ et $\Delta_i = (R_i - L_i)/2$.

Dans chacun des cas précédents, la sortie ordinaire du système flou est obtenue en calculant le point milieu de l'ensemble de type réduit du fait qu'il est un ensemble *type-1* intervalle [6].

Dans ce qui suit, nous allons illustrer, par deux exemples, les résultats de réduction de type d'un ensemble flou *type-2* intervalle moyennant cet algorithme.

- **Exemple 1**

Dans cet exemple, nous allons calculer, en utilisant l'approche (§ 5.1) les ensembles de type réduit pour trois ensembles flous *type-2* intervalle, notés $(\tilde{A}^1, \tilde{A}^2, \tilde{A}^3)$. Ces trois ensembles sont exprimés par leurs fonctions d'appartenance supérieure et inférieure $(\bar{A}^i$ et $\underline{A}^i, i \in \{1, 2, 3\})$, respectivement, de la manière suivante :

$$\begin{aligned} \underline{A}_1 &= 0.75 \exp\left(-0.5\left(\frac{x-2}{1.2}\right)^2\right) & \bar{A}_1 &= \exp\left(-0.5\left(\frac{x-2}{1.2}\right)^2\right) \\ \underline{A}_2 &= 0.75 \exp(-0.5(x-6)^2) & \bar{A}_2 &= \exp(-0.5(x-6)^2) \\ \underline{A}_3 &= 0.75 \exp(-0.5(x-9)^2) & \bar{A}_3 &= \exp(-0.5(x-9)^2) \end{aligned}$$

Ces trois ensembles sont représentés par la figure (2.5.a), en (b) on montre leurs ensembles de type réduit correspondant. Par contre, les valeurs numériques, de ces derniers, sont reportées dans le tableau (2.1), où chaque ensemble de type réduit (centroïde) est représenté par son centre et ses limites gauche et droite.

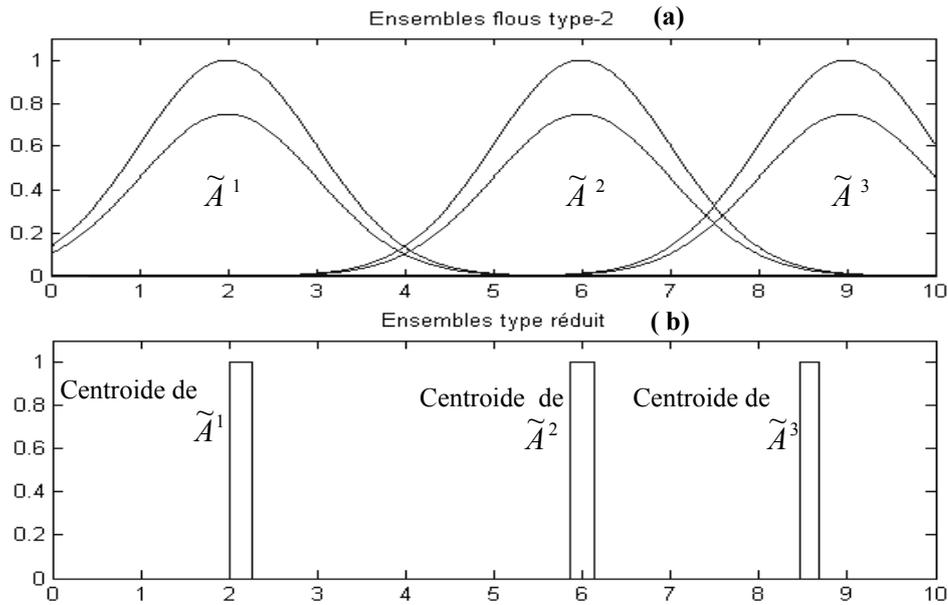


Fig.2.5. Réduction de type par la méthode du centroïde pour l'exemple 1, (a) : Ensembles flous *type-2*, (b) : Ensembles type réduit (*type-1*).

	Centres	Valeurs limites
Ensemble \tilde{A}^1	2.1251	[1.9986 2.2516]
Ensemble \tilde{A}^2	5.9999	[5.8851 6.1146]
Ensemble \tilde{A}^3	8.7132	[8.6203 8.8061]

Tab 2.1. Résultats de réduction de type de l'exemple 1.

• **Exemple 2**

Dans cet exemple, nous allons appliquer la procédure de réduction de type du (§ 5.2) sur des ensembles flous *type-2* intervalle résultants d'une inférence (*max-produit*) dans un système flou *type-2* ayant deux variables d'entrée x_1 et x_2 et une seule variable de sortie y dans l'intervalle $[0, 20]$ (figure 2.6.a, b et c). Tous les ensembles prémisses et conséquences sont de *type-2* intervalle obtenus à partir de gaussiennes à variance fixe et moyenne incertaine (comme dans l'exemple 2 du chapitre 1). Les paramètres de ces ensembles sont donnés par le tableau (2.2).

	Prémises						Conséquences		
	Variable (x_1)			Variable (x_2)					
	\tilde{A}_1^1	\tilde{A}_1^2	\tilde{A}_1^3	\tilde{A}_2^1	\tilde{A}_2^2	\tilde{A}_2^3	\tilde{B}^1	\tilde{B}^2	\tilde{B}^3
m_1	3	9.5	15	4	10	16	3	9.75	16
m_2	4.5	10	16	5	11	17	4	10.25	17
σ	1.5	2.5	1.5	2	1.5	2	2.5	1.5	2.5

Tab.2.2. Paramètres des fonctions d'appartenance de l'exemple 2.

Les règles floues ont la forme (2.28) suivante :

$$R^{(i)} : \text{Si } x_1 \text{ est } \tilde{A}_1^i \text{ et } x_2 \text{ est } \tilde{A}_2^i \text{ alors } y \text{ est } \tilde{B}^i \quad (2.28)$$

L'entrée appliquée au système flou ainsi définie est $X = [6 \ 2]^T$. Elle a deux appartenances non nulles \tilde{A}_1^1 et \tilde{A}_1^2 pour x_1 et une autre pour x_2 \tilde{A}_2^1 figure (a et b de la page suivante), par conséquent, les deux règles suivantes sont activées.

$$R^{(1)} : \text{Si } x_1 \text{ est } \tilde{A}_1^1 \text{ et } x_2 \text{ est } \tilde{A}_2^1 \text{ alors } y \text{ est } \tilde{B}^1 \quad \text{ou}$$

$$R^{(2)} : \text{Si } x_1 \text{ est } \tilde{A}_1^2 \text{ et } x_2 \text{ est } \tilde{A}_2^1 \text{ alors } y \text{ est } \tilde{B}^1$$

En (d), on représente l'ensemble de type réduit ($[y_l, y_r] = [3.0008, 4.0156]$) ainsi que la sortie defuzzifiée (ordinaire) ($y_{cos} = 3.5082$) où nous avons pris 2000 échantillons.

6. Conclusion

Ce chapitre est une étude détaillée des systèmes flous type-2. Parmi les points que nous avons traité on peut citer : les opérations sur les ensembles flous type-2, qui consiste une généralisation des opérations des ensembles flous type-1, la structure des systèmes et contrôleurs flous type-2, et surtout, d'une manière plus détaillée, le traitement de la sortie de ces derniers qui exige d'abord la réduction de type puis la defuzzification. Pour ne pas rester dans le cadre descriptif, nous avons fait état de deux exemples d'application. Dans le chapitre suivant, nous allons appliquer les contrôleurs flous type-1 et type-2 sur des systèmes non linéaires. Les résultats seront comparés à chaque fois, avant de tirer les conclusions.

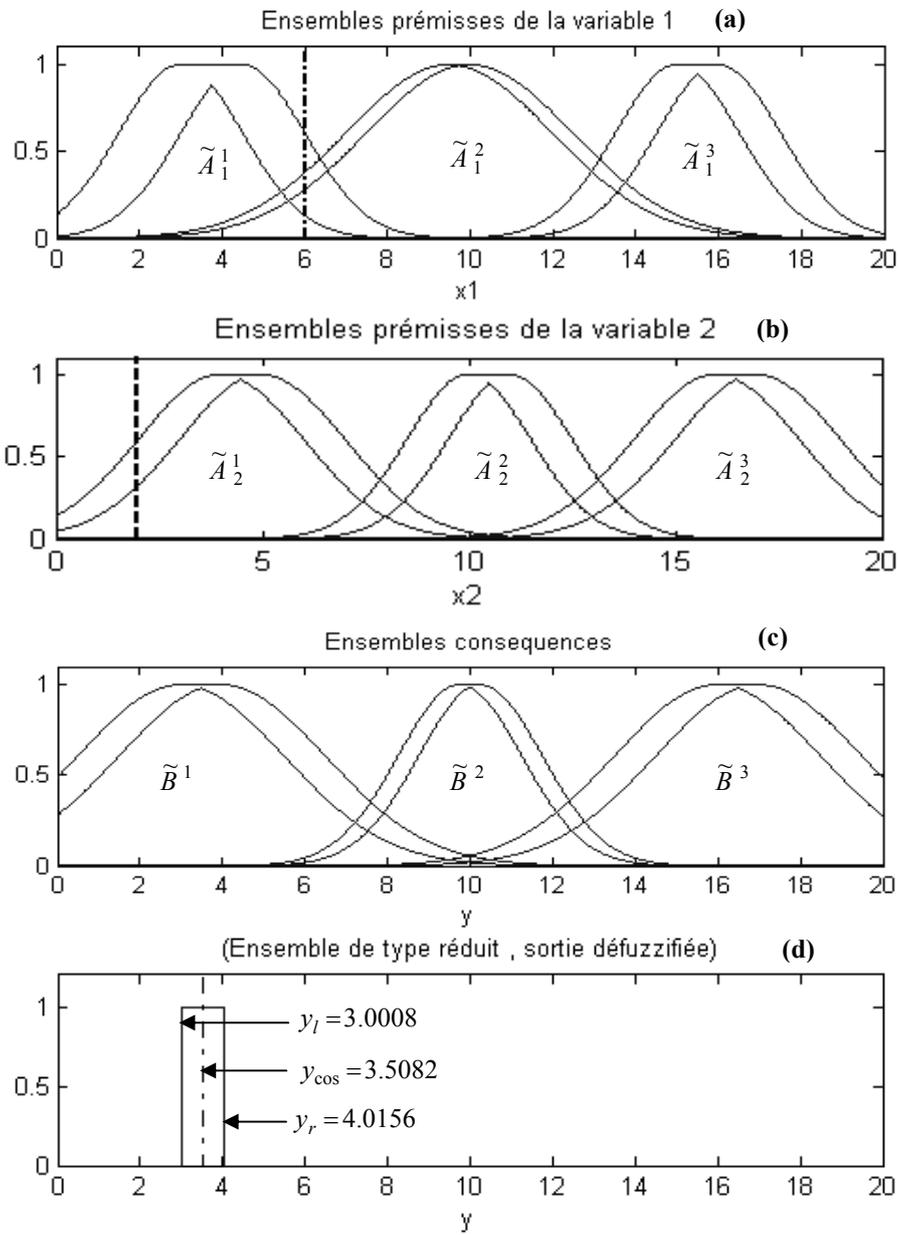


Fig 2.6. (a) et (b) : Ensembles flous antécédents, (c) : Ensembles flous conséquences, (d) : Ensemble type réduit de l'exemple 2.

Chapitre 3

Application des systèmes flous *type-1* et *type-2* à la commande des systèmes non linéaires.

1. Introduction

L'incertitude peut affecter les décisions d'un expert et elle apparaît sous différentes formes. Le concept de l'information est complètement lié au concept de l'incertitude. L'aspect fondamental de cette liaison c'est que l'incertitude est impliquée dans chaque situation de résolution d'un problème de décision. Elle peut être le résultat de la défaillance de l'information qui peut être incomplète, imprécise, non complètement fiable, vague, contradictoire ou insuffisante dans d'autres cas [17] [23]. Le raisonnement flou est introduit dans le but de prendre en compte ces incertitudes. Les systèmes flous utilisant des ensembles flous *type-1* représentent les grandeurs par des valeurs d'appartenance appartenant à l'intervalle $[0, 1]$. Cependant, en présence d'incertitudes, la définition des prémisses et des conséquences devient difficile. Les systèmes flous *type-2* prennent en compte ces incertitudes. Car elles utilisent des ensembles flous *type-2* dans lesquels les valeurs d'appartenance sont elles-mêmes des ensembles flous *type-1* [6].

Dans cette optique, de nombreuses applications utilisant des systèmes flous *type-2* sont envisageables rien qu'en ce qui concerne les systèmes non linéaires et leurs commande. Nous présentons, dans ce chapitre, deux applications des systèmes flous *type-2* et nous nous garderons de faire des conclusions trop hâtives en ce qui concerne la supériorité des systèmes flous *type-2* par rapport à ceux du *type-1*. Ce n'est pas parce qu'une technique est nouvelle qu'elle est forcément plus efficace.

2. Commande non linéaire floue

Les procédés dynamiques non linéaires sont très complexes et peuvent présenter toute sorte de comportements étranges (voir chaotique [15]). Contrairement à la commande linéaire, la commande non linéaire ne dispose pas de techniques universelles capables de présenter des résultats généraux pour l'analyse et la synthèse des systèmes non linéaires. D'une manière générale, l'analyse et la commande non linéaires sont des problèmes difficiles et la majorité des méthodes présentées dans la littérature sont restreintes à des classes particulières de systèmes non linéaires [15].

L'objectif principal d'une loi de commande est d'imposer au procédé commandé des propriétés et des performances désirées. La logique floue a été utilisée avec succès dans plusieurs domaines. Souvent, la structure de commande floue de la figure (3.1) est utilisée [20][21][22][23][24].

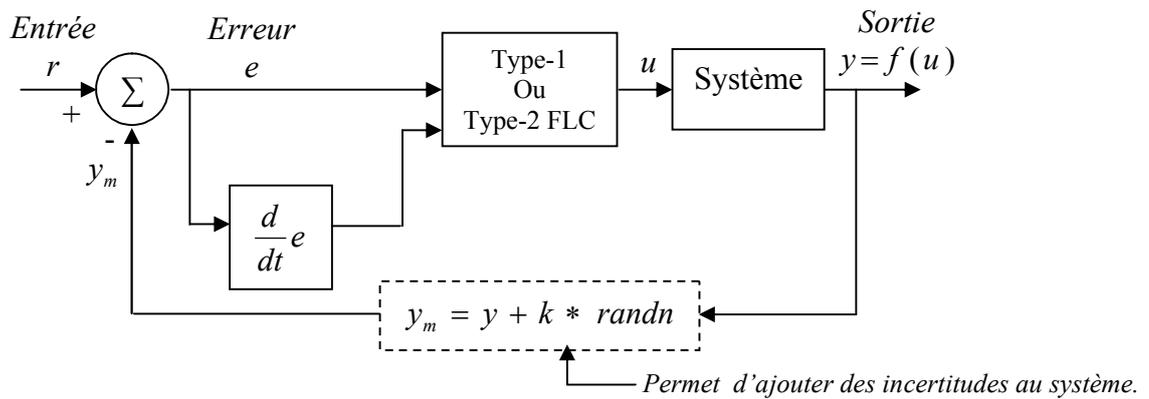


Fig.3.1. Structure de commande utilisée dans les tests expérimentaux.

3. Etude des incertitudes numériques [25]

3.1. Contexte et objectif :

Cette application met en évidence les avantages de l'utilisation des ensembles flous *type-2* dans l'implémentation des contrôleurs flous (FLC). En effet, les différents instruments (amplificateurs, détecteurs, convertisseurs analogique /numérique et numérique /analogique, etc.) introduisent quelques valeurs imprédictibles dans l'information collectée. Toutes ces

incertitudes se traduisent au niveau de la base des règles floues par des fonctions d'appartenance des prémisses et des conséquences incertaines [4]. Nous allons illustrer comment utiliser un FLC *type-2* pour minimiser l'effet de ces incertitudes. Nous allons confirmer ceci par des mesures quantitatives des erreurs.

3.2. Critères de performance

Pour quantifier les erreurs, nous utilisons trois critères de performance largement utilisés, à savoir : l'intégral de l'erreur au carré (ISE), l'intégral de la valeur absolue de l'erreur (IAE) et l'intégral du temps multiplié par la valeur absolue de l'erreur (ITAE) [23]. Ces trois critères sont définis comme suit :

- 1) Intégral de l'Erreur au Carré (ISE).

$$ISE = \int_0^{+\infty} e^2 dt \quad (3.1)$$

- 2) Intégral de la valeur Absolue de l'Erreur (IAE).

$$IAE = \int_0^{+\infty} |e| dt \quad (3.2)$$

- 3) Intégral du Temps multiplié par la valeur Absolue de l'Erreur (ITAE).

$$ITAE = \int_0^{\infty} t |e| dt \quad (3.3)$$

Le choix du critère à utiliser dépend du type de la réponse désirée. L'erreur contribue différemment dans chaque critère. Le critère ISE est important dans le cas des réponses avec des dépassements minimums lorsque la charge varie. Le critère ITAE implique la variable temps et, par conséquent, pénalise les erreurs de plus en plus à travers le temps. Le critère IAE permet une conception intermédiaire par rapport aux deux premiers [23].

3.3. Simulation et résultats

La structure de commande de la figure (3.1) a été simulée en utilisant le logiciel *Matlab*. Le processus commandé est caractérisé par le modèle suivant :

$$y(i) = 0.14.y(i-3) + 0.9.y(i-2) + 0.005.u(i-1) + 0.5.u(i-2) \quad (3.4)$$

L'objectif est d'assurer une réponse la plus proche possible de la trajectoire de référence. Les performances des deux contrôleurs flous (*type-1* et *type-2*) sont testées en considérant les deux cas suivants :

- ✓ Système idéal sans aucune source d'incertitudes (tests 1 et 2).
- ✓ Système en présence d'incertitudes (tests 3 et 4).

