



32-530-533-1

UNIVERSITE SAAD DAHLAB DE BLIDA

Faculté de Technologie
Département d'Electronique



MEMOIRE DE MAGISTER

En électronique
Spécialité : Communication

Un processus d'apprentissage hybride pour la base de connaissances d'un système basé sur des règles floues.

Par

MELEK HIND Epouse Blidia



Devant le jury composé de :

M.Bounekhla	Professeur	USD-Blida	Président
H.Salhi	Professeur	USD-Blida	Examineur
A.Guessoum	Professeur	USD-Blida	Examineur
K.Benmansour	Maitre de conférences(A)	UYF-Médéa	Examineur
M.Hadj sadok	Maitre de conférences(A)	USD-Blida	Promoteur



UNIVERSITE SAAD DAHLAB DE BLIDA

Faculté de Technologie

Département d'Electronique

MEMOIRE DE MAGISTER

En électronique

Spécialité : Communication

Un processus d'apprentissage hybride pour la base de connaissances d'un système basé sur des règles floues.

Par

MELEK HIND Epouse Blidia

Devant le jury composé de :

M.Bounekhla	Professeur	USD-Blida	Président
H.Salhi	Professeur	USD-Blida	Examineur
A.Guessoum	Professeur	USD-Blida	Examineur
K.Benmansour	Maitre de conférences(A)	UYF-Médéa	Examineur
M.Hadj sadok	Maitre de conférences(A)	USD-Blida	Promoteur

Blida, 2013

RESUME

Dans les systèmes basés sur les règles floues, la dérivation automatique des règles floues linguistiques à partir des données numériques est une tâche importante. Ces derniers temps, plusieurs contributions ont été proposées dans la littérature pour faire face à ce problème, tels que les réseaux de neurones, les algorithmes génétiques et les méthodes Adhoc-Data-Driven ...etc. .

Dans cette thèse nous introduisons une nouvelle approche pour l'apprentissage des règles floues avec des algorithmes d'optimisation par colonie de fourmis (ACO). Pour ainsi faire, cette tâche d'apprentissage sera formulée comme un problème d'optimisation combinatoire.

Notre processus d'apprentissage sera basé sur une méthodologie des règles coopératives (COR), dont ont fournit un espace de recherche qui nous permet d'obtenir des systèmes flous avec une bonne interprétabilité et une bonne exactitude.

Un spécifique algorithme de colonie de fourmis, le meilleur- mauvais système de fourmis, BWAS (Best-Worst-Ant-System) sera employé à cette fin, en raison de son efficacité pour la résolution d'une multitude de problèmes d'optimisations.

Les mots clés :

Système flou, optimisation, Algorithmes de colonie de fourmis, ACO, Méthodologie COR, Algorithme BWAS, Interprétabilité, Exactitude, Efficacité.

ABSTRACT

In the systems based on fuzzy rules, the automatic derivation of linguistic fuzzy rules from numerical data is an important task. Lately, a large number of contributions based on techniques such as neural networks, genetic algorithms and Adhoc-Data-Driven methods were proposed to deal with this problem. In this thesis we introduce a new approach to learning fuzzy rules with algorithms ant colony optimization (ACO). To do so, the learning task is formulated as a combinatorial optimization problem.

Our learning process is based on a methodology of cooperative rules (COR), we provide a search space that enables us to obtain fuzzy models with a good interpretability and good accuracy. A specific ant colony algorithm, the Best-Worst- Ant- System (BWAS), will be used for this purpose because of its effectiveness in solving many other optimization problems.

Keywords:

Systems based on fuzzy rules, optimization, ant colony algorithms, ACO, COR Methodology, Algorithm BWAS, Interpretability, Accuracy, Efficiency.

ملخص

في مجال النمذجة مع الأنظمة القائمة على قواعد غامضة و الاشتقاق من قاعدة المعلومات التلقائية من بيانات رقمية مهمة صعبة, لذا في الآونة الأخيرة عدد كبير من المساهمات القائمة على تقنيات مختلفة كالشبكات العصبية و الخوارزمية الجينية و مستعمرة النمل اقترحت لحل هذا المشكل. في هذه الأطروحة نقدم تقنية جديدة للتعلم مع قواعد غامضة تجمع بين منهجية للقواعد التعاونية باستعمال خوارزمية محددة تقوم على أساس تحسين مستعمر النمل.

كلمات البحث :

النمذجة و التحسن ، مستعمرة النمل منهجية القواعد التعاونية تفسير و الدقة و الكفاءة اسوء مجال في مستعمرة النمل

REMERCIEMENTS

Qu'il me soit d'abord permis de remercier et d'exprimer ma gratitude envers le bon dieu, qui m'a donné la patience et le courage pour que je puisse terminer ce travail.

Je tiens à remercier Monsieur **Hadj sadok M'hamed**, Maître de conférences (A) au département d'Electronique de l'USDB, pour avoir dirigé ce travail ainsi que pour ses nombreux conseils et son soutien tout au long du déroulement de ce mémoire.

Je remercie vivement Monsieur **M. Bounekhla** Professeur au département d'Electronique de l'USDB, pour m'avoir fait l'honneur de présider le jury d'examen.

Je tiens également à remercier Monsieur **A. Guessoum**, Professeur au département d'Electronique de l'USDB, monsieur **H .Salhi** Professeur au département d'Electronique de l'USDB et également monsieur **K.Benmensour** Maître de Conférences (A) à l'UYF de Médéa , pour l'intérêt qu'ils ont porté à mon travail, pour les observations enrichissantes faites dans leurs rapports, ainsi que pour m'avoir fait l'honneur d'être membres de jury.

Enfin, je remercie toute ma famille pour l'affection, la patience et le soutien inconditionnel tout au long de ces longues années d'études, en particulier:

A Ma très chère mère, A mon mari Fodhil, A mon neveu Djallal et mes enfants Yousra, Lyna, Sanaa et Mehdi.

TABLE DES MATIERES

MEMOIRE DE MAGISTER	0
RESUME	0
ABSTRACT	1
ملخص	2
REMERCEments	3
TABLE DES MATIERES	4
LISTE DES ILLUSTRATIONS, GRAPHIQUES et tableaux	7
INTRODUCTION	9
CHAPITRE 1	11
1.1. Introduction.....	11
1.2. Ensemble flou et terminologie	11
1.3. Opérations sur les ensembles flous.....	12
1.4. Les relations floues.....	14
1.5. Opérations sur les relations floues	15
1.6. L'opération d'extension cylindrique.....	17
1.7. Raisonnement approximatif.....	18
1.7.1. Introduction :.....	18
1.7.2. Variables linguistiques :.....	19
1.7.3. Propositions floues :.....	19
1.7.4. Règle de production floue " if - then " :.....	21
1.7.5. Règles d'inférences :.....	23
1.7.6. Représentation de la signification d'une règle " if - then " :.....	24
1.7.7. Implication de Dienes Rescher :.....	25
1.7.8. Implication de Lukasiewicz :.....	25
1.7.9. Implication de Zadeh :.....	26
1.7.10. Implication stochastique :.....	26
1.7.11. Implication de Mamdani :.....	26
1.7.12. Inference d'une regle " if - then " :.....	26
1.7.13. Représentation d'un ensemble de règles :.....	35

CHAPITRE 2	37
2.1. Introduction.....	37
2.2. Définition de colonie de fourmis.....	37
2.3. La stigmergie.....	38
2.4. Optimisation par colonie de fourmis et problème de voyageurs de commerce	39
2.4.1. Définition du problème	39
2.4.2. Principe général : l'algorithme Ant system	40
2.4.3. Variantes.....	43
2.4.4. Choix des paramètres.....	45
2.5. Autres problèmes combinatoires	46
2.6. Formalisation et propriétés d'un algorithme de colonie de fourmis..	46
2.6.1. Formalisation	47
2.6.2. Phéromones et mémoire.....	49
2.6.3. Intensification/diversification	49
2.7. Conclusion.....	51
CHAPITRE 3	52
3.1. Introduction.....	52
3.2. Méthodologie des règles coopératives COR	54
3.3. Introduction à l'optimisation de colonie de fourmis ACO	57
3.4. Le meilleur mauvais système de fourmis (BWAS).....	57
3.5. L'apprentissage du système flou par l'approche BWAS	58
3.5.1. Représentation du problème pour l'apprentissage des règles floues coopératives.....	58
3.5.2. Information heuristique.....	60
3.5.3. Initialisation du phéromone	62
3.5.4. Fonction d'adéquation (fitness).....	62
3.5.5. L'algorithme du meilleur mauvais système de fourmis BWAS ..	62
3.6. Conclusion.....	67
CHAPITRE 4	68
4.1. Description du ballon dirigeable.....	68
4.2. Modélisation du ballon dirigeable	68
4.3. Contrôle du ballon dirigeable par un contrôleur flou simple.....	81
4.4. Synthèse d'un premier contrôleur flou	85

4.5. Organigramme de l'approche COR - BWAS	92
4.6. Résultats de la simulation d'un contrôleur flou optimisé	96
APPENDICE A	110
APPENDICE B	112
REFERENCES	114

LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX

FIGURE 1 : REPRESENTATION D'UNE FONCTION D'APPARTENANCE	21
FIGURE 2 : REPRESENTATION, D'UN ENSEMBLE DE TERMES ($LX = LY$).....	27
FIGURE 3 : FONCTIONS D'APPARTENANCES μ_{PM} ET μ_{NS} INTERPRETANTS LES PROPOSITIONS "X IS PM" ET " Y IS NS"	28
FIGURE 4 : FUZZIFICATION DE LA VALEUR $x_0 = 3$	31
FIGURE 5 : REPRESENTATION GRAPHIQUE DE L'INFERENCE D'UNE SIMPLE REGLE	34
FIGURE 6 : COMPORTEMENT DES FOURMIS.....	37
FIGURE 7 : RECHERCHE DU PLUS COURT CHEMIN.....	38
FIGURE 8 : LE PROBLEME DU VOYAGEUR DE COMMERCE OPTIMISE PAR L'ALGORITHME AS	43
FIGURE 9 : L'ALGORITHME COR.....	56
FIGURE 10 : EXEMPLE DE PROCESSUS D'APPRENTISSAGE.	59
FIGURE 11 : PROCESSUS D'ASSIGNEMENT D'HEURISTIQUE	60
FIGURE 12 : ASSIGNEMENT HEURISTIQUE DE LA REGLE R_i POUR CHAQUE CONSEQUENCE DU SYSTEME AVEC DEUX VARIABLES D'ENTREES ET TROIS CONSEQUENCES (N_c) DE LA SORTIE FLOUE.	61
FIGURE 13: PROCESSUS D'APPRENTISSAGE	61
FIGURE 14 :PROCESSUS DE RECHERCHE LOCALE UTILISE DANS L'ALGORITHME BWAS. 65	65
FIGURE 15 : DIMENSION DU DIRIGEABLE ET POSITION DES PROPULSEURS.....	68
FIGURE 16 : REPRESENTATION DES REPERES.....	68
FIGURE 17 : VUE DU PROFIL ; ROTATION AUTOUR DE L'AXE Y (ANGLE DE TANGAGE).....	69
FIGURE 18 : VUE DE L'AVANT, ROTATION AUTOUR DE L'AXE X, ANGLE DE ROULIS.....	70
FIGURE 19: VUE DE DESSUS ; LE CAP, ROTATION AUTOUR DE L'AXE Z (ANGLE DE LACET) 70	70
FIGURE 20 : REPRESENTATION EN 3DDL D'UN CORPS RIGIDE	72
FIGURE 21: CONTROLE FLOU DU PROCESSUS LATERAL	82
FIGURE 22 : SCHEMA SYNOPTIQUE DU SYSTEME.....	83
FIGURE 23 : SYNOPTIQUE DU MODE LONGITUDINAL	83
FIGURE 24: SYNOPTIQUE DU MODE LATERAL	84
FIGURE 25 : SIMULATION DE NOTRE SYSTEME SOUS SIMULINK	85