Dans le premier cas, la sortie du système mesurée est supposée sans incertitudes (nette) $y_m(i) = y(i)$, ($k=0$). Par contre dans le second cas, la mesure est supposée entachée d'un bruit aléatoire $b(i)$ de distribution normale.

$$y_m(i) = y(i) + b(i) = y(i) + k * randn \quad (3.5)$$

On teste le système en utilisant une entrée de référence constante, $r(i) = 1$, avec ($i = 4, 5, 6, \dots$). La comparaison des performances des deux contrôleurs (*type-1* et *type-2*) est faite en utilisant les critères ISE, IAE, ITAE. Dans le tableau 3.2 nous récapitulons les valeurs obtenus pour chaque critère en considérant 100 échantillons du signal et un temps d'échantillonnage de 0.1s.

Pour les tests (1 et 2), où le signal d'erreur $e(t)$ est l'écart entre le signal de référence et la sortie réelle du système. Dans les tests (3 et 4), on suppose que la sortie mesurée est bruitée. Par conséquent, le signal d'erreur $e(t)$ et sa variation $de(t)$ sont aussi bruités.

Contrôleur flou type-1 :

Pour chacune des deux entrées du contrôleur flou *type-1*, nous définissons trois fonctions d'appartenance de formes gaussiennes caractérisant les valeurs linguistiques (négative, zéro, positive) sur les univers de discours $[-10,10]$ pour l'erreur et $[-16,16]$ pour la variation d'erreur. Les fonctions de la première variable sont centrées en -10, 0, 10 respectivement et celles de la deuxième variable en -16, 0, 16. L'écart-type de toutes les

fonctions est choisi égal à 5. Les valeurs linguistiques de la variable de sortie (NG : négative grande, N : négative, Z : zéro, P : positive et PG : positive grande) sont représentées par des fonctions singleton sur l'intervalle $[-30, 30]$. Elles ont comme centres : -30, -16.5, 0, 16.5 et 30, respectivement. Le tableau (3.1.a) résume les paramètres des fonctions d'appartenance utilisées. La méthode de defuzzification utilisée est celle du centre de gravité (CG). La table de décision considérée est celle du tableau 1.1 pour les deux types de systèmes.

Contrôleur flou type-2 :

Dans le cas du FLC *type-2*, pour chaque variable d'entrée, nous définissons trois fonctions d'appartenance floues *type-2* gaussiennes : négative, zéro, positive. Ces trois ensembles sont tous de largeur fixe. La variable de sortie possède cinq ensembles flous dans l'intervalle $[-30, 30]$, comme dans le cas du *type-1*. Pour le contrôleur flou *type-2*, tous les ensembles d'entrée/sortie sont des ensembles flous *type-2* intervalle dont les paramètres sont donnés dans le Tableau 3.1.b. L'ensemble de type réduit et la sortie defuzzifiée sont obtenus par la méthode du centroïde.

(a)				(b)			
Variables	Terme	Centre (<i>m</i>)	Ecart- Type (<i>sigma</i>)	Variables	Terme	Centres [<i>m</i> ₁ , <i>m</i> ₂]	Ecart- Type (<i>sigma</i>)
Erreur <i>e</i>	négative	-10	5	Erreur <i>e</i>	négative	[-10.25, -9.75]	5
	zéro	0	5		zéro	[-0.25, 0.25]	5
	positive	10	5		positive	[9.75, 10.25]	5
Variation de l'erreur <i>de/dt</i>	négative	-16	5	Variation de l'erreur <i>de/dt</i>	négative	[-16.25, -15.75]	5
	zéro	0	5		zéro	[-0.25, 0.25]	5
	positive	16	5		positive	[15.75, 16.25]	5
Sortie <i>u</i>	NG	-30		Sortie <i>u</i>	NG	[-30.25, -29.75]	
	N	-16.5			N	[-16.75, -16.25]	
	Z	0			Z	[-0.25, 0.25]	
	P	16.5			P	[16.25, 16.75]	
	PG	30			PG	[29.75, 30.25]	

Tab 3.1. Caractéristiques des ensembles d'entrée / sortie, systèmes de type singleton :

(a): système flou *type-1*, (b) : système flou *type-2*.

✓ **Test1.** En absence du bruit en utilisant FLC type-1 :

Dans ce test, la sortie mesurée est supposée sans bruit. La réponse du système est illustrée par la figure (3.2.a). On constate que la réponse suit la référence. Les valeurs quantitatives des critères ISE, IAE et ITAE sont portées sur le tableau (3.2) et leurs évolutions sont illustrées par les figures (3.2 : b, c et d).

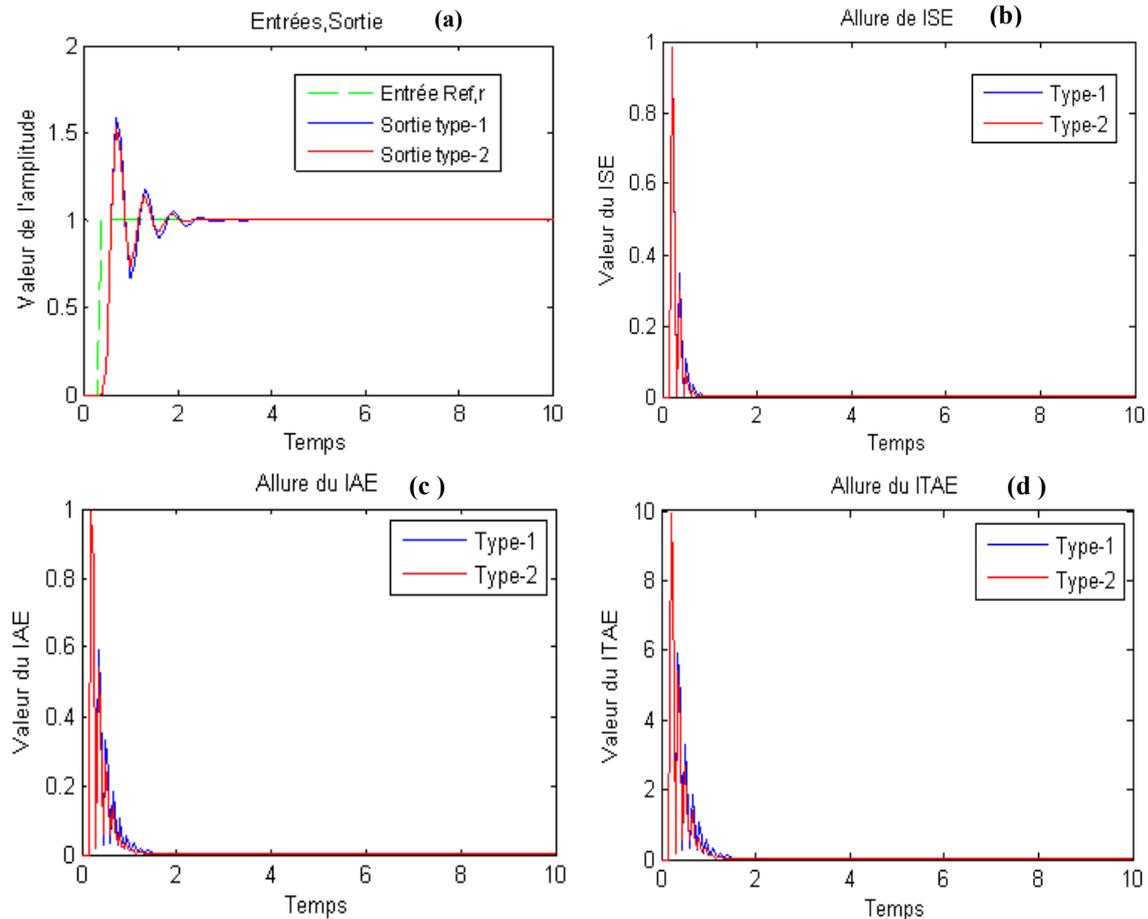


Fig. 3. 2. Résultats de simulation : (a) réponse du système FLC *type-1* et FLC *type-2* en absence du bruit, (b) : Allure de ISE, (c) Allure : de IAE, (d) Allure : de ITAE.

✓ **Test 2.** En absence du bruit en utilisant FLC type-2 :

Nous utilisons les mêmes conditions pour tester le FLC *type-2*. La réponse du système est représentée aussi sur la figure (3.2.a). Les valeurs des trois critères de performance sont portées sur le tableau (3.2). Nous constatons que la réponse est très proche de celle obtenue avec FLC *type-1*, mais avec des erreurs plus faibles pour le FLC *type-2*. Ce qui constitue un léger avantage par rapport au FLC *type-1*.

✓ **Test 3.** En présence du bruit en utilisant FLC type-1:

Dans ce cas, nous avons simulé le système de l'équation (3.4) avec un bruit aléatoire en fixant ($k = 0.3$). La figure (3.3.a) représente la réponse du système avec bruit de mesure. Les évolutions des critères sont représentées sur les figures (b, c, d).

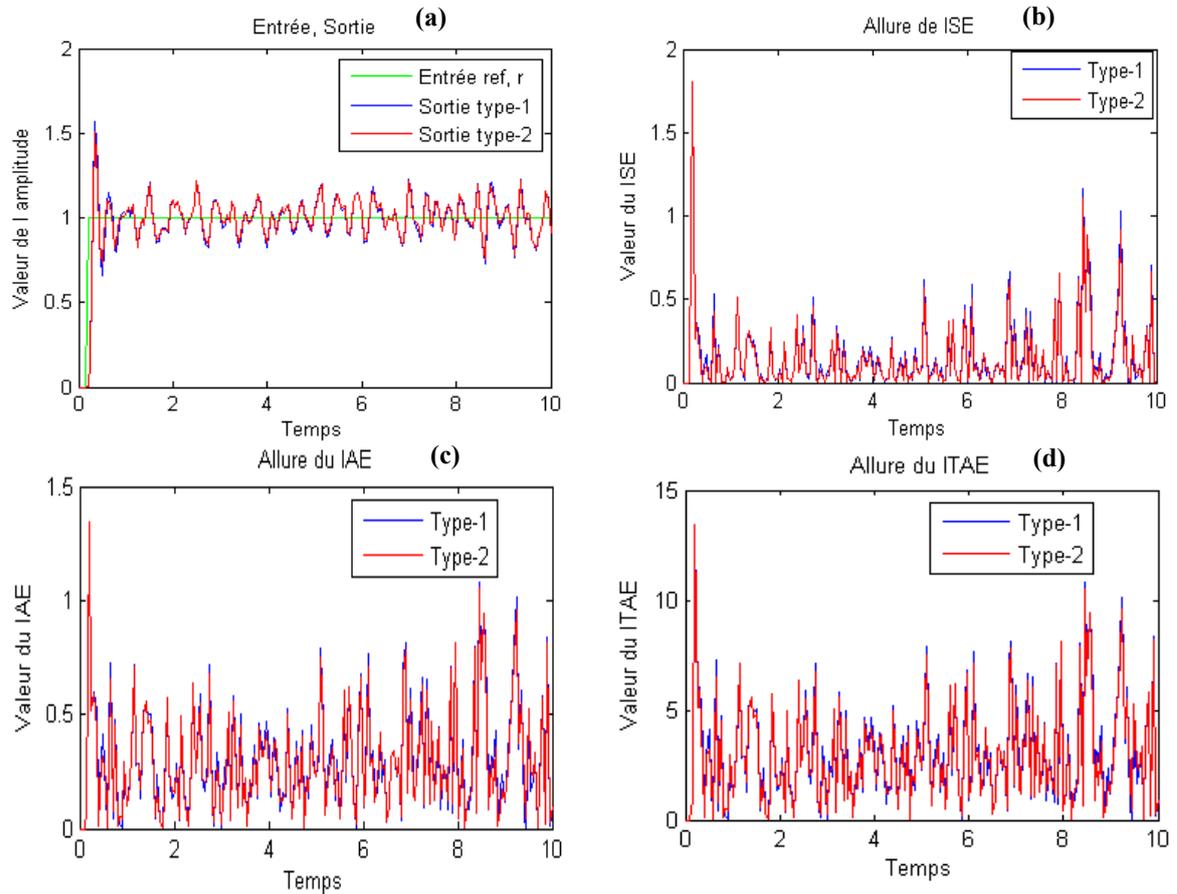


Fig.3.3. Résultats de simulation : (a) réponse du système FLC type-1 et FLC type-2 en présence du bruit, (b) : Allure de ISE, (c) Allure : de IAE, (d) Allure : de ITAE.

✓ **Test 4.** En présence du bruit en utilisant FLC type-2 :

Le FLC *type-2* est utilisé dans les mêmes conditions du test 3. Et nous confirmons ces avantages par rapport à FLC *type-1* utilisé dans l'expérience 3. Les évolutions des différentes grandeurs sont reportées sur les figures (3.3: a, b, c, d).

Critères de performance	FLC TYPE-1		FLC TYPE-2	
	En absence du bruit	En présence du bruit	En absence du bruit	En présence du bruit
ISE	2.7292	12.3080	2.2545	11.4817
IAE	4.5916	27.6409	3.9478	26.6015
ITAE	45.9160	276.4092	39.4778	266.0148

Tab. 3.2. Comparaison des critères de performance entre le contrôleur flou *type-1* et *type-2*. Valeurs obtenues pour 100 échantillons.

Commentaires :

La structure du FLC *type-2*, incorporant des fonctions avec une dimension supplémentaire, leur permet une meilleure prise en compte des incertitudes. Mais, ceci au prix d'un degré plus élevé de complexité. Le problème principal de ces systèmes est la réduction d'ordre. Dans notre cas, nous avons considérés des fonctions d'appartenance *type-2* intervalle, relativement plus faciles à manipuler.

Pour une comparaison meilleure, trois critères de performances ont été utilisés (ISE, IAE, ITAE). Les résultats de simulation montrent une légère supériorité des FLS *type-2* par rapport aux FLS *type-1*. Mais, l'écart reste minime même en présence du bruit.

4. Etude des incertitudes linguistiques

Les systèmes flous sont constitués de règles. La connaissance utilisée pour construire ces règles est de nature incertaine [15]. Cette incertitude mène alors à des règles dont les prémisses ou les conséquences sont incertaines.

L'objectif de cette application est la simulation des incertitudes linguistiques contenues dans la base de règles d'un système flou. Pour cela, on a eu recours à une application typique qui est le contrôle du pendule inversé de la figure (3.4).

4.1. Modélisation et objectif de la commande :

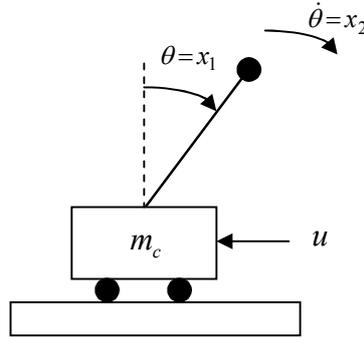


Fig.3.4. Représentation du pendule inversé.

Le modèle dynamique du pendule inversé est donné par :

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f(x_1, x_2) + g(x_1, x_2)u \end{aligned} \quad (3.6)$$

$$\text{où : } f(x_1, x_2) = \frac{g \sin x_1 - \frac{mlx_2^2 \cos x_1 \sin x_1}{m_c + m}}{l \left(\frac{4}{3} - \frac{m \cos^2 x_1}{m_c + m} \right)} \quad ; \quad g(x_1, x_2) = \frac{\cos x_1}{m_c + m} \frac{1}{l \left(\frac{4}{3} - \frac{m \cos^2 x_1}{m_c + m} \right)}$$

avec m_c est la masse du chariot, m est la masse du pole, $2l$ est la longueur du pole, g est l'accélération de la pesanteur et u est la force appliquée. Dans notre cas, on considère $m_c = 0.5 \text{ kg}$, $m = 0.2 \text{ kg}$, $l = 0.3 \text{ m}$ et $g = 10 \text{ m.s}^{-2}$. [26]

L'objectif est de concevoir deux contrôleurs flous (*type-1* et *type-2*) capables de ramener le pendule à son état d'équilibre (stabilité critique) $x_1 = x_2 = 0$. Une structure de contrôle systématique et pratique a été proposée dans [26]. Elle est similaire à celle de la figure (3.1), la seule différence réside dans la chaîne de retour où un gain proportionnel est introduit (voir figure 3.5).

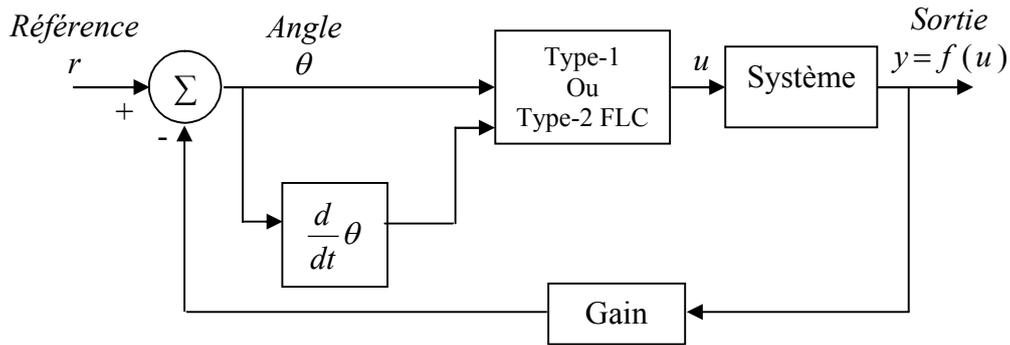


Fig.3.5. Structure de commande utilisée pour le contrôle du pendule inversé.

4.2. Analyse de la stabilité et conception des FLC :

La stabilité est l'une des aspects les plus importants dans le domaine de l'automatique. Dans cette application, la stabilité de la boucle est analysée par l'approche de Lyapunov dont la philosophie de base est l'extension d'une observation physique fondamentale : 'Si l'énergie totale d'un système est continuellement dissipée, alors le système convergera vers un état d'équilibre'.

Nous supposons que l'équation exacte du système de la figure (3.4) est inconnue et que nous avons seulement les informations partielles suivantes [26]:

1. Le système possède deux degrés de libertés $\theta = x_1$ et $\dot{\theta} = \dot{x}_1$ respectivement. Donc $\dot{x}_1 = x_2$.
2. \dot{x}_2 est proportionnelle à u , c-à-d, lorsque u augmente (diminue) \dot{x}_2 augmente (diminue).