FIGURE 26 : BASE DES REGLES	86
FIGURE 27 : FONCTIONS D'APPARTENANCES ENTREE E	87
FIGURE 28 : FONCTIONS D'APPARTENANCES VARIATION D'ERREUR Δe	88
FIGURE 29 : SORTIE DU CONTROLEUR (COMMANDE EA)	89
FIGURE 30 : LA SORTIE DU PROCESSUS	89
FIGURE 31 : SUIVI DE LA TRAJECTOIRE.....	90
FIGURE 32 : SUPERPOSITION DES DEUX SIGNAUX (SORTIE DU PROCESSUS – SIGNAL DE REFERENCE)	90
FIGURE 33 : NOUVELLE BASE DE REGLES	96
FIGURE 34: FONCTIONS D'APPARTENANCES ENTREES E	97
FIGURE 35: FONCTIONS D'APPARTENANCES VARIATION D'ERREUR ΔE	98
FIGURE 36: SORTIE DE LA COMMANDE FLOUE (DEUXIEME CONTROLEUR FLOU)	99
FIGURE 37 :SIGNAL DE SORTIE DU PROCESSUS	100
FIGURE 38 :SIGNAL DE REFERENCE.....	101
FIGURE 39: SUIVI DE LA TRAJECTOIRE.....	101
FIGURE 40 : BASE DES REGLES	102
FIGURE 41: FONCTIONS D'APPARTENANCE ENTREE E	102
FIGURE 42: FONCTIONS D'APPARTENANCES VARIATION D'ERREURS ΔE	103
FIGURE 43: SORTIE DU CONTROLEUR FLOU	104
FIGURE 44: SORTIE DU SIGNAL OPTIMAL DU SYSTEME	105
FIGURE 45: SUIVI DE LA TRAJECTOIRE.....	105
FIGURE 46 : SUPERPOSITION DES SUIVIS DE LA TRAJECTOIRE	106
FIGURE 47: SUPERPOSITION DES DEUX SIGNAUX (SORTIE DU PROCESSUS OPTIMISEE ET DE REFERENCE)	106
FIGURE 48 : SUPERPOSITION DES TROIS SIGNAUX (REFERENCE –SORTIE DU PROCESSUS (1 ^{ER} ESSAI) –SORTIE DU PROCESSUS OPTIMISEE)	107

INTRODUCTION

Les systèmes basés sur les règles floues (FRBS), constituent actuellement un des domaines les plus importants dans l'application de la théorie des ensembles flous. La définition automatique de la base des connaissances (Base des règles et base de données), pour une application concrète est considérée comme un problème dur. Pour résoudre ce type de problème, plusieurs contributions dans le domaine, ont été proposées en littérature. On citera les méthodes AdHoc-Data-Driven [1], les réseaux de neurones [2], les algorithmes génétiques [2, 3], la classification floue [3],....etc.

La mise en œuvre des systèmes flous FRBS, impose les conditions suivantes [3] :

- **Interprétabilité** : elle se rapporte aux possibilités du système flou à exprimer le comportement du système d'une manière compréhensible. C'est une propriété subjective qui dépend du nombre de règles floues à réduire.
- **Exactitude** : elle se rapporte aux possibilités que possède le système flou pour représenter loyalement le système réel.

Avec ces deux aspects, la vitesse du processus de dérivation des règles floues a également une importance vitale. En effet, un apprentissage rapide possède plusieurs avantages intéressants, telle que la possibilité d'être employé comme un mécanisme antérieur pour comprendre les caractéristiques du problème à résoudre [5].

Dans cette contribution, nous présentons une nouvelle approche pour le problème d'apprentissage des règles floues basée sur la méthodologie des règles coopératives COR (cooperative rules methodology)[6], et on se servira des algorithmes d'optimisation de colonie de fourmis ACO pour trouver un bon équilibre entre les trois conditions mentionnées (interprétabilité, exactitude et rapidité de l'apprentissage).

Pour appliquer les algorithmes ACO au problème d'apprentissage des règles floues, on doit formuler le problème comme un problème d'optimisation combinatoire, et les caractéristiques liées aux algorithmes ACO, tels que

l'information heuristique, l'initialisation de phéromone, la fonction fitness, la construction d'une solution et la mise à jour de phéromone doivent être introduits.

La méthodologie des règles coopératives COR accomplit la tâche d'apprentissage en deux étapes :

- La Construction de l'espace de recherche.
- La sélection de l'ensemble des règles floues les plus coopératives.

Ce travail sera guidé par une mesure de l'erreur globale (MSE)

Puisque l'information heuristique sur la qualité locale de chaque conséquent pour chaque règle est disponible, les algorithmes ACO s'adaptent correctement à cette méthodologie, pour exécuter une meilleure recherche, et augmenter la vitesse de convergence, en produisant ainsi des systèmes flous interprétables et précis d'une manière rapide.

Le meilleur mauvais algorithme du système des fourmis BWAS (Best-Worst-Ant-System [7]) est employé, pour sa bonne efficacité, montrée dans la résolution de plusieurs problèmes d'optimisations en littérature [8,9].

Le mémoire est subdivisé en quatre chapitres :

Dans le premier chapitre, nous avons présenté la théorie des ensembles flous, et le raisonnement approximatif alternatif au raisonnement exacte dont l'étude a été présentée dans le cadre de la théorie des possibilités.

Le deuxième chapitre définit la métaheuristique de colonie de fourmis, en mettant en évidence l'élaboration de ces algorithmes dans la résolution des problèmes combinatoires.

Dans le troisième chapitre, nous avons exposé la méthode de dérivation de la base des connaissances d'un système flou par l'approche hybride (COR-BWAS).

Le quatrième chapitre présente les résultats prometteurs de simulation par l'application de la méthode de dérivation de la base des connaissances sur un ballon dirigeable.

CHAPITRE 1

ENSEMBLE FLOU ET LOGIQUE FLOUE

1.1. Introduction

Les concepts théoriques de la logique floue ont été formulé en 1965 par le professeur Lotfi Zadeh , il a introduit la notation des sous-ensembles flous qui s'appuie sur le degré d'appartenance d'un élément à un sous-ensemble .

Cette approche, permet de modéliser des connaissances imparfaitement décrites ou imprécises .

Dés 1975, Mamdani et Assilian ont publié des premiers résultats qui permettent une exploitation de cette théorie dans le système de réglage, en utilisant des structures de contrôleur relativement simple, ils ont obtenu de meilleurs résultats lors de la commande de certains processus que ceux fournis par un régulateur standard .

1.2. Ensemble flou et terminologie

Soit U un ensemble dénombrable ou non et F un sous ensemble de U .

$$F \subset U \quad (1.1)$$

Utilisons pour indiquer qu'un élément u de U appartient à F , le concept de " fonction caractéristique" $\mu_F (u)$, on écrira alors :

$$\mu_F = \begin{cases} 1 & \text{si } u \in F \\ 0 & \text{si } u \notin F \end{cases} \quad (1.2)$$

On définit alors le sous ensemble flou \tilde{F} de U comme étant l'ensemble des couples ordonnés :

$$\tilde{F} = \{ (u, \mu_F(u), \forall u \in U) \} \quad (1.3)$$

Où, $\mu_F(u)$ est une fonction caractéristique d'appartenance qui prend ses valeurs dans le segment $[0.1]$ [10]. Elle indique le degré ou niveau d'appartenance de l'élément u à \tilde{F} , ainsi u de U pourrait :

- ne pas appartenir à \tilde{F} $\mu_F(\mu) = 0$
- appartenir un peu à \tilde{F} $\mu_F(\mu)$ est voisin de 0

- appartenir assez à \tilde{F} entre 0 et 1
- appartenir fortement à \tilde{F} $\mu_F(\mu)$ est voisin de 1
- appartenir à \tilde{F} $\mu_F(\mu) = 1$