Nous choisissons comme fonction de Lyapunov V tel que :

$$V(x_1, x_2) = \frac{1}{2}(x_1^2 + x_2^2) \quad (3.7)$$

Il est clair que V est une fonction définie positive. En dérivant V on obtient :

$$\dot{V} = x_1 \dot{x}_1 + x_2 \dot{x}_2 = x_1 x_2 + x_2 \dot{x}_2 \quad (3.8)$$

Donc, on impose :

$$x_1 x_2 + x_2 \dot{x}_2 < 0 \quad (3.9)$$

au voisinage de $(0, 0)^T$.

Nous pouvons, maintenant dériver suffisamment de conditions pour que (3.9) soit satisfaite. Si x_1 et x_2 ont des signes opposés alors $x_1 x_2 < 0$ et (3.9) sera vérifiée si $\dot{x}_2 = 0$; si

x_1 et x_2 sont positifs, (3.9) sera vérifiée si $\dot{x}_2 < -x_1$; et si x_1 et x_2 sont négatifs, (3.9) est satisfaite si $\dot{x}_2 > -x_1$.

On peut traduire les conditions précédentes aux règles floues suivantes :

- Si x_1 est positif (P) et x_2 est positif (P) alors \dot{x}_2 est négatif grand (NG).
- Si x_1 est négatif (N) et x_2 est négatif (N) alors \dot{x}_2 est positif grand (PG).
- Si x_1 est positif (P) et x_2 est négatif (N) alors \dot{x}_2 est zéro (Z).
- Si x_1 est négatif (N) et x_2 est positif (P) alors \dot{x}_2 est zéro (Z).

Cependant, connaissant que \dot{x}_2 est proportionnel à u , on peut remplacer \dot{x}_2 par u dans chacune des règles précédentes. Néanmoins, et après plusieurs tests, la base de règles du tableau 1.1 à été adoptée.

Pour compléter la structure des contrôleurs flous, on doit exprimer chaque terme linguistique de la base de règles par un ensemble flou, puis, on doit définir la méthode d'inférence floue [26].

Pour chacune des deux entrées du contrôleur flou *type-1*, nous définissons trois fonctions d'appartenance de formes gaussiennes caractérisant les valeurs linguistiques (négative, zéro, positive) (tableau 3.3.a).

(a)				(b)			
Variables	Terme	Centre (m)	Ecart-Type (σ)	Variables	Terme	Centres [m_1, m_2]	Ecart-Type (σ)
Angle θ	négative	-0.6	0.35	Angle θ	négative	[- 0.7 , - 0.5]	0.35
	zéro	0	0.35		zéro	[- 0.1 , 0.1]	0.35
	positive	0.6	0.35		positive	[0.5 , 0.7]	0.35
Dérivée de l'angle $\dot{\theta}$	négative	-1.2	0.45	Dérivée de l'angle $\dot{\theta}$	négative	[-1.3 , -1.1]	0.45
	zéro	0	0.45		zéro	[- 0.1 , 0.1]	0.45
	positive	1.2	0.45		positive	[1.1 , 1.3]	0.45
Sortie (Force) u	NG	-60		Sortie (Force) u	NG	[- 62 , - 58]	
	N	-30			N	[- 32 , - 28]	
	Z	0			Z	[- 2 , 2]	
	P	30			P	[28 , 32]	
	PG	60			PG	[58 , 62]	

Tab 3.3. Caractéristiques des ensembles d'entrée / sortie : (a) système flou *type-1*, (b) système flou *type-2*, les systèmes sont de type singleton.

Dans le cas du FLC *type-2*, pour chaque variable d'entrée nous définissons trois fonctions d'appartenance floues *type-2* : négative, zéro, positive. Par contre, La variable de sortie possède cinq ensembles flous, aussi bien dans le cas du *type-1* que dans le cas du *type-2*. Pour le contrôleur flou *type-2*, tous les ensembles d'entrée/sortie sont des ensembles flous *type-2* intervalle.

La méthode d'inférence est *max-produit*, par contre, la méthode de réduction de type est celle des centres d'ensembles exposée dans (§ 5.2) du chapitre précédent.

4.3. Résultats de simulation :

Les conditions initiales sont : $\theta=0 \text{ rad}$ et $\dot{\theta}=0.2 \text{ rad/s}$. La figure (3.6) donne les résultats de simulation du système (3.6) en utilisant le contrôleur flou *type-1*. On constate bien que le système retrouve son état équilibre dans un temps $t_r \approx 1.5 \text{ s}$ (figure 3.6.a). La figure (b) montre l'évolution de la fonction donnée par (3.8) qui est toujours négative. Ce qui signifie que le système est stable. Les figures (3.6.c) et (3.6.d) donnent, respectivement, les allures de la vitesse angulaire ainsi que celle de la force appliquée en fonction du temps.

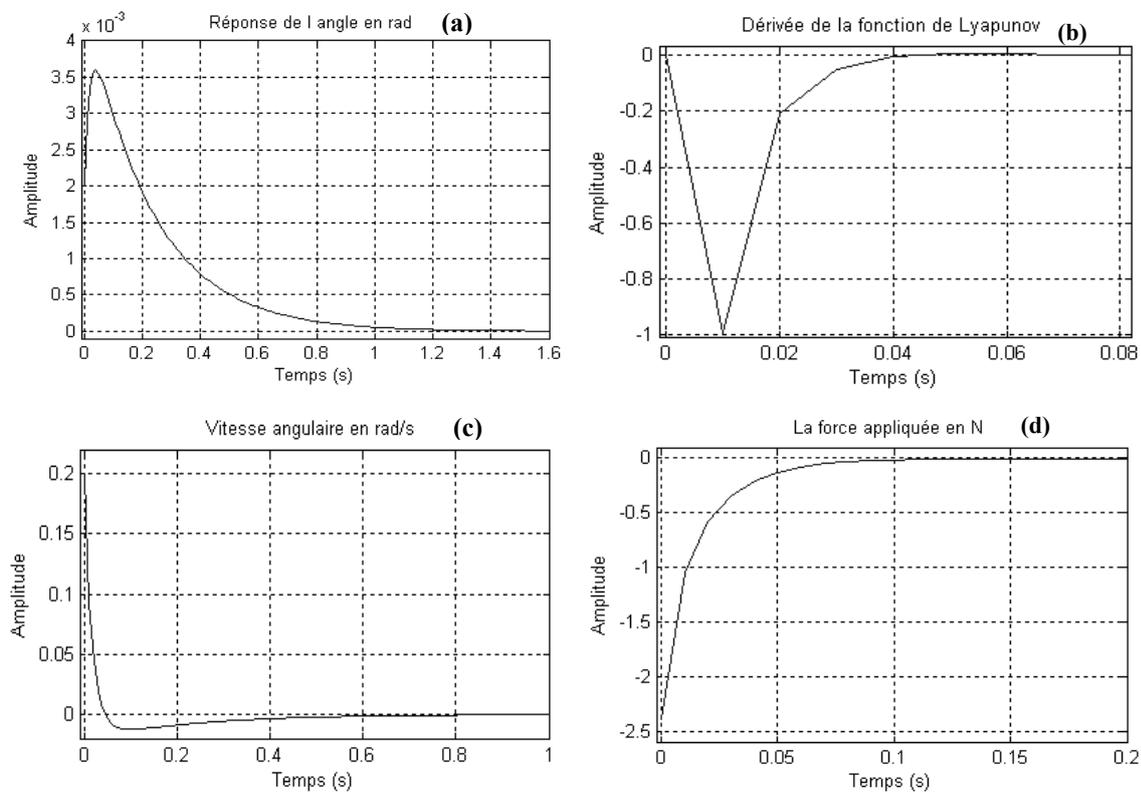


Fig. 3.6. Résultats de simulation du pendule inversé en utilisant le contrôleur flou *type-1* : (a) : Réponse de l'angle, (b) : Dérivée de la fonction de Lyapunov, (c) : Vitesse angulaire et (d) : La force appliquée.

La figure (3.7) montre les résultats de simulation du système (3.6) en utilisant le contrôleur flou *type-2*, où la position d'équilibre est atteinte en un temps $t_r \approx 1s$ (inférieur à 1.5 s), (voir figure 3.7.a). La figure (3.7.b) montre l'évolution de la fonction donnée par (3.8) qui est toujours négative. Par conséquent le système est stable. Les figures (3.7.c) et (3.7.d) montrent, respectivement, les allures de la vitesse angulaire ainsi que celle de la force appliquée en fonction du temps.

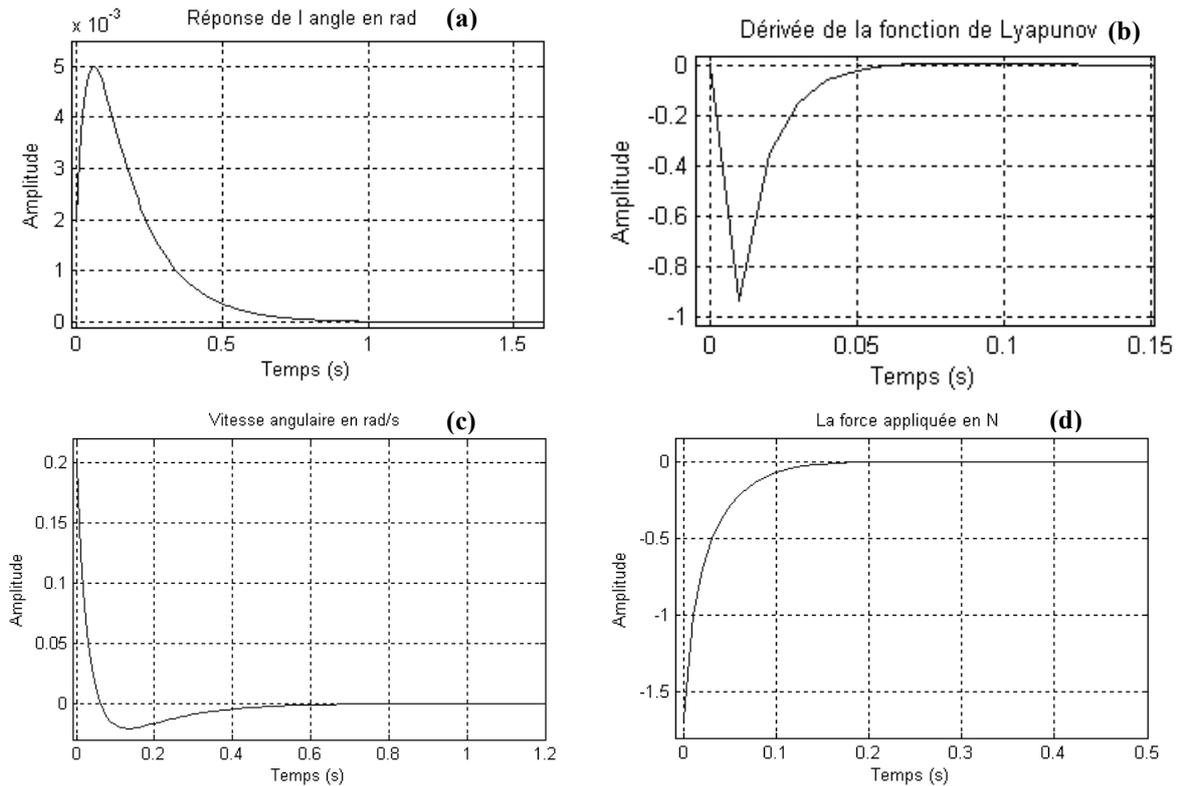


Fig.3.7. Résultats de simulation du pendule inversé en utilisant le contrôleur flou *type-2* : (a) : Réponse de l'angle, (b) : Dérivée de la fonction de Lyapunov, (c) : Vitesse angulaire et (d) : La force appliquée.

Cas de conséquences non singletons :

En vue d'obtenir la réponse du système (3.6), dans le cas des systèmes flous non singleton, nous allons considérer les deux contrôleurs flous précédents avec une modification des fonctions d'appartenance des conséquences qui ne sont plus des singletons mais des gaussiennes avec une valeur de l'écart-type fixe égale à 10. Les deux réponses du système sont montrées par la figure (3.8. a et b). On remarque, ici, une fois de plus, que le temps de réponse obtenu avec le contrôleur flou *type-2* est inférieur à celui du *type-1*.

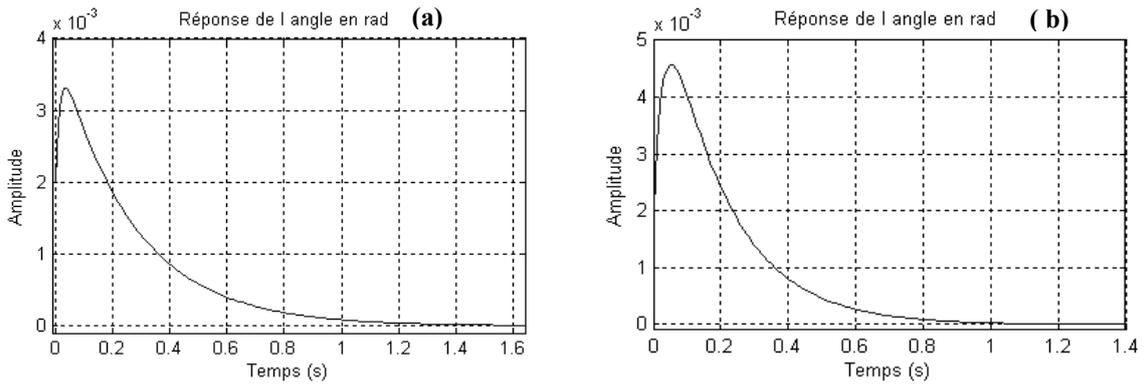


Fig.3.8. Résultats de simulation du pendule inversé en utilisant : (a) le contrôleur flou *type-1*, (b) : le contrôleur flou *type-2*, de type non singleton.

Cas de contrôleur type-2 singleton avec FOU variable:

Le test suivant consiste à faire varier le *FOU* dans le cas du contrôleur flou *type-2* singleton pour déterminer comment ce paramètre peut influencer sur les performances du système notamment sur la stabilité [26] (pour *FOU*, voir définition 7 en annexe).

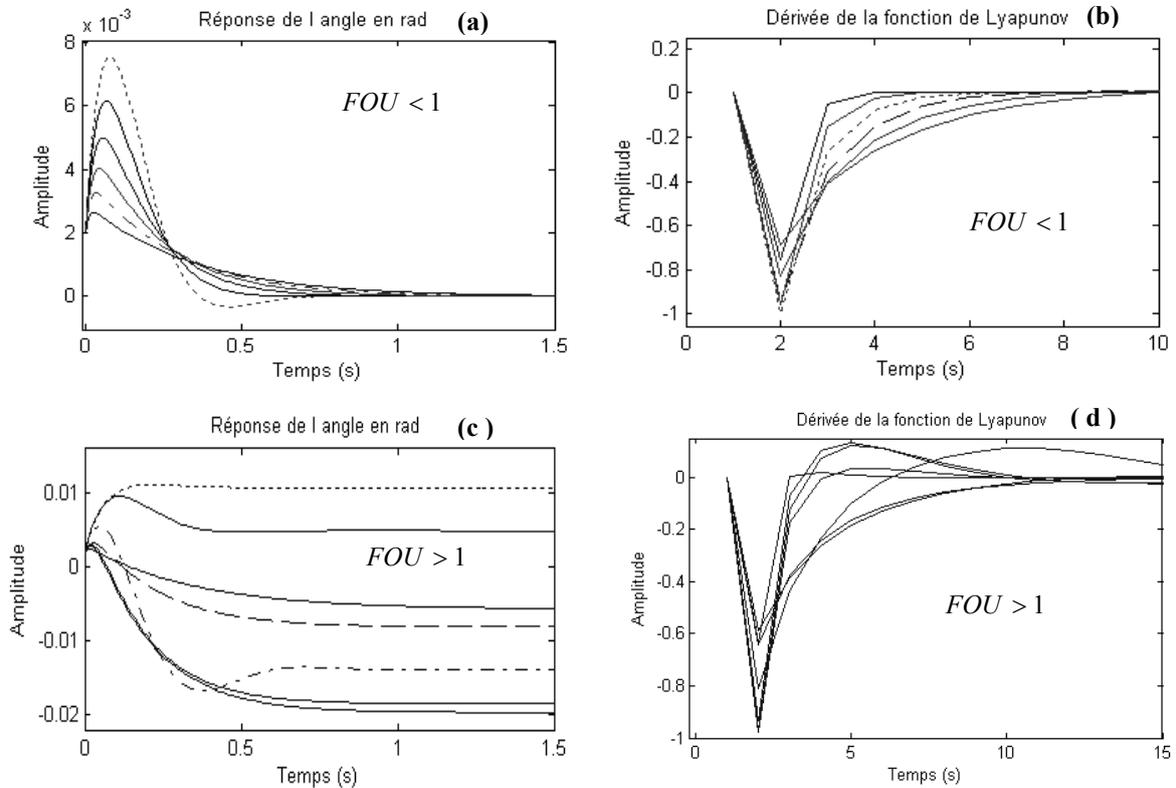


Fig.3.9. Résultats de simulation du pendule inversé en utilisant le contrôleur flou *type-2* : (a) et (c) : Réponse de l'angle (b) et (d) : Dérivée de la fonction de Lyapunov en variant le *FOU*.

Pour des valeurs du $FOU \in [0 \ 1]$, le système retrouve son équilibre en un temps inférieur à $t_r \approx 1.5 \text{ s}$ (figure 3.9.a), tout en gardant sa stabilité (figure 3.9.b). Par contre, pour des valeurs du $FOU > 1$ le système ne peut retrouver son équilibre (figure 3.9.c) et l'allure de (3.8) donnée par la figure (3.9.d) n'est pas toujours négative, ce qui revient à dire que le système est instable.

5. Conclusion

D'après les résultats de simulations obtenus, on peut dire que la structure de commande donnée par la figure (3.5) est valable aussi bien que pour les contrôleurs flous *type-1* que ceux du *type-2*. Néanmoins, les ensembles flous utilisés dans ces derniers possèdent un domaine (FOU) qui peut être modulé, et ce, sans perte de stabilité si on donne une valeur convenable au gain proportionnel de la chaîne de retour. Dans le cas du pendule inversé, le système garde sa stabilité pour des valeurs du $FOU \in [0 \ 1]$ et une valeur du gain égale à l'unité. L'explication est que ce domaine $FOU \in [0 \ 1]$ garantit la stabilité du système car on a utilisé des ensembles flous *type-2* intervalle qui sont eux même définis sur l'intervalle $[0 \ 1]$ [25].

Dans le chapitre suivant, le relevé d'expertise sera traité. Il s'agit d'utiliser un algorithme génétique (AG) pour l'ajustement optimal des facteurs d'échelle, des fonctions d'appartenance de la base de règles intervenant dans la synthèse des contrôleurs flous.

Chapitre 4

Optimisation des contrôleurs flous *type-1* et *type-2* par les *AG* :

Application à la commande d'un robot mobile.

1. Introduction

L'objectif principal de ce chapitre est de concevoir un contrôleur flou *type-2* optimisé par les algorithmes génétiques. Les performances de ce contrôleur hybride (*CFLT2/AG*) seront évaluées à travers un exemple de simulation qui portera sur la commande d'un robot mobile. Les résultats seront comparés à la fois avec ceux d'un contrôleur flou *type-2* dont les paramètres sont obtenus après plusieurs tests empiriques, ainsi qu'avec ceux d'un contrôleur hybride à base de la logique floue *type-1* (contrôleur flou *type-1* dont les paramètres sont optimisés par les *AG* : *CFLT1/AG*). Et ce, en absence et en présence de perturbations.

Le chapitre est organisé en trois parties :

Dans la première partie, nous donnerons un aperçu général sur la robotique mobile, toute en se focalisant sur le contrôle flou des robots mobiles à roues non holonomes. Les études et les applications récemment apparues serviront comme références pour notre application [27] [28] [29] [30].

Dans la deuxième partie, on exposera la technique d'optimisation : algorithme génétique (*AG*). Le principe ainsi que les différents opérateurs relatifs à son fonctionnement seront abordés.

La troisième partie concerne les résultats de simulation obtenus en utilisant lesdits contrôleurs. Nous verrons comment les *AG* peuvent améliorer les performances des contrôleurs flous *type-1* et *type-2*, et ce, en optimisant que les paramètres des fonctions d'appartenance des prémisses et des conséquences.

2. La robotique mobile

2.1. Introduction

Depuis quelques années, un intérêt croissant est porté au sein de la communauté robotique au développement de systèmes intelligents autonomes dans le cadre de la robotique mobile. Un tel intérêt peut être perçu comme une conséquence logique à l'apparition d'applications potentielles (nettoyage, manutention ou encore de l'aide à la mobilité de personnes âgées ou handicapées) et au désir de mettre les robots sur des tâches nouvelles telles que les opérations d'intervention sur sites accidentés, la manipulation sur sites sensibles en l'occurrence nucléaires, ou l'exploration de sites maritimes ou planétaires [31].

Dans de tels contextes, et tout comme dans le cas des applications traditionnelles de la robotique mobile (ralliement de cible, évitements d'obstacles...etc.) munir un robot mobile d'une capacité décisionnelle lui permettant de naviguer d'une façon autonome est une tâche à la fois difficile et motivante [29]. En effet, un robot mobile autonome doit être en mesure de prendre des décisions, quant aux mouvements à réaliser, et ce, en exploitant les informations dont il dispose sur l'environnement et même, dans certain cas, des variations de l'environnement par rapport à une base de connaissance [31].

2.2. La logique floue en robotique mobile

Différentes méthodes ont été appliquées dans le but de résoudre le problème du contrôle du mouvement des robots mobiles. En 1991, Kanayama a proposé un contrôleur stable pour un véhicule non holonome en utilisant la fonction de Lyapunov. En 2001, Lee et Tai, dans le même souci, ont conçu un contrôleur basé sur la contrainte de saturation [20]. En outre, la majorité des structures proposées sont basées sur des approches intelligentes comme les réseaux de neurones artificiels [20] [29] (Fierro and Lewis 1998, Song and Sheen, 2000) et la logique floue (Bentalba 1997, Ishikawa 1991, Lee 1999, Pawlowski 2001) [20].

Dans la plupart des applications de la robotique mobile, si la dynamique du robot peut être décrite analytiquement, il est difficile -s'il n'est pas impossible- d'obtenir un modèle mathématique explicite de l'interaction du robot avec son environnement [28]. Le manque de précision et d'informations sur l'environnement limite l'applicabilité des structures de contrôle conventionnelles dans le domaine de la robotique mobile [28]. De plus, un robot mobile est contraint à plusieurs sources d'incertitudes imposées par l'environnement réel [27] [28]. Pour toutes ces raisons, l'utilisation d'un contrôleur flou *type-2* pour contrôler le mouvement d'un robot mobile s'avère très prometteur.

2.3. Modélisation

Dans la majorité des publications et des applications mentionnées auparavant, peu d'intérêt a été porté pour le contrôle dynamique des robots mobiles non holonomes où les couples et les forces sont les vraies entrées [20] [29]. Comme le but principal visé par cette application est la mise en commun des avantages des systèmes flous *type-2* et des algorithmes génétiques, nous nous limiterons à l'utilisation du modèle cinématique du robot mobile où les vitesses (linéaire et angulaire) seront considérées comme grandeurs de commande.

Nous représentons par la figure (4.1) un robot mobile en vue d'une modélisation cinématique. En effet, un robot mobile peut être considéré comme étant un véhicule à roues autonome et capable d'exécuter des missions dans des environnements connus où inconnus [29]. Ce type de robot mobile possède deux roues pivotantes montées sur un même axe (Y_m) et une roue libre (passive) au devant porté sur l'axe (X_m) [20] [29]. Le corps du robot est symétrique autour de l'axe perpendiculaire à l'axe des deux roues pivotantes (X_m). Dans ce qui suit, nous supposons que le mouvement de la roue d'équilibre du devant peut être ignoré dans la dynamique du robot, d'où le modèle cinématique suivant [20] [29]:

$$\dot{q} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_a \\ V_o \end{bmatrix} \quad (4.1)$$

où $q = [x, y, \theta]^T$ est le vecteur des coordonnées généralisées qui décrit la position du robot, (x, y) sont les coordonnées cartésiennes du centre de gravité du corps du robot, θ est l'angle entre les deux directions (X_m, X_r), $[V_a, V_o]^T$ est le vecteur des vitesses linéaire (d'avancement) et angulaire (d'orientation), respectivement. La contrainte de la non holonomie est exprimée par l'équation (4.2) qui signifie que le robot mobile ne peut se déplacer que dans la direction perpendiculaire à l'axe des deux roues pivotantes (X_m).

$$\dot{y} \cos \theta - \dot{x} \sin \theta = 0 \quad (4.2)$$

Le modèle (4.1) peut être réécrit comme suit :

$$\begin{cases} v_x = V_a \cdot \cos \theta \\ v_y = V_a \cdot \sin \theta \\ \dot{\theta} = V_o \end{cases} \quad (4.3)$$

où : v_x et v_y sont les projections de la vitesse linéaire sur l'axe (X_r) et (Y_r), respectivement.

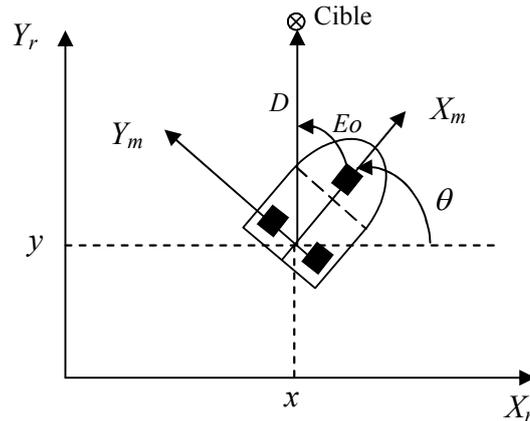


Fig.4.1. Représentation graphique d'un robot mobile à roues.

Dans cette application, la structure de commande est très similaire à celle utilisée dans les deux applications du chapitre précédent. Le contrôleur flou possède deux sorties (la vitesse d'avancement (V_a) et la vitesse d'orientation (V_o)) qui serviront comme des grandeurs de commande. Le rôle de l'algorithme génétique est l'optimisation des paramètres des ensembles flous des variables d'entrée/sortie des contrôleurs (voir figure 4.2).

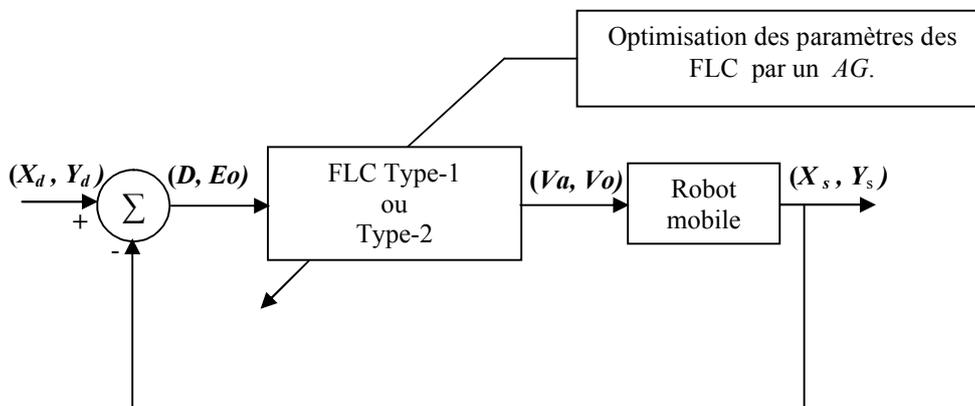


Fig 4.2. Structure du système de commande à base d'un AG.

3. Les algorithmes génétiques (AG)

Les *AG* sont des techniques d'optimisation qui s'inspirent fortement des mécanismes de l'évolution biologique liés aux principes de sélection et d'évolution naturelle [32]. Ils diffèrent des méthodes classiques d'optimisation essentiellement en quatre points fondamentaux [32] [34].

- ✓ Les *AG* utilisent un codage des éléments de l'espace de recherche et non pas les éléments eux-mêmes.
- ✓ Les *AG* recherchent une solution à partir d'une population de points et non pas à partir d'un seul point.
- ✓ Les *AG* n'imposent aucune régularité sur la fonction étudiée (continuité, dérivabilité, convexité...). C'est un des grands atouts des algorithmes génétiques.
- ✓ Les *AG* ne sont pas déterministes et utilisent des règles probabilistes de transition.

Le principe général du fonctionnement d'un algorithme génétique est représenté sur la figure (4.3) [34]. Un *AG* fera évoluer un ensemble de solutions, appelé population, aléatoirement initialisé, dans le but de guider efficacement la recherche vers de bonnes solutions dans l'espace de recherche. Pour passer d'une génération k à la génération $k+1$, les trois opérations suivantes sont appliquées successivement pendant deux phases : la phase de coopération qui fait appel aux opérateurs : de *sélection*, où les bons individus se reproduisent mieux que les mauvais, puis, l'opérateur de *croisement*, appliquée aux paires d'individus (les parents) d'une certaine proportion de la population pour en produire de nouveaux (les enfants). La phase d'adaptation individuelle qui fait appel à l'opérateur de *mutation* qui sera également appliqué à une certaine proportion de la population. Enfin, les nouveaux individus sont évalués et intégrés à la population de la génération suivante.

Plusieurs critères d'arrêts de l'algorithme sont possibles : le nombre de génération peut être fixé a priori ou bien l'algorithme peut être arrêté lorsque la population n'évolue plus suffisamment rapidement.

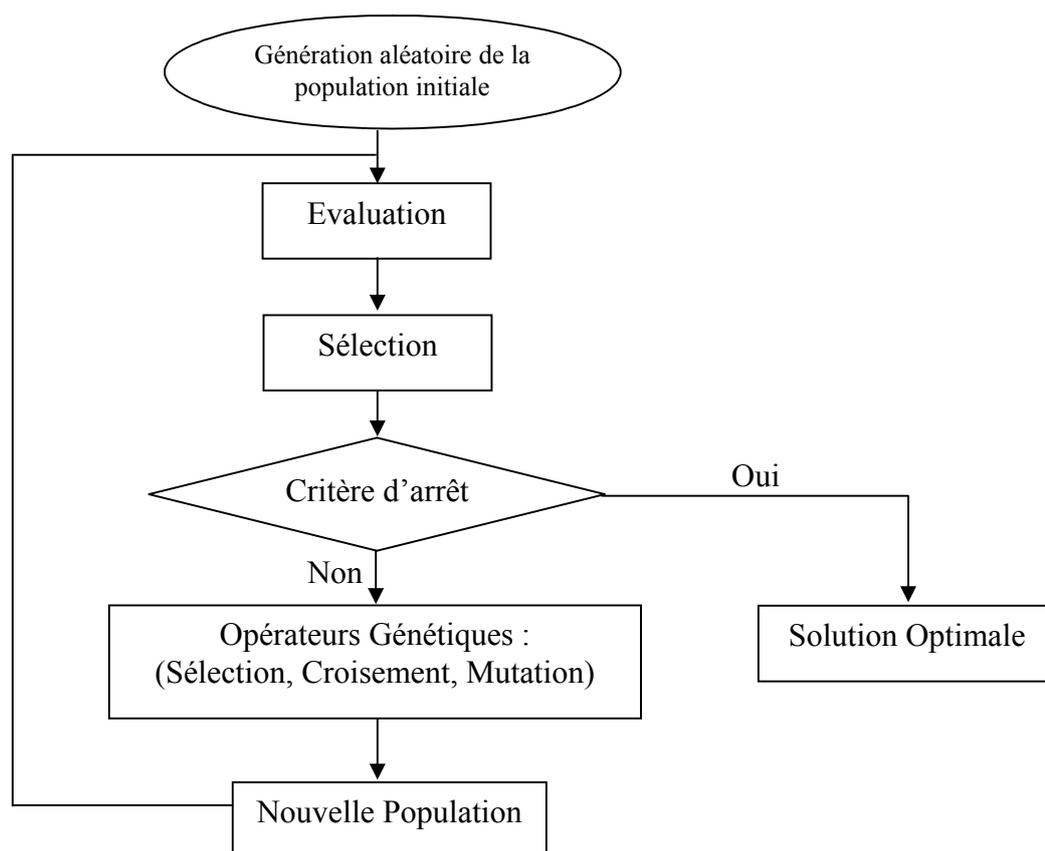


Fig 4.3. Organigramme d'un AG.

Pour utiliser un AG sur un problème d'optimisation, on doit disposer des éléments suivants :

- Un mécanisme de génération de la population initiale. Ce mécanisme doit être capable de produire une population d'individus non homogène qui servira de base pour les générations futures. Le choix de la population initiale est important. Car, il peut rendre plus ou moins rapide la convergence vers l'optimum global. Dans le cas où l'on ne connaît rien du problème à résoudre, il est important que la population initiale soit répartie sur tout le domaine de recherche.
- Un principe de codage de chromosomes de la population. Cette étape associe à chacun des points de l'espace d'état une structure de données. Elle se place généralement après une phase de modélisation mathématique du problème à traiter. La qualité du codage des données conditionne le succès des AG. Plusieurs codes sont utilisés. Mais, le codage binaire et le codage réel sont les plus courants.

Codage binaire : ce codage a été le premier à être utilisé dans le domaine des AG. Il présente plusieurs avantages : alphabet minimum, facilité de mise en place d'opérateurs

génétiques et existence de fondements théoriques. Néanmoins ce codage présente deux inconvénients : les performances de l'algorithme sont dégradées lorsque l'espace de recherche est multidimensionnel. Deux nombres voisins (exemple 7 et 8) peuvent être éloignés dans le codage binaire naturel (0111 et 1000) [32].

Codage réel : il a le mérite d'être simple. L'évaluation des chromosomes est plus rapide car le codage réel évite le transcodage du binaire vers les réels à chaque évaluation. Le codage réel est désormais largement utilisé, notamment dans les domaines applicatifs pour l'optimisation de problème à variables réelles. Cependant, il possède deux inconvénients majeurs : il a besoin d'opérateurs appropriés et manque de fondements théoriques [32].

- Un mécanisme d'évaluation des chromosomes. Un algorithme d'optimisation nécessite généralement la définition d'une fonction rendant compte de la pertinence des solutions potentielles, à partir des grandeurs à optimiser : C'est la *fonction d'adaptation f* (ou *fitness function*).
- Un mécanisme de sélection des chromosomes candidats à l'évolution. Comme son nom l'indique, la sélection permet d'identifier statistiquement les meilleurs individus de la population courante qui seront autorisés à se reproduire. Cette opération est fondée sur la performance des individus, estimée à l'aide de la fonction d'adaptation. Le paramètre principal de cette étape de sélection est ce que l'on appelle la pression sélective, qui correspond globalement au quotient de la probabilité de sélection du meilleur individu par la probabilité de sélection de l'individu moyen de la population courante. Ce paramètre gère la rapidité de concentration de la population autour de son meilleur individu.

La méthode la plus courante est la sélection proportionnelle. Ce mode de sélection consiste à associer à chaque individu un secteur de la roulette (de loterie) dont la surface est proportionnelle à sa performance (*fitness*) (figure 4.4) [34].

Le mécanisme de sélection proportionnelle a ses avantages mais aussi ses inconvénients. En particulier, si un individu est très supérieur à la moyenne, il constituera, presque exclusivement, la population suivante et on aura perdu toute diversité (risque de convergence prématurée). De même, s'il y a très peu de différence entre les qualités des différents individus, la recherche stagnera et se comportera plutôt comme une recherche aléatoire.

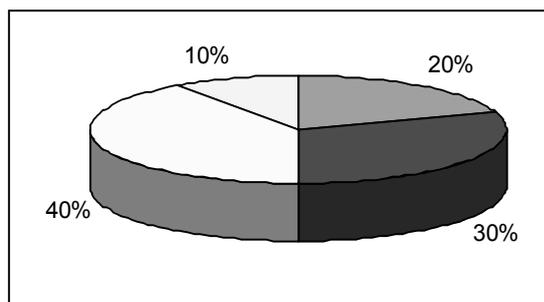


Fig. 4.4. La roue de loterie.