Dans la suite, on utilisera la notation suivante, pour représenter un ensemble flou \tilde{F} :

$$\tilde{F} = \mu_F(u_1) / u_1 + \mu_F(u_2) / u_2 + \dots + \mu_F(u_n) / u_n \tag{1.4}$$

$$\tilde{F} = \sum_{i=1}^n \mu_F(u_i) / u_i \quad , \text{ si le domaine } U \text{ est dénombrable, et :}$$

$$\tilde{F} = \int_U \mu_F(u) / u \quad , \text{ si le domaine } U \text{ est continu.}$$

Les signes " + " et " \int " désignent respectivement une énumération dénombrable et non-dénombrable. Par exemple, l'ensemble flou \tilde{A} tel que :

$$\tilde{A} = \{ (1 , 0.3) , (2 , 0.8) , (3 , 1) \} , \tag{1.5}$$

S'écrira :

$$\tilde{A} = 0.3 / 1 + 0.8 / 2 + 1 / 3 \quad , \tag{1.6}$$

Où 0.3, 0.8 et 1 sont les degrés d'appartenance des éléments 1,2 et 3 de l'ensemble flou \tilde{A} .

1.3. Opérations sur les ensembles flous

Soient \tilde{A} et \tilde{B} deux sous ensembles flous de U, on définira :

1) L'union : $\tilde{A} \cup \tilde{B}$, par le plus petit sous ensemble flou qui contient à la fois \tilde{A} et \tilde{B} . c'est-à-dire :

$$\forall u \in U : \mu_{A \cup B}(u) = \text{MAX} (\mu_A(u) , \mu_B(u)) . \tag{1.7}$$

2) L'intersection : $\tilde{A} \cap \tilde{B}$, par le plus grand sous ensemble flou contenu à la fois, dans \tilde{A} et dans \tilde{B} . c'est-à-dire :

$$\forall u \in U : \mu_{A \cap B}(u) = \text{MIN} (\mu_A(u) , \mu_B(u)). \quad (1.8)$$

3) La complémentation : \bar{A} étant le complément de A :

$$\forall u \in U : \mu_{\bar{A}}(u) = 1 - \mu_A(u). \quad (1.9)$$

Ceci est une extension très simple de ces mêmes opérations en logique à deux valeurs (Boolienne), la seule condition à vérifier étant, pour l'intersection, par exemple, la table Boolienne 1 .on peut choisir d'autres extensions, comme :

$\mu_{A \cap B}(u) = \mu_A(u) \cdot \mu_B(u)$, Pour l'intersection et $\mu_{A \cup B}(u) = \min(1, \mu_A(u) + \mu_B(u))$ pour l'union, l'essentiel est que leurs tables de vérités soient vérifiées.

\cap	0	1
0	0	0
1	0	1

Table 1

\cup	0	1
0	0	1
1	1	1

Table 2

Pour représenter plus généralement les trois opérations (\cap , \cup et complémentation) on utilise les normes triangulaires.

Définition 1 :

Une norme triangulaire T-norme " T " dénote une classe de fonctions binaires pouvant représenter l'opération " \cap " et satisfaisant les critères suivants :

1- $a \text{ T } b = b \text{ T } a$,

2- $(a \text{ T } b) \text{ T } c = a \text{ T } (b \text{ T } c)$,

3- si $a \leq c$ et $b \leq d$ alors $a \text{ T } b \leq c \text{ T } d$,

4- $a \text{ T } 1 = a$.

$\text{Min}(a, b)$, $\text{max}(0, a+b-1)$, sont des exemples de T-normes.

Définition 2 :

Une norme triangulaire S-norme "\$", dénote une classe de fonction binaire pouvant représenter l'opération "U" et satisfaisant les trois premiers critères de la T-norme, avec en plus, le critère : $a \$ 0 = a$.

Définition 3 :

L'opération complémentation "□" doit au moins satisfaire les critères suivants[10] :

- 1- $1 - \square(0) = 1$,
 - 2- si $a < b$ alors $\square(a) > \square(b)$,
 - 3- $\square(\square(a)) = a$.
- (1.10)

Grâce à ces trois définitions, on peut généraliser les opérations d'intersection, d'union et de complémentation déjà considérées précédemment.

On aura :

$$\begin{aligned} \mu_{A \cap B}(u) &= \mu_A(u) T \mu_B(u) \\ \mu_{A \cup B}(u) &= \mu_A(u) \$ \mu_B(u) \\ \mu_A(u) &= C(\mu_A(u)) \end{aligned} \quad (1.11)$$

Les normes T et \$ incluent un grand nombre d'opérations de même classe.

1.4. Les relations floues

Tout comme une relation binaire classique, une relation binaire floue peut être considérée comme un ensemble flou de couple (x, y) . Chaque couple (x, y) étant muni d'un degré d'appartenance $\mu_R(x, y)$. Formellement, on a la définition suivante :

Définition 4 :

Si U et V sont deux univers continus et $\mu_R : U \times V \longrightarrow [0, 1]$ alors :

$$R = \int_{U \times V} \mu_R(u, v) / (u, v) \quad (1.12)$$

est une relation floue sur $U \times V$.

Dans le cas où U et V sont dénombrables (discrets), alors :

$$R = \sum_{U \times V} \mu_R(u, v) / (u, v) \quad (1.13)$$

On définit de la même manière, une relation floue tertiaire.

Pour le cas discret, une relation floue binaire est représentée sous une forme matricielle.

Exemple :

Soient $U = \{X_1, X_2, X_3\}$, $V = \{y_1, y_2\}$ et R une relation floue représentée

par :

$$R = 1 / (x_1, y_1) + 0.3 / (x_1, y_2) + 0.8 / (x_2, y_1) + 0.8 / (x_2, y_2) + 0.9 / (x_3, y_1) + 1 / (x_3, y_2) \quad (1.14)$$

Sa forme matricielle sera :

	X_1	X_2	X_3
Y_1	1	0.8	0.9
Y_2	0.3	0.2	1

Table 3

1.5. Opérations sur les relations floues

Les relations floues ont une très grande importance dans divers applications (automatique, modélisation des systèmes,...), elles permettent de décrire les interactions entre les différentes variables.

Soient S et R deux relations floues binaires, on définit alors :

1) L'intersection entre les deux relations R et S définies sur $X \times Y$ comme :

$$\forall (x, y) \in X \times Y : \mu_{R \cap S}(x, y) = \min(\mu_R(x, y), \mu_S(x, y)). \quad (1.15)$$

2) L'union, comme :

$$\forall (x, y) \in X \times Y : \mu_{R \cap S}(x, y) = \max(\mu_R(x, y), \mu_S(x, y)) \quad (1.16)$$

On notera que les opérations min et max peuvent être remplacées respectivement par les opérations T-norme et S-norme.

3) La projection d'une relation R sur un univers Y, dans le cas binaire, comme :

$$\text{Proj R sur Y} = \int_Y^{\text{sup}} x \mu_R(x, y) / y. \quad (1.17)$$

On notera que la définition entière de la projection est très complexe, on se limitera donc aux relations binaires.

Lorsque les domaines sont discrets, on utilise l'opération "max" au lieu de "sup". Si on considère la relation R suivante, on aura alors :

		Y ₁	Y ₂	Y ₃	Y ₄	<u>Proj sur x</u>
R =	X ₁	0.8	1	0.1	0.7	1
	X ₂	0	0.8	0	0	0.8
	X ₃	0.9	1	0.7	0.8	1
<u>Proj sur Y</u>		0.9	1	0.7	0.8	

Table 4

- Le résultat d'une projection d'une relation sur un univers est, comme on le voit, un ensemble flou.

1.6. L'opération d'extension cylindrique

Contrairement à la projection, prolonge (étend) un ensemble flou à une relation binaire, une relation binaire à une relation tertiaire, etc... .

Cette opération est très importante dans la mesure où elle permet, par exemple, de prendre l'intersection d'une relation floue définie sur $X \times Y$ et d'un ensemble flou \tilde{A} défini sur un univers X , si celui-ci est étendu à $X \times Y$.

Exemple : soit les ensembles flous :

$$\begin{aligned}\tilde{A} &= 1/x_1 + 0.8/x_2 + 1/x_3 \\ \tilde{B} &= 0.9/y_1 + 0.8/y_2 + 0.7/y_3 + 0.8/y_4\end{aligned}\quad (1.18)$$

Les extensions cylindriques de \tilde{A} et de \tilde{B} sur $X \times Y$ sont :

	y_1	y_2	y_3	y_4
x_1	1	1	1	1
x_2	0.8	0.8	0.8	0.8
x_3	1	1	1	1

Table 5

	y_1	y_2	y_3	y_4
	0.9	1	0.7	0.8
	0.9	1	0.7	0.8
	0.9	1	0.7	0.8

Table 6

Ainsi la combinaison d'ensembles flous avec des relations floues est possible grâce aux opérations de " projection " et " d'extension". Cette combinaison notée "o" est appelée " COMPOSITION "

- 1) La composition d'un ensemble flou \tilde{A} défini sur X et d'une relation floue R définie sur $X \times Y$, est un ensemble flou, \tilde{B} défini sur Y , tel que :

$$\tilde{B} = A \circ R = \text{Proj de } (C_e(\tilde{A}) \cap R) \text{ sur } Y \quad (1.19)$$

Si, on remplace " \cap " par "min" et "proj" par "max", on aura :

$$\mu_B(y) = \max_x (\mu_A(x), \mu_R(x, y)) \quad (1.20)$$

Cette opération est appelée "composition max-min".

Si, on remplace " \cap " par le produit et "proj" par "max", on aura :

$$\mu_B(y) = \max_x \mu_A(x) \cdot \mu_R(x, y). \quad (1.21)$$

Cette opération est appelée "composition produit max".

1.7. Raisonnement approximatif

1.7.1. Introduction :

Le raisonnement approximatif est la partie la plus riche de la logique floue, elle couvre une variété de règles d'inférence dont les prémisses contiennent des propositions floues.

L'inférence en raisonnement approximatif diffère de celle vue en logique classique. Dans la première la conséquence d'un ensemble de propositions floues donné dépend essentiellement du sens attaché à ces propositions floues. Ainsi l'inférence en raisonnement approximatif est une computation d'ensembles flous qui représentent en fait, les interprétations de certaines propositions floues.

Exemple :

Étant donné deux fonctions d'appartenance μ_A et μ_B représentant l'interprétation d'une proposition floue "X is A" et celle d'une conditionnelle floue "if X is A then Y is B", on peut calculer la fonction d'appartenance qui représente la signification de la conséquence "Y is B" [11].

Avant d'étudier les différentes règles d'inférence en raisonnement approximatif, définissons quelques notions utiles à la représentation des connaissances.

1.7.2. Variables linguistiques :

L'élément fondamental de représentation des connaissances en raisonnement approximatif est la variable linguistique. D'après le professeur Zadeh [12], c'est une variable dont les valeurs sont des mots ou des phrases dans un langage qui peut être naturel ou artificiel.

Exemple :

La variable " vitesse " est linguistique si ses valeurs sont linguistiques, c'est-à-dire, si son ensemble de termes est tel que :

$$L(\text{vitesse}) = \{ \text{très lente , lente , moyenne , rapide ,} \} .$$

En raisonnement approximatif on caractérise souvent une variable linguistique par le quarté $\{ X , LX , X , M \}$, dans lequel, X est le nom symbolique de la variable linguistique, LX est l'ensemble des valeurs linguistique de X (ensemble des termes) X est le domaine physique ou encore l'univers de discours des valeurs quantitatives de X, et M est une fonction sémantique, qui nous donne l'interprétation d'une valeur linguistique en terme d'éléments quantitatifs de X.

Exemple :

X = " vitesse " : variable linguistique

LX = { lente , moyenne , rapide , très lente . . } : ensemble des termes

X = [0 Km /h , 200 Km /h] : domaine physique de X

M, nous donne l'interprétation d'une valeur linguistique quelconque de l'ensemble des termes LX en termes de valeurs du domaine X.

1.7.3. Propositions floues :

Le raisonnement approximatif est utilisé pour représenter et raisonner en connaissances exprimées en " primitifs atomiques ". Par exemple la translation symbolique formelle de l'expression est :

" L'angle a la valeur négative-Small "

- 1) Un symbole θ est choisi pour représenter la variable physique "Angle".
- 2) Un symbole NS est choisi pour représenter la valeur particulière " Négative-Small ".
- 3) L'expression précédente est réécrite de la façon suivante :
" L'angle a la propriété d'être négative -small "
- 4) La représentation symbolique de celle-ci est :
 $\theta \text{ is NS}$

Une telle expression est appelée " proposition floue atomique ". Son interprétation sera définie par un ensemble flou NS et par une fonction d'appartenance μ_{NS} sur le domaine physique E de la variable linguistique Angle.

$$\forall \theta \in E : NS = \mu_{NS} = \text{Fonctions d'appartenance"}. \quad (1.22)$$

$\mu_{NS}(\theta)$ spécifie le degré avec lequel une valeur quantitative particulière de la variable physique " Angle ", appartient à l'ensemble des valeurs physique négative-Small de l'angle.