Des opérateurs permettant de diversifier la population au cours des générations et d'explorer l'espace d'état :

- Opérateur de *croisement* : le croisement a pour but d'enrichir la diversité de la population en manipulant la structure des chromosomes. Pour un problème discret, le croisement utilisé avec les chaînes de bits est le croisement à découpages de chromosomes (slicing crossover) qui se fait en trois étapes [34]. Au départ, deux solutions P_1 et P_2 appelées parents sont sélectionnées en fonction de leurs performances. Puis, on tire aléatoirement une position inter gènes dans chacun des parents. Enfin, on échange les deux sous-chaînes de chacun des chromosomes. Ce qui produit deux autres solutions C_1 et C_2 appelées enfants (figure 4.5.a). Ce mécanisme présente l'inconvénient de privilégier les extrémités des individus. Ce qui peut générer des fils plus ou moins proches de leurs parents. Pour éviter ce problème, on peut étendre ce principe en découplant le chromosome en trois sous-chaînes comme illustré sur la figure (4.5.b) ou encore en plusieurs sous chaînes [34].
- Opérateur de *mutation* : elle peut être définie comme étant l'inversion d'un bit (dans le cas du codage binaire) dans un chromosome (figure 4.6). Cela revient à modifier aléatoirement la valeur d'un paramètre du dispositif. Les mutations jouent le rôle de bruit et empêchent l'évolution de se figer. Elles permettent d'assurer une recherche aussi bien globale que locale, selon le poids et le nombre des bits mutés [34].

La mutation par bit et par génération se fait avec une probabilité P_m entre 0.001 et 0.01 [32]. On peut prendre également $P_m = 1/n_b$ où n_b est le nombre de bits composant un chromosome. Il est aussi possible d'associer une probabilité différente à chaque gène. Ce taux

est généralement faible puisqu'un taux élevé risque de conduire à une solution sous optimale. Pour éviter que les meilleurs individus soient altérés, on utilise une stratégie élitiste qui permet de reproduire à l'identique les meilleurs individus de la population. L'opérateur de reproduction ne joue alors que sur les individus restants.

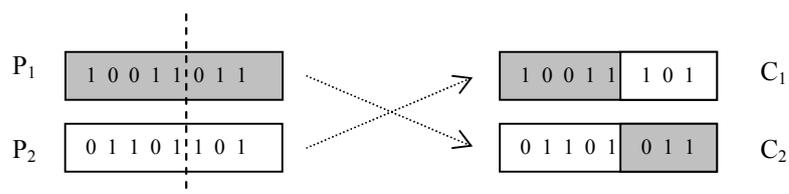


Fig. 4.5.a. Croisement en 1 point.

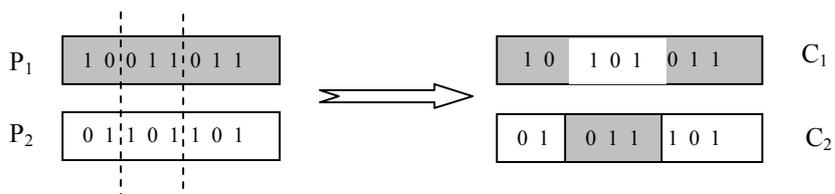


Fig. 4. 5. b. Croisement en 2 points.

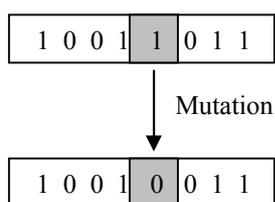


Fig.4. 6. Opérateur de mutation.

4. Navigation à base des FLC *type-1* et *type-2* : réglage heuristique

La structure de commande de la figure (4.2) a été simulée en utilisant le logiciel *Matlab* où le modèle du robot mobile est celui de (4.1).

Le tableau (4.1) montre la table de décision des deux contrôleurs flous, *type-1* et *type-2*, où la $l^{i\text{ème}}$ règle est exprimée comme suit :

$$R^{(l)} : \text{Si } D \text{ est } A_1^l \text{ et } Eo \text{ est } A_2^l \text{ alors } Va \text{ est } B_1^l \text{ et } Vo \text{ est } B_2^l , \text{ cas du } \textit{type-1} \quad (4.4)$$

$$R^{(l)} : \text{Si } D \text{ est } \tilde{A}_1^l \text{ et } Eo \text{ est } \tilde{A}_2^l \text{ alors } Va \text{ est } \tilde{B}_1^l \text{ et } Vo \text{ est } \tilde{B}_2^l , \text{ cas du } \textit{type-2} \quad (4.5)$$

(Va, Vo)

<i>D</i> \ <i>Eo</i>	N	Z	P
P	<i>N / NG</i>	<i>N / Z</i>	<i>N / PG</i>
M	<i>N / NG</i>	<i>G / Z</i>	<i>N / PG</i>
G	<i>M / NG</i>	<i>G / Z</i>	<i>M / PG</i>

Tab. 4.1. Table de décision du FLC *type-1* et *type-2*.

L'objectif recherché est d'assurer l'atteinte de la cible en empruntant la trajectoire la plus courte (optimale). Les performances du système sont testées en considérant des contrôleurs flous conçus d'une manière heuristique en absence et en présence d'une source d'incertitudes (tests 1et 2).

En absence du bruit, les deux grandeurs d'entrée des contrôleurs sont supposées sans incertitudes (nettes). Dans le cas contraire, la mesure distance robot-cible est supposée entachée d'un bruit b , d'où :

$$D_m = D_s + b \quad (4.6)$$

Pour chacune des deux entrées du contrôleur flou *type-1*, nous définissons trois fonctions d'appartenance de formes gaussiennes caractérisant les valeurs linguistiques (P : petit, M : moyenne, G : grande) pour la distance et (N : négative, Z : zéro, P : positive) pour l'erreur d'orientation, sur les univers de discours $[0, 500]$ et $[-\pi, \pi]$, respectivement. Les valeurs linguistiques de la première variable de sortie (vitesse d'avancement) (N : nulle, M : moyenne, G : grande) sont représentées par des fonctions singleton dont les centres sont (0, 10, 20). Pour la seconde variable de sortie (vitesse d'orientation), on a (NG : négative grande, N : négative, Zéro, P : positive, PG : positive grande) (figure 4.7).

Les fonctions d'appartenance des prémisses et des conséquences du contrôleur flou *type-2* sont représentées par la figure (4.8).

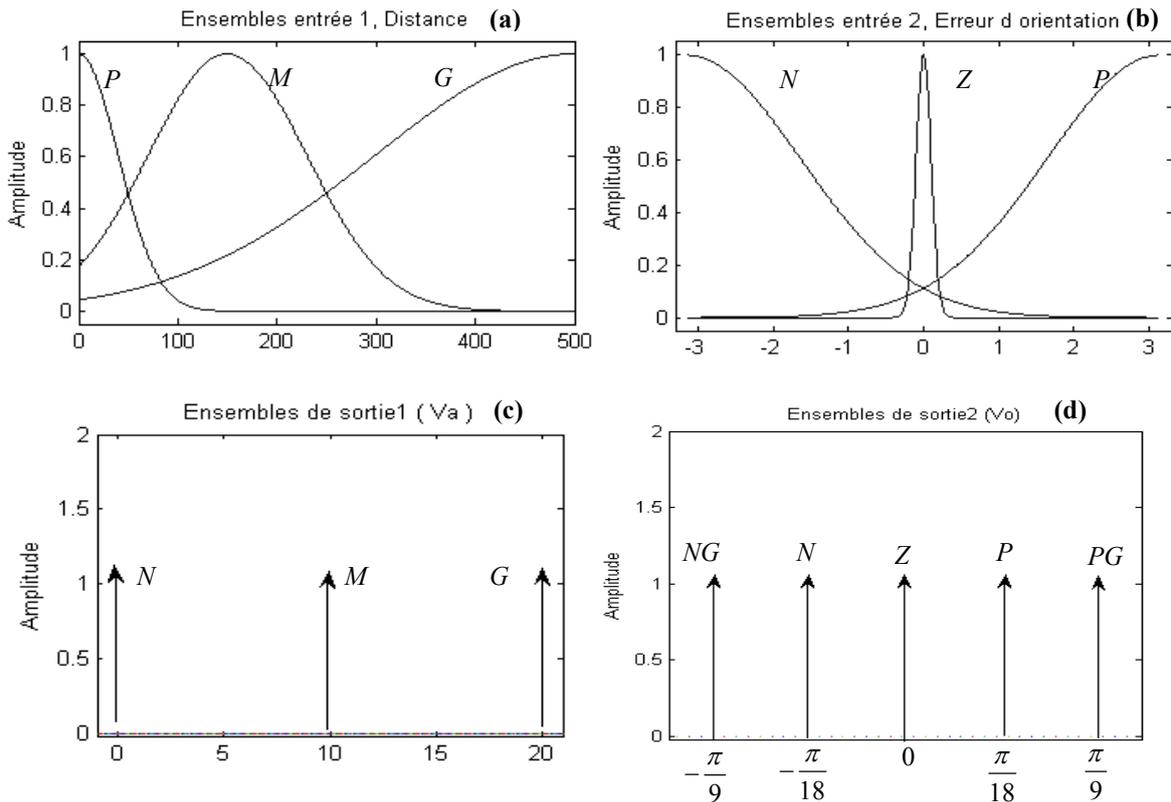


Fig. 4.7. Fonctions d'appartenance des variables d'entrée/sortie, (a) : Distance robot -cible, (b) : Erreur d'orientation, (c) : vitesse d'avancement, (d) : vitesse d'orientation du FLC *type-1*.

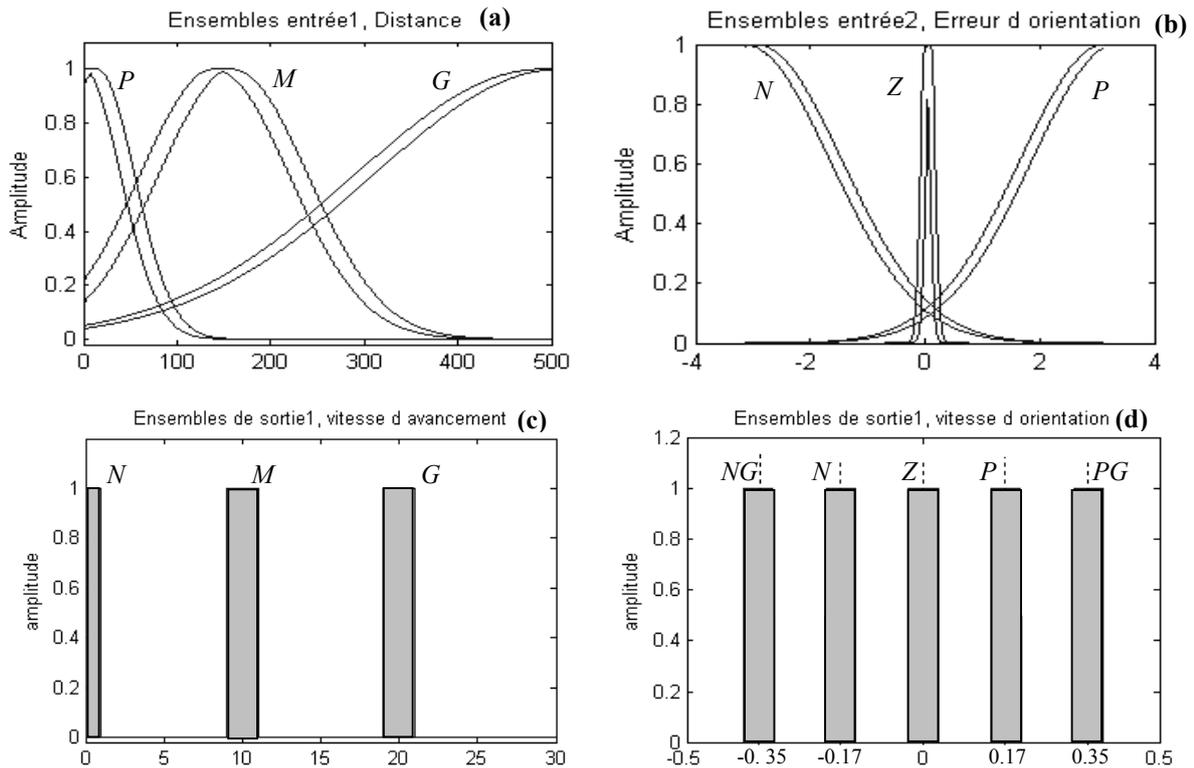


Fig. 4.8. Fonctions d'appartenance des variables d'entrée/sortie, (a) : Distance robot -cible, (b) : Erreur d'orientation, (c) : vitesse d'avancement, (d) : vitesse d'orientation du FLC type-2.

Test 1 : FLC type-1, en absence et en présence du bruit :

Le robot mobile est muni du contrôleur flou *type-1* dont les ensembles prémisses et conséquences sont représentées sur la figure (4.7). On constate, qu'à partir des conditions ($X_I=100, Y_I=100, E_o=-\pi$), le robot mobile a pu atteindre l'objectif (la cible dont les coordonnées $X_f=400, Y_f=100$), et ce, en absence ainsi qu'en présence du bruit (figure 4.9 a et b). L'évolution de la distance séparant le robot de la cible et la vitesse linéaire (d'avancement), dans les deux cas précédents, sont illustrées par la figure (4.10).

Test 2 : FLC type-2, en absence et en présence du bruit :

Le robot mobile est muni du contrôleur flou *type-2* dont les ensembles prémisses et conséquences sont représentées sur la figure (4.8). A partir des même conditions initiales ($X_I=100, Y_I=100, E_o=-\pi$), le robot mobile converge vers la cible ($X_f=400, Y_f=100$), et ce, en absence ainsi qu'en présence du bruit (figure 4.9 a et b). L'évolution de la distance séparant le robot de la cible et la vitesse linéaire (d'avancement), dans les deux cas

précédents, sont illustrés par la figure (4.10). En outre, l'évolution de l'erreur en fonction du bruit, pour les deux contrôleurs flous *type-1* et *type-2*, est représentée par la figure (4.11).

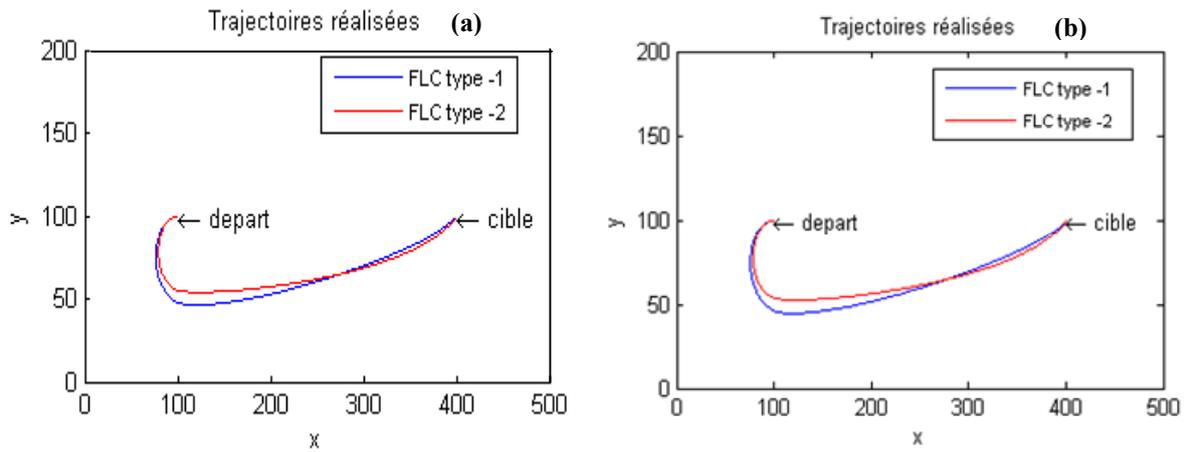


Fig. 4.9. Trajectoires réalisées par le robot mobile, (a) : en absence du bruit, (b) : en présence du bruit.

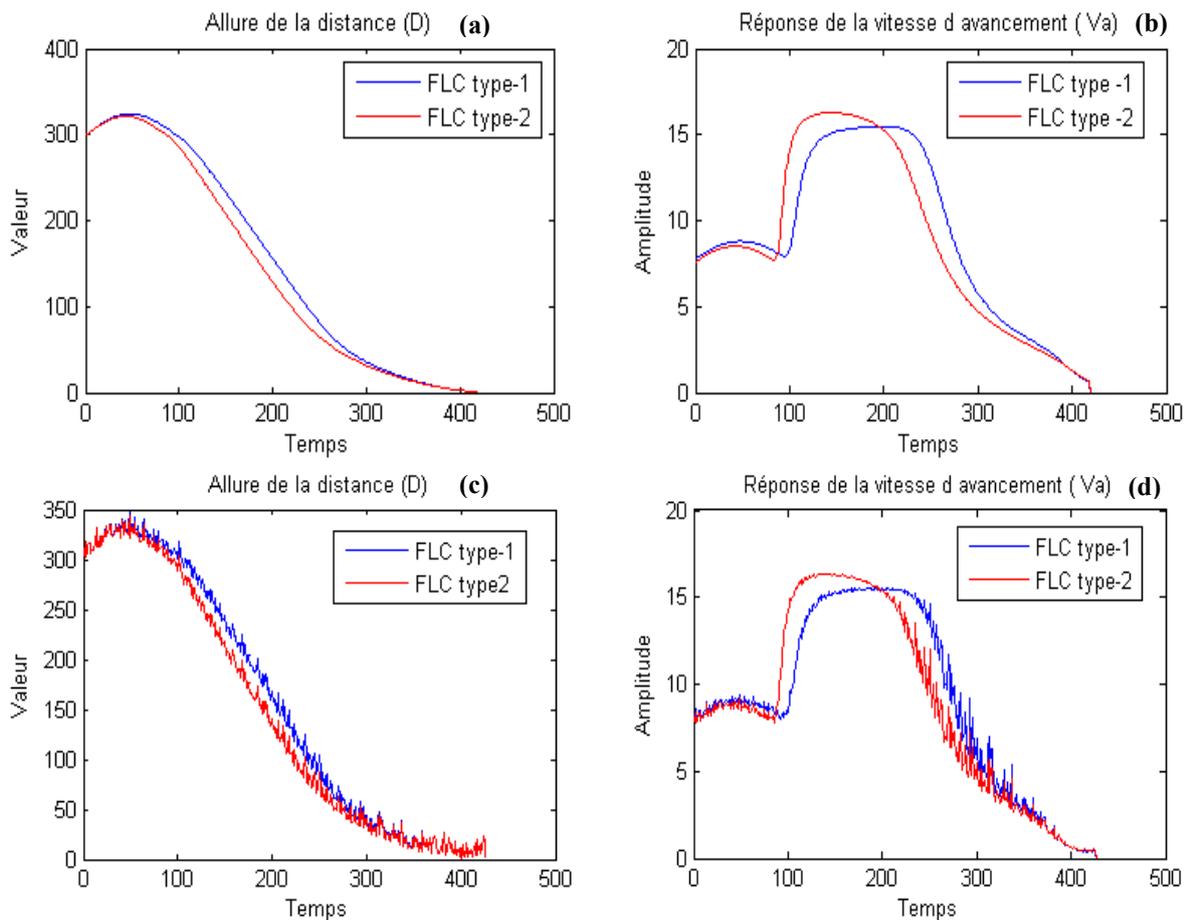


Fig.4.10. Résultats du FLC *type-1* et *type-2* : Distance robot-cible et Vitesse d'avancement (a) et (b) en absence du bruit et (c) et (d) en présence du bruit.

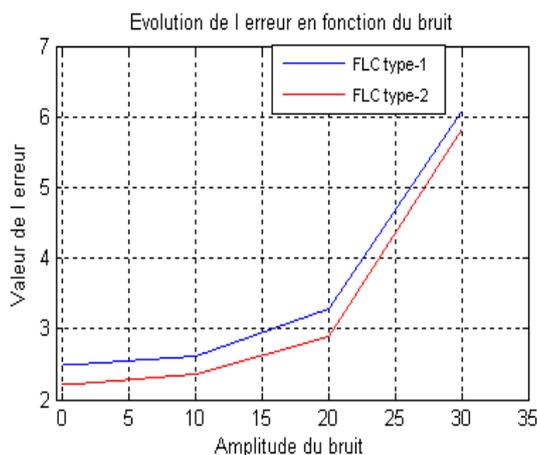


Fig.4.11. Evolution de l'erreur en fonction de l'écart-type du bruit gaussien.

- **Commentaires**

Des résultats obtenus dans les deux tests précédents, on peut dire que les contrôleurs flous *type-1* et *type-2* conçus d'une manière heuristique ont permis au robot mobile d'atteindre le but (la cible), aussi bien en absence qu'en présence de perturbation. Le robot s'oriente, d'abord, vers l'objectif puis avance. En absence ainsi qu'en présence du bruit (figures 4.9. et 4.11), les performances du contrôleur flou *type-2* sont toujours supérieures à celles du contrôleur flou *type-1*.

5. Optimisation des FLC par les AG

Dans la deuxième partie, nous avons appliqué un *AG* pour le réglage des différents paramètres (centres et largeurs) des ensembles prémisses et conséquences des deux contrôleurs flous *type-1* et *type-2* des figures (4.7 et 4.8). La symétrie de quelques ensembles des variables d'entrée/sortie des contrôleurs nous permet de réduire le nombre de paramètres (gènes) constituant les chromosomes.

Après une série de tests, nous avons opté pour les paramètres du tableau (4.2). Les paramètres des fonctions d'appartenance sont codés par des chaînes réelles. L'opérateur de mutation utilisé se résume comme suit : On génère deux nombre au hasard : le premier (p) vaut soit -1 soit +1 avec une probabilité égale. Le second (r), déterminant indirectement l'amplitude du changement, est un nombre produit aléatoirement dans l'intervalle $[0, 1]$ suivant une distribution uniforme. Le gène (g_k) muté donne le gène (g'_k) par l'expression suivante :

$$\begin{aligned}
 g'_k &= g_k - (g_k - g_{\min}) (1 - r) & \text{si } p = -1 \\
 g'_k &= g_k + (g_{\max} - g_k) (1 - r) & \text{si } p = +1
 \end{aligned}
 \tag{4.7}$$

Où g_{\min} et g_{\max} désignent respectivement les bornes inférieure et supérieure de la valeur du gène g_k .

Pour les FLC, la méthode d'inférence *max-produit* est utilisée pour l'inférence et la méthode des centres d'ensembles est retenue pour la défuzzification des sorties.

Paramètre	Valeur
Codage	Réel
Taille de population	100
Méthode de sélection	Roulette
Croisement	Plusieurs points
Mutation	Plusieurs points
Pc	0.65
Pm	0.01
Nombre de générations	60

Tab 4.2. Caractéristiques de l'algorithme génétique utilisé.

Notre objectif consiste à minimiser l'erreur dans le critère (*fonction objectif*) est donnée par la relation (4.7) :

$$E = t_N \cdot D_N \tag{4.8}$$

où t_N et D_N sont, respectivement, le temps et la distance en valeurs normalisées.

Nous définissons la fonction d'adaptation f par une simple transformation de la fonction objectif précédente comme suit :

$$f = \lambda \exp(-\sigma E) \tag{4.9}$$

Avec : $\lambda=10$ et $\sigma = 80$.

Pour le contrôleur flou *type-1*, nous avons utilisé treize paramètres pour chaque individu de la population. Pour cela, on désigne un chromosome de 13 gènes (g_1, \dots, g_{13}) de type réel où chaque gène représente un paramètre (centre ou largeur) d'un certain ensemble de la variable d'entrée ou de sortie du contrôleur flou (figure 4.12).

Pour le contrôleur flou *type-2*, le nombre de paramètres dans chaque individu de la population est de 21. La structure (chromosome), de type réel, se compose des gènes (g'_1, \dots, g'_{21}) où chaque gène désigne le centre ou la largeur d'un certain ensemble de la variable d'entrée ou de sortie du contrôleur flou (figure 4.13).

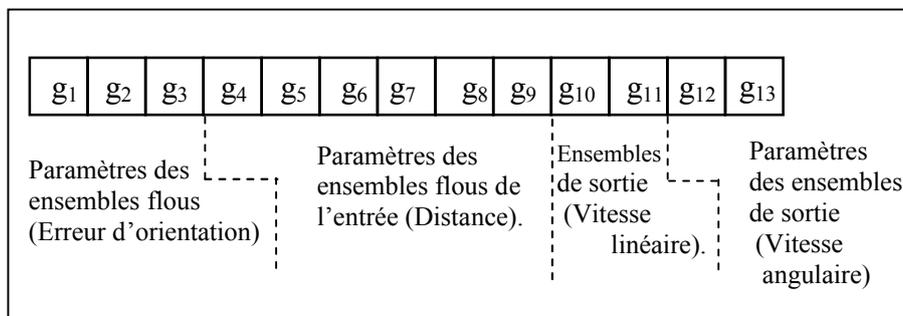


Fig 4.12. Représentation schématique du chromosome du FLC *type-1*.

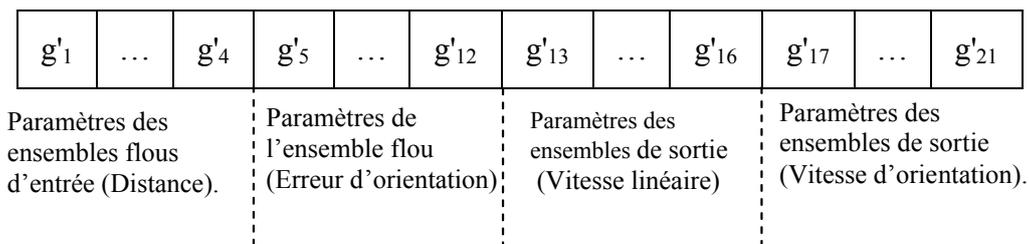


Fig 4.13. Représentation schématique du chromosome du FLC *type-2*.

Test 1 : Simulation en absence du bruit

La figure (4.14) représente l'évolution de la fonction d'adaptation et du critère d'erreur à travers les générations. La figure (4.15) montre les ensembles flous optimisés des variables d'entrée/sortie du contrôleur flou *type-1*. Ceux du *type-2* sont illustrés par la figure (4.16). La figure (4.17) montre les deux trajectoires réalisées par le robot mobile.

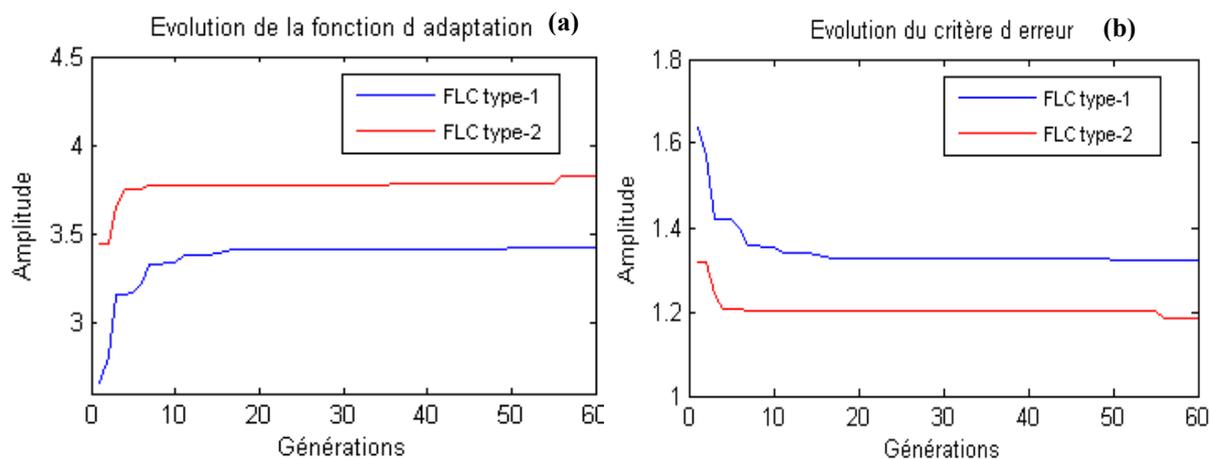


Fig 414. Les évolutions, (a) : de la fonction d'adaptation, (b) : du critère d'erreur à travers les générations.

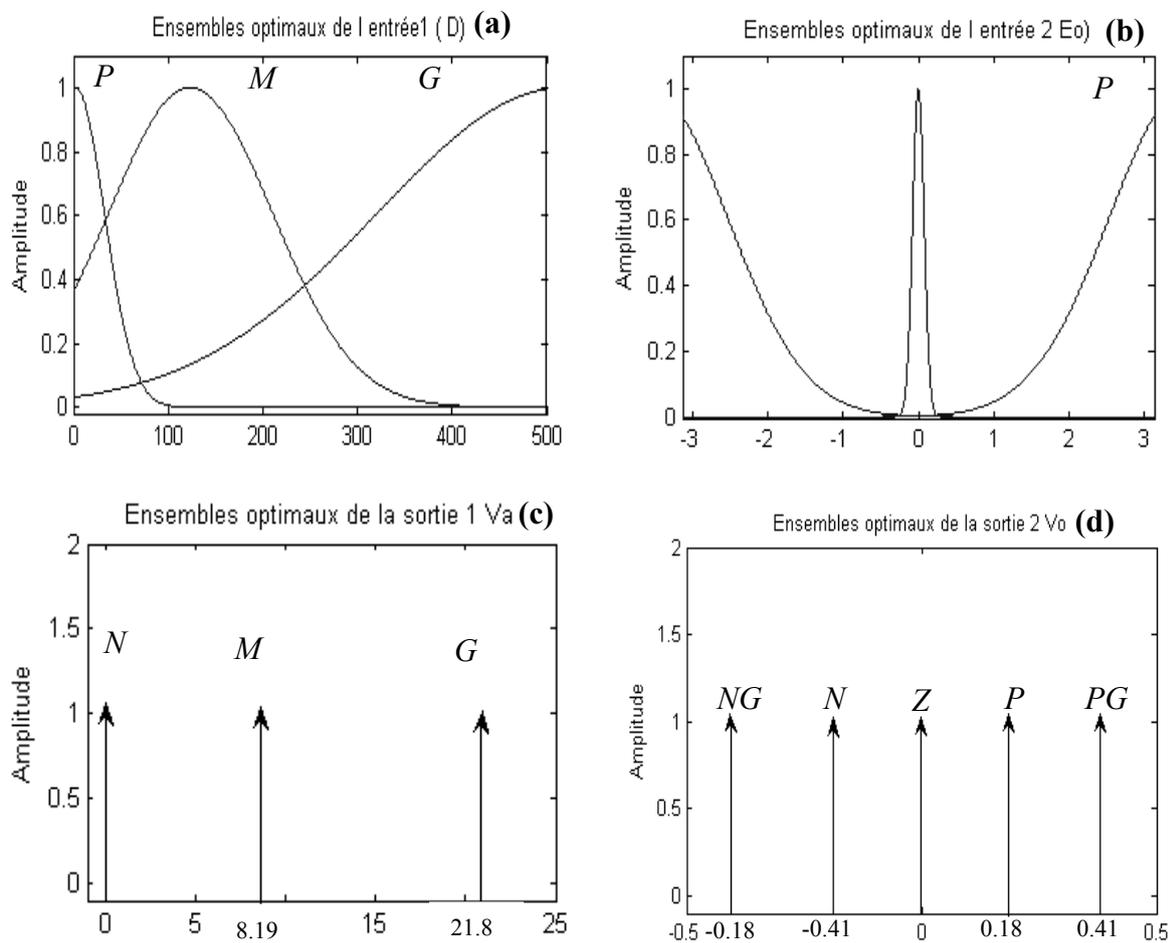


Fig. 4.15. Fonctions d'appartenance optimisées des variables d'entrée/sortie, (a) : Distance robot -cible, (b) : Erreur d'orientation, (c) : vitesse d'avancement, (d) : vitesse d'orientation du FLC type-1.

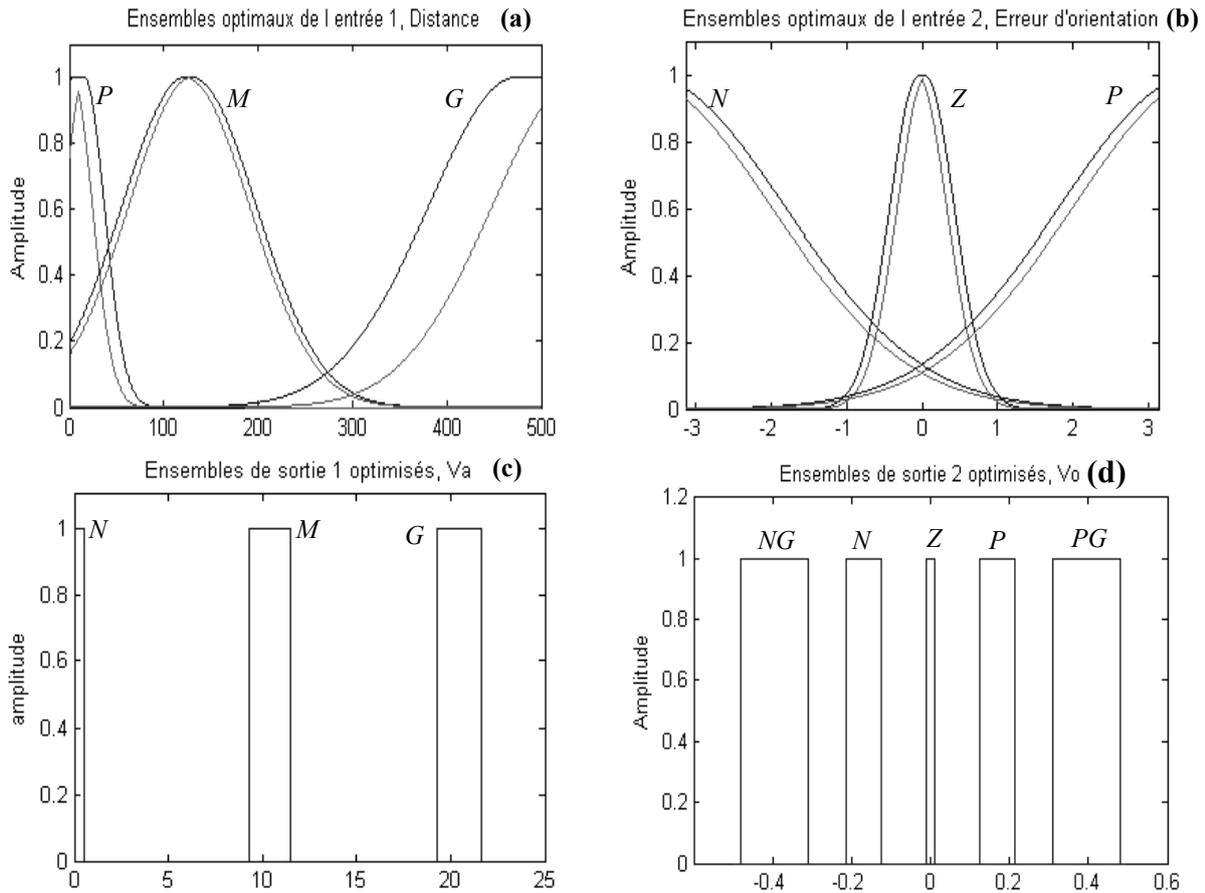


Fig. 4.16. Fonctions d'appartenance optimisées des variables d'entrée/sortie, (a) : Distance robot-cible, (b) : Erreur d'orientation, (c) : vitesse d'avancement, (d) : vitesse d'orientation du FLC type-2.