L'interprétation de l'expression symbolique " $\theta \text{ is NS}$ " nous aide à prendre une décision sur le degré avec lequel cette expression symbolique est satisfaite étant donné une valeur spécifique de l'angle. Ceci se résume à :

- 1) Etant donné une valeur (VA) appelée, variable d'assignement , assignée au symbole θ .

Exemple,

$$VA(\theta) = -1.5 \in \text{au domaine physique E de } \theta . \quad (1.23)$$

La fonction de cette variable d'assignement est de convertir le symbole θ représentant la variable physique " Angle " en une valeur physique de E .

- 2) On détermine le degré d'appartenance de la valeur -1.5 dans μ_{NS} , soit, $\mu_{NS}(-1.5) = 0.7$. ce degré d'appartenance représente le degré avec lequel l'expression symbolique $\theta \text{ is NS}$ est satisfaite.

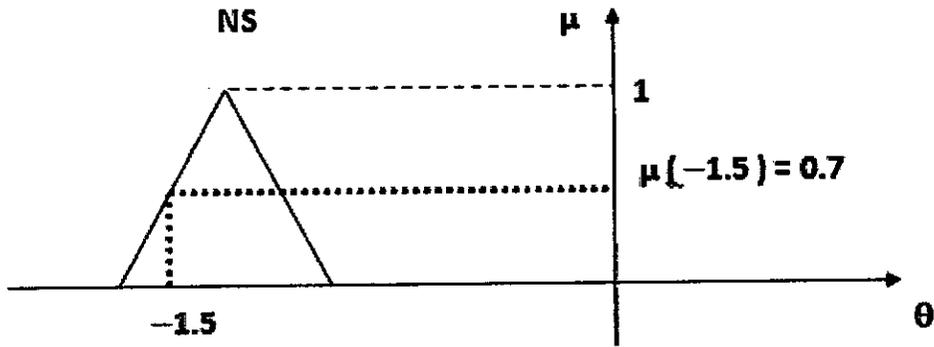


Figure 1 : Représentation d'une fonction d'appartenance

En se basant sur cette notion de proposition floue atomique et en utilisant les coordinations " and ", " or ", " not " et " if – then " , on peut former des propositions floues composées plus complexes, et représenter des connaissances plus larges ,

exemple :

X is A and Y is B.

X is A or Y is B.

X is not A .

(X is A and Y is not B) or Z is Cetc

(1.24)

La signification de ces propositions floues composées sera donnée par l'interprétation de ces coordonneurs " and " , " or " et " not " comme conjonction , disjonction et négation .

1.7.4. Règle de production floue " if - then " :

Une conditionnelle floue ou une règle de production floue " If – Then " est exprimée symboliquement par :

If (proposition floue 1) then (proposition floue 2)

Où proposition floue 1 et 2 peuvent être atomiques ou composées. La règle de production " If – then " est utilisée dans le domaine de l'automatique pour

décrire le rapport entre les variables d'état de processus sous contrôle et les variables de sortie de contrôle.

Exemple, si θ et θ' sont deux variables d'état d'un processus donné, et U est une variable de sortie de contrôle, alors :

$$\text{If } \theta \text{ is NL and } \theta' \text{ is PL then } U \text{ is NS} \quad (1.25)$$

est une expression symbolique de rapport suivant, énoncé en langage naturel. Si la valeur courante de l'angle θ est très inférieure à la référence zéro (NL : Négative Large) et que la dernière variation de l'angle est très supérieure à zéro (PL : Positive Large), alors une petite diminution de la précédente valeur de sortie de contrôle U s'impose (NS : Négative Small).

Soit deux propositions floues atomiques :

$$p : X \text{ is } A \quad \text{et} \quad q : Y \text{ is } B \quad (1.26)$$

A et B étant deux ensembles flous définis dans le même univers de discours U . La règle de production :

$$\text{If } X \text{ is } A \text{ then } Y \text{ is } B$$

Affirme explicitement qu'on peut calculer une valeur de Y seulement si on dispose d'une valeur de X . Si une valeur de Y est donnée, il n'existe pas de règle explicite qui nous permette de déterminer la valeur de X . Ainsi la causalité est seulement dans la direction $X \longrightarrow Y$.

La signification de cette règle de production symbolique est représentée comme une relation floue définie sur $X \times Y$, où X et Y sont les domaines des variables linguistiques X et Y . La structure de cette relation floue est la suivante :

1) La signification de " X is A ", appelée l'antécédent de la règle, est représentée par l'ensemble flou :

$$\tilde{A} = \int_x \mu_A(x) / x \quad (1.27)$$

2) La signification de " Y is B ", appelée la conséquence de la règle, est représentée par l'ensemble flou :

$$\tilde{B} = \int_Y \mu_B(y) / y \quad (1.28)$$

3) La signification de la conditionnelle floue est alors une relation floue μ_R telle que :

$$\forall x \in X \text{ et } y \in Y : \mu_R(x, y) = \mu_A(x) * \mu_B(y) \quad (1.29)$$

Où " * " peut être soit le produit cartésien soit un opérateur d'implication floue quelconque.

1.7.5. Règles d'inférences :

En raisonnement approximatif, on distingue deux règles d'inférence importantes :

- La règle compositionnelle d'inférence.
- La modus ponens généralisé (G.M.P).

La première utilise une relation floue pour représenter explicitement la connexion entre deux propositions floues. La seconde utilise une règle de production " If – then ".

- La règle modus ponens généralisé le schéma d'inférence symbolique suivant :

$$\begin{array}{l} \text{Prémisse 1} \quad : \quad S_1 \quad \text{is} \quad Q_1 \\ \text{Prémisse 2} \quad : \quad \text{If } S_1 \quad \text{is} \quad P_1 \quad \text{then } S_2 \quad \text{is} \quad P_2 \end{array} \quad (1.30)$$

$$\text{Conséquence} : \quad S_2 \quad \text{is} \quad Q_2 .$$

Où S_1 et S_2 sont les noms symboliques d'objets et P_1 , P_2 , Q_1 et Q_2 sont des propriétés quelconques de ces objets .

- La règle compositionnelle d'inférence peut être considérée comme un cas spécial de la GMP, sa forme symbolique générale est :

$$\text{Prémisse 1} \quad : \quad S_1 \quad \text{is} \quad Q_1 \quad (1.31)$$

Premisse 2 : $S_1 \quad R \quad S_2$

Conséquence : $S_2 \quad \text{is} \quad Q_2$

Où " $S_1 \quad R \quad S_2$ " se lit " S_1 est en relation avec S_2 " et sa signification est représentée comme une relation floue μ_R . Donc, on a une relation floue R à la place de la règle de production " if – then ".

Cette règle compositionnelle d'inférence a été définie par Zadeh [12] de la manière suivante :

Si R est une relation floue de $U \times V$, et \tilde{A} est un ensemble flou de U , alors l'ensemble flou \tilde{B} de V induit par \tilde{A} est donné par la composition de R et \tilde{A} .

$$\tilde{B} = \tilde{A} \circ R$$

\tilde{A} joue le rôle d'une relation unaire, l'opération composition " \circ " a été décrite précédemment comme suit :

$$\tilde{B} = \tilde{A} \circ R = \text{proj de } (\text{Ce } (\tilde{A}) \cap R) \text{ sur } Y. \quad (1.32)$$

En remplaçant " \cap " par " min " et " proj " par " max " , on obtient alors la règle compositionnelle de Zadeh .

On notera que quand la règle compositionnelle d'inférence est utilisée, une relation floue explicite R doit être donnée à priori, ce qui n'est pas nécessaire pour la GMP.

1.7.6. Représentation de la signification d'une règle " if - then " :

Un grand nombre de relations exprimées sous forme d'implication floues peuvent être utilisées pour la représentation de la signification d'une règle de production :

If (proposition 1) then (proposition 2).

Cette diversité d'implications floues sont générées en employant les normes triangulaires T-norme et S-norme , grâce à l'apport de la théorie de la "

logique multi-évaluée " de LUKASIEWICZ [13]. Cette logique, bien appropriée à la formulation de la théorie des ensembles flous, contient une infinité de valeurs de vérités appartenant toutes au segment $[0, 1]$.

Etant donné deux propositions floues p et q , représentant respectivement l'antécédent et la conséquence d'une règle de production "If - then":

$$p = "X \text{ is } A" \quad \text{et} \quad q = "Y \text{ is } B"$$

Définissons quelques implications floues les plus utilisées en littérature.

1.7.7. Implication de Dienes Rescher :

L'interprétation de la règle floue "If p then q " est la même que celle de la proposition composée "not p or q ". sachant que la coordination "or" est exécutée par l'union, laquelle peut être remplacée par "max" et que "not p " est $\bar{p} = 1 - p$, on aura, en utilisant l'opération d'extension :

$$\begin{aligned} \mu_{(\text{not } p \text{ or } q)} &= \max (\mu_{ce(\bar{p})} (x, y) , \mu_{ce(q)} (x, y)) \\ &= \max (1 - \mu_{ce(p)} (x, y) , \mu_{ce(q)} (x, y)) \end{aligned} \quad (1.33)$$

C'est l'implication de Dienes-Rescher, elle est notée R_b et est définie comme suit :

$$\mu_{R_b} (x, y) = \max (1 - \mu_A (x) , \mu_B (y)) \quad (1.34)$$

$\mu_A (x)$ et $\mu_B (y)$ représentent respectivement les significations de l'antécédent et de la conséquence de la règle.

1.7.8. Implication de Lukasiewicz :

Cette implication, comme la précédant est basée sur l'équivalence :

$$"p \longrightarrow q" = \text{not } p \text{ or } q,$$

mais, pour l'opérateur "or" on utilise la somme bornée "min $(1, 1 - p + q)$ " au lieu du "max". ceci conduit à la relation R_a , définie comme suit :

$$\mu_{R_a} (x, y) = \max \min (1 , 1 - \mu_A (x) + \mu_B (y)) \quad (1.35)$$

1.7.9. Implication de Zadeh :

En logique à deux valeurs, $p \longrightarrow q$ a la même valeur de vérité que " $(p \wedge q) \sqcap \bar{p}$ ". C'est cette même équivalence que Zadeh utilise [12], pour obtenir :

$$\mu_{R_m}(x, y) = \max [\min (\mu_A(x), \mu_B(y)), 1 - \mu_A(x)] \quad (1.36)$$

1.7.10. Implication stochastique :

Cette implication est basée sur l'égalité :

$$P(A/B) = 1 - P(A) + P(A).P(B) \quad (1.37)$$

Elle s'écrit :

$$\mu_R(x, y) = \min (1, 1 - \mu_A(x) + \mu_A(x) \cdot \mu_B(y)) \quad (1.38)$$

1.7.11. Implication de Mamdani :

C'est l'implication la plus importante et la plus utilisée dans le domaine de l'automatique (contrôle des processus). Sa définition est basée sur l'opération intersection " \cap ", c'est-à-dire sur l'équivalence :

$$"p \longrightarrow q" = "p \wedge q".$$

Elle est notée R_c et définie comme suit :

$$\mu_{R_c}(x, y) = \min (\mu_A(x), \mu_B(y)) \quad (1.39)$$

1.7.12. Inference d'une règle "if - then ":

Soit la règle de production :

$$\text{If } X \text{ is } A \text{ then } Y \text{ is } B, \quad (1.40)$$

Dans laquelle les variables linguistiques X et Y sont définies respectivement sur les domaines U et V , et les valeurs linguistiques A et B appartiennent aux ensembles des termes respectives des domaines U et V .

Selon la nature des univers de discours U et V , discret ou continu, on distingue fondamentalement deux approches pour manipuler une telle règle.

Dans la première approche la fonction d'appartenance d'un ensemble flou est représentée comme un vecteur de nombres (domaine discret), dans la seconde , elle est exprimée comme une forme fonctionnelle typique (triangle , cloche , trapèze , etc.). Comme ces deux approches ne sont pas principalement différentes, nous utiliserons la première pour une meilleure illustration du mécanisme d'inférence de notre règle à travers un exemple.