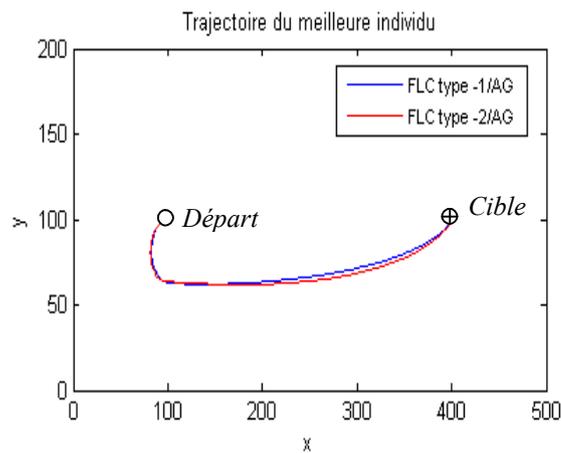


Fig 4. 17. Trajectoires optimales réalisées par le robot.

Test 2 : Simulation en présence du bruit :

Considérons maintenant deux types de bruit ajoutés à l'entrée des contrôleurs (distance robot-cible). Le premier est un bruit aléatoire de distribution normale ($b(i)=k*randn$). Le deuxième est un bruit sinusoïdale ($b(i)=A\cos(i)$).

Les deux contrôleurs flous *type-1* et *type-2* sont testés pour différentes valeurs de k ($k=0,\dots,40$). L'erreur (en valeur normalisée) est quantifiée puis comparée à celle obtenue par un réglage heuristique des paramètres (voir tableau 4.3). L'évolution de l'erreur est donnée par la figure (4.18).

		Valeurs de k					
		0	1	10	20	30	40
Type du contrôleur	<i>FLC type-1</i>	2.5123	2.5591	6.2714	7.2709	8.9753	11.6174
	<i>FLC type-2</i>	2.2518	2.3110	2.9174	7.0532	8.7806	10.1456
	<i>Type-1/AG</i>	1.3731	1.3803	1.4375	1.4999	1.5678	1.6360
	<i>Type-2/AG</i>	1.1895	1.1942	1.2279	1.2651	1.3243	1.3854

Tab 4.3. Valeurs de l'erreur pour différentes valeurs de l'écart-type k .

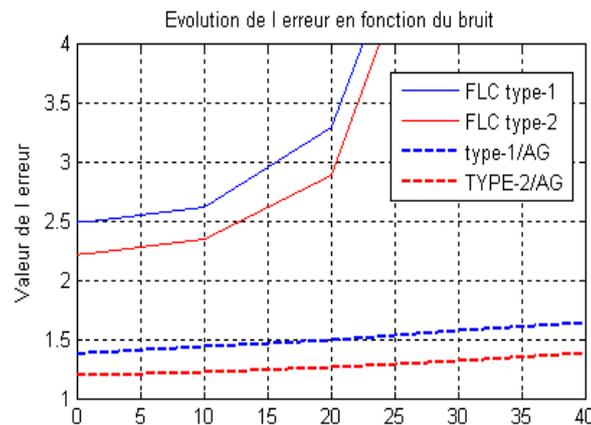


Fig 4.18. Evolution de l'erreur en fonction de l'écart-type du bruit gaussien.

Dans le second cas, le bruit est sinusoïdal: $b(i) = A \cos(i)$. Pour différentes valeurs de A ($A=10, \dots, 100$), les valeurs de l'erreur sont reportées sur le tableau (4.4). La figure (4.19) montre l'évolution de la distance robot-cible ainsi que celle de la vitesse d'avancement, respectivement. La trajectoire réalisée par le meilleur individu est représentée par la figure (4.20.a). L'évolution des erreurs est donnée par (4.20.b).

		Amplitude du bruit					
		10	20	30	40	60	100
Type du contrôleur	<i>FLC type-1</i>	2.5123	2.5591	6.2714	7.2709	8.9753	11.6174
	<i>FLC type-2</i>	2.2518	2.3110	2.9174	7.0532	8.7806	10.1456
	<i>Type-1/AG</i>	1.3929	1.4181	1.4443	1.4695	1.5146	1.5849
	<i>Type-2/AG</i>	1.2225	1.2559	1.2909	1.3238	1.3824	1.4752

Tab 4.4. Valeurs de l'erreur pour différentes amplitudes du bruit sinusoïdal.

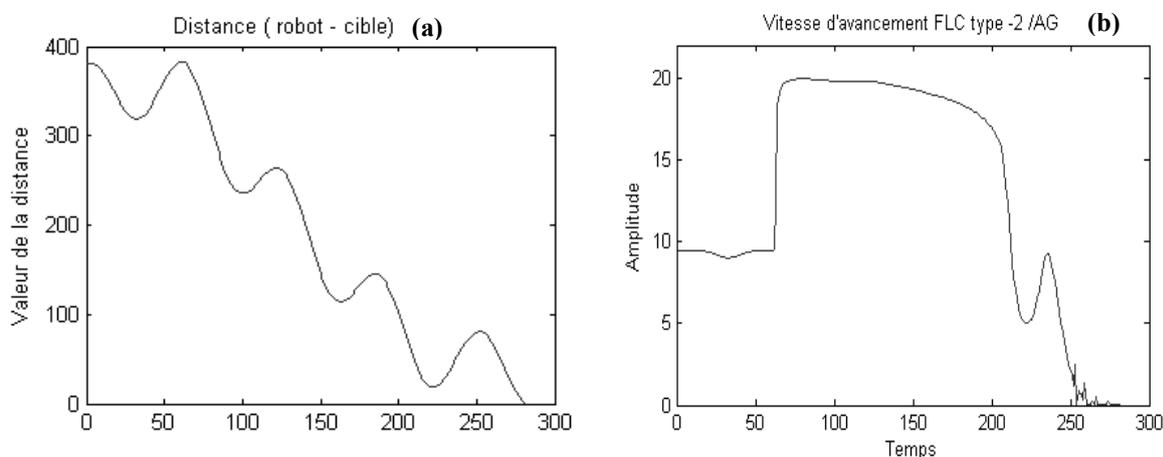


Fig 4.19. Evolution, (a) : de la distance robot-cible, (b) : vitesse d'avancement pour FLC type -2 en présence d'un bruit sinusoïdal.

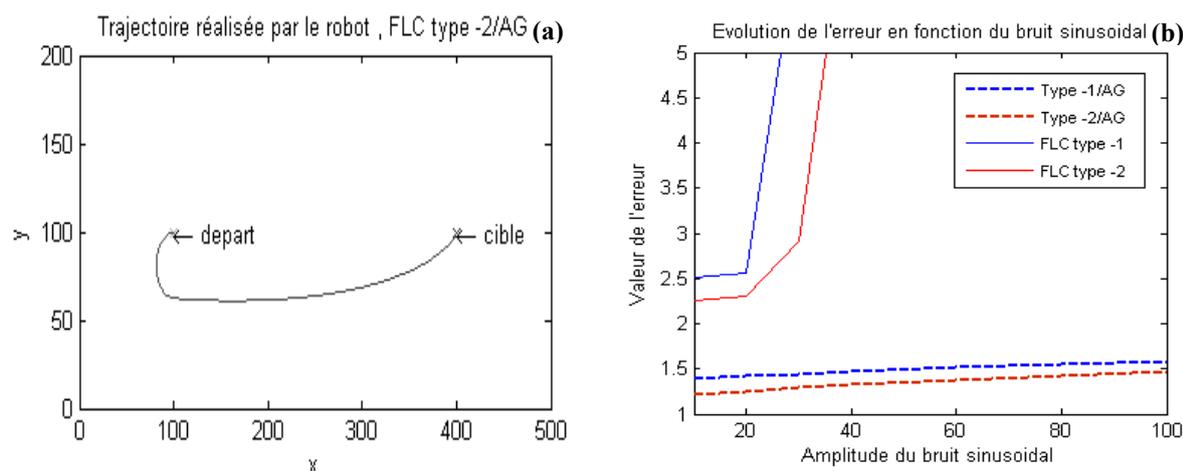


Fig 4.20. (a) Trajectoire du robot pour le FLC type-2 /AG, en présence d'un bruit sinusoïdal.

(b) Evolution de l'erreur en fonction de l'amplitude du bruit sinusoïdal.

- **Commentaires**

Des résultats obtenus, on constate que l'erreur est minimale avec les contrôleurs flous dont les paramètres sont optimisés par les AG.

En absence ou en présence de bruit, les performances du contrôleur flou *type-2* dont les paramètres sont optimisés par un *AG* sont meilleures par rapport à :

- celles du contrôleur flou *type-2* dont les paramètres sont réglés d'une manière heuristique ;
- celles du contrôleur hybride à base de la logique floue *type-1* optimisé par AG.

6. Conclusion

Dans ce chapitre, nous avons appliqué une structure de commande floue *type-2* à un modèle cinématique d'un robot mobile à roues non holonome. Dans un premier temps, les paramètres du contrôleur flou (centres et largeurs des fonctions d'appartenance des prémisses et des conséquences) sont réglés d'une manière heuristique. Les résultats obtenus, en absence et présence de perturbations sont comparés à ceux obtenus avec un contrôleur *type-1*. Dans la seconde partie de l'application, les paramètres des deux contrôleurs sont optimisés par un *AG*. Nous avons testé les deux contrôleurs en présence de deux sources d'incertitudes (bruit de distribution normale et un bruit sinusoïdal). Pour les deux types de bruit, on a constaté la supériorité des contrôleurs *type-2* par rapport à ceux du *type-1*.

Conclusion générale

Conclusion générale.

Le travail présenté dans ce mémoire a porté principalement sur l'étude des systèmes flous *type-2* et leur application à la commande des systèmes non linéaires. Les contrôleurs flous *type-2*, vue leur structure intégrant des fonctions avec une dimension supplémentaire leur permettant une meilleure prise en compte de ces incertitudes, donnent des performances supérieures par rapport à ceux du *type-1* (classiques). Mais, cela, au prix d'un degré plus élevé de complexité qui fait, même, la raison de leur timide utilisation. Ainsi pour confirmer l'avantage des systèmes flous *type-2*, nous avons fait état, dans le troisième chapitre, de deux applications. Des résultats obtenus, on constate, quoique minime, dans certain cas, la supériorité des contrôleurs flous *type-2* par rapport à ceux du *type-1*.

En dépit des avantages des contrôleurs flous par rapport aux autres techniques de contrôle, leur mise en œuvre présente un inconvénient majeur du moment où la présence d'un expert humain est nécessaire pour un bon réglage des paramètres (forme et nombre de fonctions d'appartenance, facteurs d'échelle, règles floues.. .). Ainsi, pour remédier à cet inconvénient, nous avons fait appel, dans le quatrième chapitre, aux algorithmes génétiques. Les résultats obtenus sur un modèle cinématique d'un robot mobile ont montré l'apport de cette approche par rapport au réglage empirique. La robustesse de ces contrôleurs hybrides obtenus vis-à-vis de l'une des contraintes qui caractérisent un environnement réel (bruit) a été aussi testée.

Enfin, on peut dire que ce travail nous a permis de prévoir de nouvelles extensions futures, telles que :

- ✓ Généralisation du type de la logique floue *type-2* intervalle vers le *type-2* gaussien.
- ✓ L'utilisation d'autres méthodes d'optimisation globale telle que le recuit simulé pour une éventuelle étude comparative.
- ✓ Application de ces structures à des systèmes non linéaires plus complexes (modèle dynamique d'un robot mobile).
- ✓ Implémentations pratiques.

Bibliographies

- [1] A. Kaufman, ‘ Introduction à la théorie des sous-ensembles flous’, Tome1, Masson, 1973.
- [2] L. Gacogne, ‘Elément de logique floue,’ Hermes, 1997.
- [3] O. GUENOUNOU, ‘Optimisation des contrôleurs flous par les algorithmes génétiques hiérarchisés. Application sur un actionneur asynchrone’, Mémoire de Magister en automatique et traitement et traitement du signal, Université A. MIRA de Bejaia, 2003.
- [4] J. M. Mendel, ‘Type-2 fuzzy sets made simple,’ IEEE transactions on fuzzy systems, vol. 10, no. 2, avril 2002.
- [5] N. N. Karnik, J. M. Mendel, ‘An introduction to type-2 fuzzy logic systems’, USC, Report, <http://sipi.usc.edu/~mendel/report>, 1998.
- [6] N. N. Karnik, J. M. Mendel, ‘ Type-2 fuzzy logic systems,’ IEEE transactions on fuzzy systems, vol. 7, no. 6, 1999.
- [7] L. A. Zadeh, ‘The concept of a linguistic variable and its application to approximate reasoning-1’, Information sciences, vol. 8, pp. 199 – 249, 1975.
- [8] J. M. Mendel, ‘Fuzzy logic systems for engineering: A tutorial,’ IEEE proceedings, vol. 83, no. 3, pp. 345-377, 1995.
- [9] J. M. Mendel, ‘Computing derivatives in interval type-2 fuzzy logic systems,’ IEEE transactions on fuzzy systems, vol. 12, no. 1, 2004.
- [10] Q. Liang, N. N. Karnik, J. M. Mendel, ‘ Connection admission control in ATM networks using survey-based type-2 fuzzy logic systems,’ IEEE transactions on systems, Man, Cybernetics, Part C, vol. 30, no. 3, Aug.2000.
- [11] Q. Liang, J. M. Mendel, ‘Interval type-2 fuzzy logic systems: theory and design’, IEEE transactions on fuzzy, vol. 8, no. 5, 2000.
- [12] N. N. Karnik, J. M. Mendel, ‘Type-2 fuzzy systems: Type reduction’, In SMC’98 Conference Proceedings. 1998 IEEE International conference on systems, Man, and Cybernetics, Part 2, pp. 2046-2051, 1998.

-
- [13] J. M. Mendel, 'On the importance of interval sets in type-2 fuzzy logic systems', 20th NAFIPS international conference, Vancouver, BC, Canada, 2001.
- [14] L. A. Zadeh, 'Fuzzy sets,' Information and control, vol. 8, pp. 338-353, 1965.
- [15] C. KHEIREDDINE, 'Structures d'identification et de commande des systèmes non linéaires basées sur les techniques floues, thèse de doctorat de l'université de Batna, 2006.
- [16] B. Bouchon-Meunier, C. Marsala, 'Logique floue, principe, aide à la décision', Lavoisier, 2003.
- [17] G. J. Klir, B. Yuan, 'Fuzzy sets and fuzzy logic: theory and applications,' Prentice Hall, 1995.
- [18] J. R. JANG, C. T. Sun, 'Neuro-Fuzzy modeling and control', Proceedings of the IEEE, vol. 83, no. 3, 1995.
- [19] L. Ruthowski, 'Computational intelligence: methods and techniques,' Springer, 2008.
- [20] O. Castillo, P. Melin, 'Type-2 fuzzy logic: theory and applications,' Springer, 2008.
- [21] K. M. Passino, S. Yurkovich, 'Fuzzy control', Addison-Wesley, 1998.
- [22] L. Reznik, 'Fuzzy controllers', Biddles Ltd, 1997.
- [23] R. Supelveda, O. Castio, P. Melin, A. Rodriguez-Diaz, O. Montiel, 'Handling Uncertainty in Controllers using Type-2 Fuzzy Logic,' presented at the 2005 IEEE International Conference on Fuzzy Systems.
- [24] R. Supelveda, O. Castio, P. Melin, A. Rodriguez-Diaz, O. Montiel, 'Expérimental study of intelligent controllers under uncertainty using *type-1* and *type-2* fuzzy logic', inf. Sciences, 177(10): 2023-2048, 2007.
- [25] M. Ider, B. Soualmi, B. Mendil, 'Incorporation des incertitudes dans les contrôleurs en utilisant la logique floue type-2', 3rd international conference on electrical and engineering, Algiers, Algeria, May 19-21, 2009.
- [26] O. Castio, N. Cazarez, P. Melin, 'Design of stable type-2 fuzzy logic controllers based on a fuzzy Lyapunov approach', 2005 IEEE International Conference on Fuzzy Systems, Vancouver, BC, Canada.
- [27] H. A. Hagrass, 'Hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots', IEEE Trans on Fuzzy Systems, Vol. 12, No. 4, pp. 524-539, 2004.
- [28] S. Nurmaini, S. Z. Hashim, 'An embedded fuzzy type-2 controller based sensor behaviour for mobile robot', Eighth international conference on intelligent systems design and applications, 2008.

- [29] R. Martinez, O. Castillo, L. T. Aguilar, 'Optimisation with genetic algorithms of interval type-2 fuzzy logic controllers for an autonomous wheeled mobile robot: a comparison under different kinds of perturbations', IEEE international conference on fuzzy systems, 2008.
- [30] F. Fahimi, 'Autonomous robots, modelling, path planning, and control', Springer, 2009.
- [31] F. LACHEKHAB, 'Planification et navigation à base de la logique floue d'un nrobot mobile dans un environnement partiellement connu', Mémoire de Magister en RAI, Ecole Militaire Polytechnique, 2003.
- [32] D. Goldberg, 'Algorithme génétique : exploration, optimisation et apprentissage automatique', Addition Wesley, 1989.
- [33] X. Ye, L. Fu, Y. Zhang, ' Type-2 fuzzy system and level set,' IEEE Third international conference on semantics, Knowledge and grid, 2007.
- [34] M. Gen, R. Cheng, ' Genetic algorithms & engineering optimisation', Wesley, 2000.
- [35] N. R. Cazarez-Castro, L. T. Aguilar, 'Hybrid genetic optimisation of a type-2 fuzzy logic controller', 8th international conference on hybrid intelligent systems, 2008.
- [36] M. Y. Hsiano, T. Hseng, S. Li, ' Design of Interval type-2 fuzzy logic controller', 2006 IEEE International Conference on Systems, Man, and Cybernetics, Taipei, Taiwan.
- [37] D. Wang, L. Acar, 'An analysis of type-1 and type-2 fuzzy logic systems', Internationnal symposium on intelligent control /intelligent systems and Semiotics, Combridge, MA, 1999.
- [38] H. Hagra, 'Type-2 fuzzy logic controller for autonomous mobile robot', Budapest, 2004.
- [39] S. N. Sivanandam, S. N. Deepa, 'Introduction to génétic algorithms', Springer-Verlag, 2008.

Annexe

1. Caractéristiques d'un ensemble flou *type-1*

Un sous-ensemble flou est complètement défini par la donnée de sa fonction d'appartenance. A partir d'une telle fonction, un certain nombre de caractéristiques du sous-ensemble flou peuvent être étudiées. Les définitions utilisées ici sont celles données dans [7] [16] [17][22].

- *Noyau* : le noyau d'un sous-ensemble flou A de X , noté $Noy(A)$, est l'ensemble de tous les éléments qui lui appartiennent totalement. Formellement :

$$Noy(A) = \{x \in X \mid \mu_A(x) = 1\} \quad (A.1)$$

- *Support* : le support d'un sous ensemble flou A de X , $Supp(A)$, est l'ensemble de tous les éléments qui lui appartiennent totalement. Formellement :

$$Supp(A) = \{x \in X \mid \mu_A(x) > 0\} \quad (A.2)$$

- *Hauteur* : la hauteur d'un sous ensemble flou A de X , notée $h(A)$, est la valeur maximale atteinte sur le support de A . Formellement :

$$h(A) = \sup_{x \in X} \mu_A(x) \quad (A.3)$$

On dira alors qu'un sous-ensemble flou est normalisé si sa hauteur $h(A)$ est égale à l'unité.

- *Cardinalité* : la cardinalité d'un sous-ensemble flou discret A de X , notée $|A|$, est définie par la somme des degrés d'appartenance, Formellement :

$$|A| = \sum_{i=1}^n \mu_A(x) \quad (A.4)$$

- α – coupe : α – coupe A_α d'un ensemble flou A est un sous ensemble ordinaire de l'univers de discours X dont tout ses éléments possèdent un degré d'appartenance supérieur ou égal à α :

$$A_\alpha = \{x \in X \mid \mu_A(x) \geq \alpha\} \quad \alpha \in [0 \quad 1] \quad (\text{A.5})$$

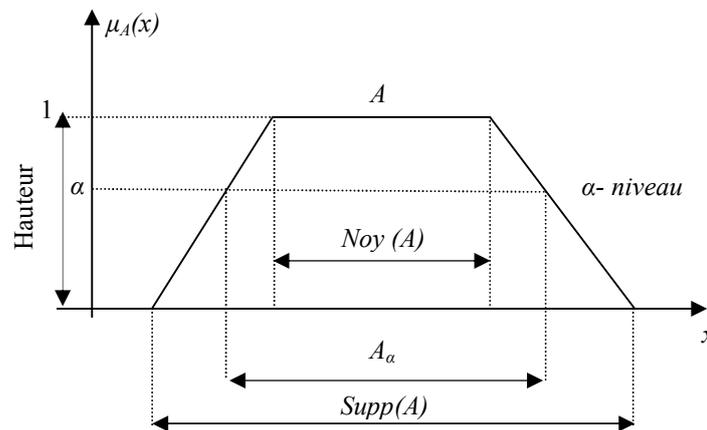


Fig.A.1. Caractéristiques d'un sous ensemble flou.

2. Principe d'extension de Zadeh [7]

Un ensemble flou unidimensionnel A sur un univers de discours X est défini par sa fonction d'appartenance $\mu_A(x)$, où x est un élément de X , c-à-d

$$A = \int_X \mu_A(x) \mid x \quad \text{cas continu} \quad (\text{A.6})$$

$$A = \sum_i \mu_A(x) \mid x_i \quad \text{cas discret} \quad (\text{A.7})$$

Où $\mu_A(x)$ désigne le degré d'appartenance d'un élément x à l'ensemble flou A .

Si A_1, A_2, \dots, A_n sont des ensembles flous sur X_1, X_2, \dots, X_n respectivement, le produit cartésien de A_1, A_2, \dots, A_n noté par $A_1 \times A_2 \times \dots \times A_n$ est défini comme un sous ensemble flou possédant la fonction d'appartenance multidimensionnelle suivante :

$$\mu_{A_1 \times \dots \times A_n}(x_1, \dots, x_n) = \mu_{A_1}(x_1) \wedge \dots \wedge \mu_{A_n}(x_n) \quad (\text{A.8})$$

Ce qui est équivalent à :

$$A_1 \times \dots \times A_n = \int_{X_1} \dots \int_{X_n} \mu_{A_1}(x_1) \wedge \dots \wedge \mu_{A_n}(x_n) \mid (x_1, \dots, x_n) \quad (\text{A.9})$$

Le terme $\mu_{A_1}(x_1) \wedge \dots \wedge \mu_{A_n}(x_n)$ représente une fonction d'appartenance multidimensionnelle représentant une relation \mathfrak{R} .

2.1. Principe d'extension monodimensionnel

Le principe d'extension des ensembles flous est essentiellement une identité de base qui permet à un domaine de définition d'une relation à être étendu de quelques points de X à des sous ensembles flous de X . Particulièrement, supposons que f est une relation de X vers Y et A un ensemble flou défini par (1.16) si A continu et par (1.17) si A est discret. Alors le principe d'extension affirme que l'image de A sous la relation f est un ensemble flou, $f(A)$ défini par :

$$f(A) = f \left[\int_X \mu_A(x) \mid x \right] = \int_Y \mu_A(x) \mid f(x) \quad \text{cas continu} \quad (\text{A.10})$$

$$f(A) = f \left[\sum_i \mu_A(x_i) \mid x_i \right] = \sum_i \mu_A(x_i) \mid f(x_i) \quad \text{cas discret} \quad (\text{A.11})$$

Par conséquent, l'image de A sous f peut être déduite en connaissant les images de u_i sous f .

- **Exemple 1**

Soit $U = 1+2+\dots+10$, et soit f l'opération « carré ». Soit *petit* un sous ensemble flou sur X défini par :

$$\textit{petit} = 1/1 + 1/2 + 0.8/3 + 0.6/4 + 0.4/5 \quad (\text{A.12})$$

par conséquent :

$$\textit{petit}^2 = f(\textit{petit}) = 1/1 + 1/4 + 0.8/9 + 0.6/16 + 0.4/25 \quad (\text{A.13})$$

2.2. Principe d'extension multidimensionnel

Soit A un ensemble flou multidimensionnel sur $X_1 \times X_2 \times \dots \times X_n$ défini par sa fonction d'appartenance multidimensionnelle $\mu_{A_1 \times \dots \times A_n}(x_1, \dots, x_n)$, et soit f une relation sur produit cartésien $X_1 \times X_2 \times \dots \times X_n$ vers l'espace Y .

L'application directe du principe d'extension nous donne :

$$\begin{aligned}
 f(A) &= f \left[\int_{X_1} \dots \int_{X_n} \mu_A(x_1, \dots, x_n) / (x_1, \dots, x_n) \right] \\
 &= \int_{X_1} \dots \int_{X_n} \mu_A(x_1, \dots, x_n) / f(x_1, \dots, x_n) \\
 &= \int_{X_1 \times \dots \times X_n} \mu_A(x_1, \dots, x_n) / f(x_1, \dots, x_n) \quad (\text{A.14}) \\
 &= \int_{X_1, \dots, X_n} \mu_{A_1}(x_1) \wedge \dots \wedge \mu_{A_n}(x_n) / f(x_1, \dots, x_n) \\
 &= \int_Y \mu_{A_1}(x_1) \wedge \dots \wedge \mu_{A_n}(x_n) / f(x_1, \dots, x_n)
 \end{aligned}$$

Les μ_{A_i} sont les projections de A sur les X_i . Parce que, généralement, on ne connaît pas A mais ces projections A_1, A_2, \dots, A_n .

- **Exemple 2**

Soit $*$ une relation du produit cartésien $X \times V$ vers W Donc nous avons :

$$w = x * v \quad x \in X, v \in V \text{ et } w \in W \quad (\text{A.15})$$

Ou w est l'image de x et de v sous $*$.

Soit A et B deux ensembles flous sur X et V respectivement :

$$A = \int_X \mu_A(x) | x \quad \text{où} \quad A = \sum_i \mu_A(x_i) | x_i \quad (\text{A.16})$$

$$B = \int_V \mu_B(v) | v \quad \text{où} \quad B = \sum_i \mu_B(v_i) | v_i \quad (\text{A.17})$$

Alors l'image de $A \times B$ sous $*$ est un sous ensemble flou $A * B$ sur W défini par :

$$f(A, B) = \int_X \int_V \mu_A(x) \wedge \mu_B(v) | (x * v) \quad (\text{A.18})$$

où :

$$f(A, B) = A * B = \sum_i \sum_j \mu_A(x_i) \wedge \mu_B(v_j) | (x_i * v_j) \quad i=1, \dots, m \quad j=1, \dots, n \quad (\text{A.19})$$

3. Terminologie de la logique floue *type-2*

Les définitions de 1 à 8 sont celles données dans [4] [9] [19] [20] [32]

Définition 1

Un ensemble flou *type-2* noté \tilde{A} est caractérisé par une fonction d'appartenance tridimensionnelle, $\mu_{\tilde{A}}(x, u)$, c-à-d :

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu(x, u) \mid (x, u) \quad J_x \subseteq [0, 1] \quad (\text{A.20})$$

où $\int \int$ dénote l'union de tous les éléments du produit cartésien sur x et u avec $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$. A chaque point fixe x de X , J_x est l'appartenance primaire de x , et x est appelé variable primaire.

Définition 2

A chaque valeur de x , noté $x = x'$, le plan bidimensionnel dont les axes sont u et $\mu_{\tilde{A}}(x', u)$ est appelé tranche verticale (vertical slice) de $\mu_{\tilde{A}}(x, u)$. Une fonction d'appartenance secondaire est une tranche verticale de $\mu_{\tilde{A}}(x, u)$.

Donc pour $x' \in X$ et $\forall u \in J_{x'} \subseteq [0, 1]$, on a :

$$\mu_{\tilde{A}}(x = x', u) \equiv \mu_{\tilde{A}}(x') = \int_{u \in J_{x'}} f_{x'}(u) \mid u \quad J_{x'} \subseteq [0, 1] \quad (\text{A.21})$$

où $0 \leq f_{x'}(u) \leq 1$. Puisque $\forall x'$, cet x' appartient à X , c-a-d $x' \in X$, alors nous notons la fonction d'appartenance secondaire par $\mu_{\tilde{A}}(x)$, qui est une fonction d'appartenance floue *type-1*. En se basant sur le concept des ensembles secondaires, on peut réinterpréter un ensemble flou *type-2* comme l'union de tous les ensembles secondaire, c-à-d, en utilisant l'équation (d.2), on peut écrire \tilde{A} sous la forme suivante :

$$\tilde{A} = \left\{ (x, \mu_{\tilde{A}}(x)) \mid \forall x \in X \right\} \quad (\text{A.22})$$

où par :

$$\tilde{A} = \int_{x \in X} \mu_{\tilde{A}}(x) | x = \int_{x \in X} \left[\int_{u \in J_x} f_x(u) | u \right] | x \quad J_x \subseteq [0, 1] \quad (\text{A.23})$$

Définition 3

Le domaine de la fonction d'appartenance secondaire est appelée degré d'appartenance primaire de x , noté J_x telle que $J_x \subseteq [0, 1] \quad \forall x \in X$.

Définition 4

L'amplitude de la fonction d'appartenance secondaire est appelée degré d'appartenance secondaire, notée $f_x(u)$.

Définition 5

Un ensemble flou intervalle est un ensemble flou *type-2* dont toutes les fonctions d'appartenance secondaires sont des ensembles *type-1* de formes intervalles, c-a-d

$$f_x(u) = 1, \quad \forall u \in J_x \subseteq [0, 1], \quad \forall x \in X. \quad (\text{A.24})$$

Les ensembles flous intervalle reflètent l'uniformité de l'incertitude au niveau de la fonction d'appartenance primaire. Ce type de fonctions d'appartenance est le plus souvent utilisé dans les systèmes flous *type-2*. Notons que ce type de fonctions d'appartenance est représenté seulement par ses domaines (intervalles) qui peuvent être exprimés en fonction des bornes gauches et droites $[l, r]$ ou par leurs centres et largeurs $[c-s, c+s]$ où $c = (l+r)/2$ et $s = (r-l)/2$.

Définition 6

Supposons que chaque fonction d'appartenance secondaire d'un ensemble flou *type-2* possède un seul degré d'appartenance unitaire.

On définit alors une fonction d'appartenance principale comme l'union de tous ces points :

$$u_{\text{principale}}(x) = \int_{x \in X} u | x \quad \text{où } f_x(u) = 1 \quad (\text{A.25})$$

Pour un ensemble flou *type-2* intervalle, nous définissons la fonction d'appartenance principale comme étant l'union de toutes les valeurs d'appartenance moyennes de la fonction

d'appartenance primaire. Notons que lorsque l'incertitude des fonctions d'appartenance disparaît, la fonction d'appartenance d'un ensemble flou *type-2* se réduit à la fonction d'appartenance principale.

Définition 7

L'incertitude dans un ensemble flou *type-2* \tilde{A} est représentée par une région bornée désignée par *FOU* «*Footprint Of Uncertainty*». C'est l'union de toutes les appartenances primaires (voir figure A.2):

$$FOU(\tilde{A}) = \bigcup_{x \in X} J_x \quad (\text{A.26})$$

Définition 8

La fonction d'appartenance supérieur et la fonction d'appartenance inférieur sont des fonctions d'appartenance de *type-1*, et elles sont les frontières du *FOU* d'un ensemble flou \tilde{A} *type-2*. La première sera notée par $\bar{\mu}_{\tilde{A}}(x) \quad \forall x \in X$. Par contre, la seconde est notée par $\underline{\mu}_{\tilde{A}}(x) \quad \forall x \in X$ alors nous avons (voir figure A.2):

$$\begin{aligned} \bar{\mu}_{\tilde{A}}(x) &= \overline{FOU(\tilde{A})} & \forall x \in X \\ \underline{\mu}_{\tilde{A}}(x) &= \underline{FOU(\tilde{A})} & \forall x \in X \end{aligned} \quad (\text{A.27})$$

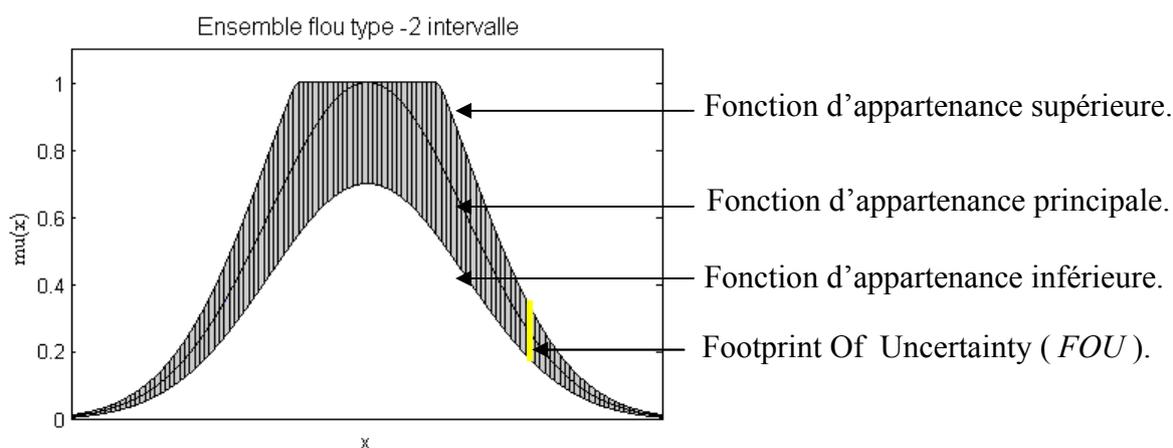


Fig. A.2. Représentation bidimensionnelle d'un ensemble flou *type-2* intervalle.

Résumé :

Le travail présenté dans ce mémoire porte sur l'étude des *systemes flous type-2* et leur application à la commande des systèmes non linéaires. Leurs capacités et le problème de réduction de type ont été étudiés. L'optimisation de leurs performances est aussi étudiée en utilisant les algorithmes génétiques. Afin de vérifier et de comparer leurs performances avec celles des systèmes flous type-1, trois exemples de commande ont été utilisés : système non linéaire, pendule inversé et robotique mobile. D'abord, les contrôleurs flous (type-1 et type-2) ont été conçus par un réglage empirique de leurs paramètres. Dans la deuxième partie, cette tâche est assignée à un algorithme génétique. La robustesse des différents contrôleurs a été testée en considérant la présence du bruit. Les résultats de simulation ont montré la supériorité des *systemes flous type-2*, mais, au prix de la complexité.

Abstract:

The work presented in this paper focuses on the study of the *type-2 fuzzy systems* and their application to the control of nonlinear systems. Their capacity and the problem of the *type reduction* have been studied. Optimizing their performance is also studied by using genetic algorithms. To verify and compare their performance with those type-1 fuzzy systems, three examples of command have been used: non-linear system, inverted pendulum and mobile robotics. First, the fuzzy controllers (type-1 and type-2) were designed by an empirical adjustment of their parameters. In the second part, this task is assigned to a genetic algorithm. The robustness of different controllers was tested by examining the presence of noise. The simulation results showed the superiority of *type-2 fuzzy systems*, but at the cost of complexity.

ملخص

يشمل العمل المعروف في هذه المذكرة دراسة لأنظمة المنطق الغامض نوع-2 وتطبيقها على مراقبة الأنظمة الغير خطية. قدراتها و مشكلتها تخفيض النوع مدروسة. أدائها الأمتل أيضا مدروس عن طريق استخدام الخوارزميات الجينية. من أجل التحقق و مقارنة أدائها مع أداء أنظمة المنطق الغامض نوع-1، ثلاثة أمثلة مراقبة تم استخدامها: نظام غير خطي، نواس مقلوب و ريبوت متحرك. أولاً، المراقب الغامض (نوع-1 ونوع-2) تم تصميمهما عن طريق تعديل تجريبي لعناصرهما. في المرحلة الثانية، هذه المهمة أسندت إلى الخوارزمية الجينية. صلابة أجهزة التحكم تم اختبارها معتمدين حالة وجود ارتيابات. النتائج أظهرت تفوق أنظمة المنطق الغامض نوع-2 ولكن على حساب نوع من التعقيد.