Exemple : inférence d'une règle " If - then "

Considérons les sept ensembles flous de la figure 2, et soit à inférer la simple règle :

If X is PM then Y is NS (1.41)

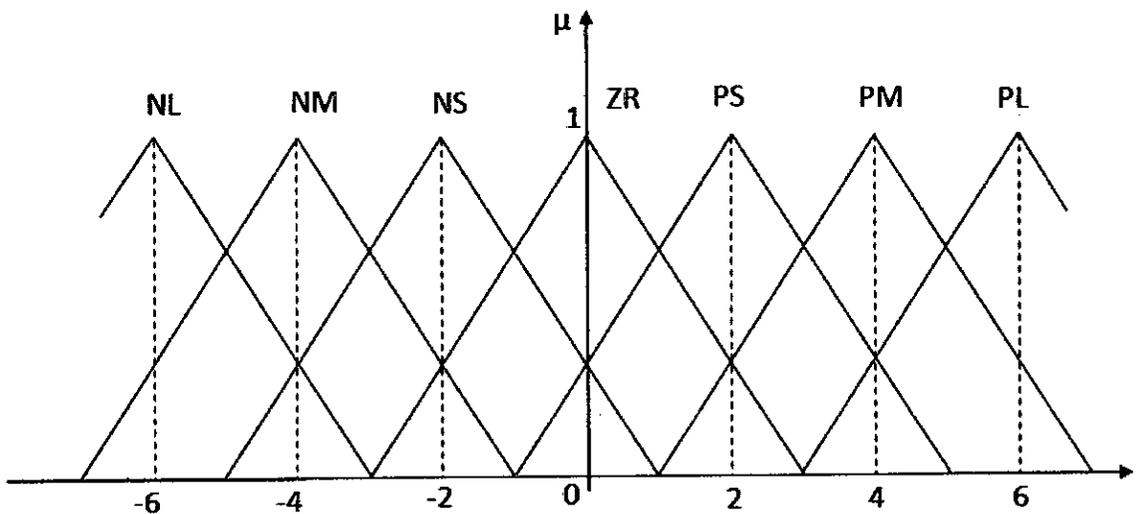


Figure 2 : Représentation, d'un ensemble de termes ($LX = LY$).

- Les variables linguistiques X et Y sont définis sur des domaines identiques $U = V = [-6, +6]$.
- L'un et l'autre de ces domaines contiennent le même ensemble des termes $LX = LY = \{NL, NM, NS, ZR, PS, PM, PL\}$.
- Les interprétations respectives des propositions floues " X is PM " et " Y is NS " sont définies par les ensembles flous $\tilde{P}M$ et $\tilde{N}S$ (ou bien par les fonctions d'appartenance μ_{Pm} et μ_{Ns} de la figure 2).
- On aura en figure 3:

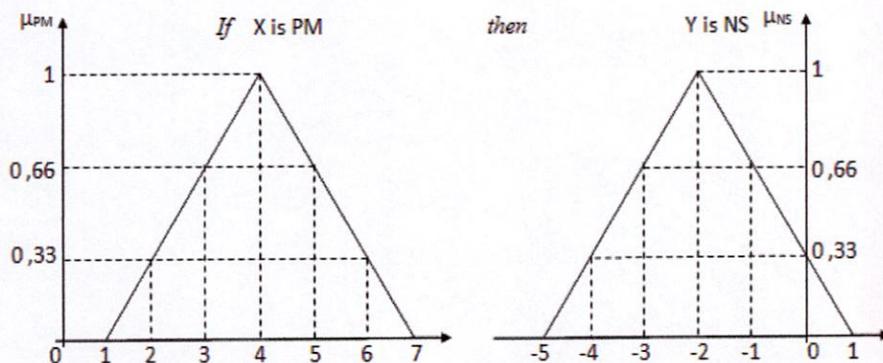


Figure 3 : Fonctions d'appartenances μ_{PM} et μ_{NS} interprétants les propositions "X is PM" et " Y is NS".

- Ainsi, $\tilde{P}M$ et $\tilde{N}S$ sont définis comme suit :

$$\begin{aligned} \tilde{P}M &= 0.33/2 + 0.66/3 + 1/4 + 0.66/5 + 0.33/6 \\ \tilde{N}S &= 0.33/-4 + 0.66/-3 + 1/-2 + 0.66/-1 + 0.33/0 \end{aligned} \quad (1.42)$$

Nous avons vu avant qu'il existait plusieurs façons de représenter la signification d'une règle " If - then ", et qu'en théorie des systèmes, c'est l'implication de Mamdani qui est la plus utilisée, on aura donc :

$$R_c = Ce(\tilde{P}M) \cap Ce(\tilde{N}S),$$

ou:

$$\mu_{R_c}(X,Y) = \min(\mu_{Pm}(X), \mu_{Ns}(Y)).$$

(1.43)

On aura, donc :

	2	3	4	5	6
0	0.33	0.66	1	0.66	0.33
-1	0.33	0.66	1	0.66	0.33
-2	0.33	0.66	1	0.66	0.33
-3	0.33	0.66	1	0.66	0.33
-4	0.33	0.66	1	0.66	0.33

$Ce(\tilde{P}M)$

Table 7

	2	3	4	5	6
0	0.33	0.33	0.33	0.33	0.33
-1	0.66	0.66	0.66	0.66	0.66
-2	1	1	1	1	1
-3	0.66	0.66	0.66	0.66	0.66
-4	0.33	0.33	0.33	0.33	0.33

$Ce(\tilde{N}S)$

Table 8

La relation R_c sur $U \times V$ sera alors :

U

	-6	-5	-----	-1	0	1	2	3	4	5	6
6	0	0	-----	0	0	0	0	0	0	0	0
5	0	0	-----	0	0	0	0	0	0	0	0
4	0	0	-----	0	0	0	0	0	0	0	0
3	0	0	-----	0	0	0	0	0	0	0	0
2	0	0	-----	0	0	0	0	0	0	0	0
1	0	0	-----	0	0	0	0	0	0	0	0
V 0	0	0	-----	0	0	0	0.33	0.33	0.33	0.33	0.33
-1	0	0	-----	0	0	0	0.33	0.66	0.66	0.66	0.33
-2	0	0	-----	0	0	0	0.33	0.66	1	0.66	0.33
-3	0	0	-----	0	0	0	0.33	0.66	0.66	0.66	0.33
-4	0	0	-----	0	0	0	0.33	0.33	0.33	0.33	0.33
-5	0	0	-----	0	0	0	0	0	0	0	0
-6	0	0	-----	0	0	0	0	0	0	0	0

Table 9

Etant donné une valeur spécifique réelle $x_0 = 3$ de X sur $U = [-6, 6]$, l'inférence de notre règle "If - then" consiste à appliquer la règle compositionnelle d'inférence, qui dit que :

Si R est une relation floue de $U \times V$, et \tilde{A} est un sous ensemble flou de U alors le sous ensemble flou \tilde{B} de V induit par \tilde{A} est donné par la composition de R et \tilde{A} .

$$\tilde{B} = \tilde{A} \circ R$$

La valeur $x_0 = 3$ ne pouvant être combinée seule avec la relation R , on procédera d'abord à la transformation de cette valeur x_0 de X , en un ensemble flou \tilde{X}_0 ayant une fonction d'appartenance $\mu_0(x_0)$ égale à zéro partout sauf au point X_0 , figure

(3), c'est la procédure de fuzzification.

$$\mu_x(x_0) = \begin{cases} 1 & \text{si } x = x_0 \\ 0 & \text{sinon} \end{cases} \quad (1.44)$$

Qui est illustrée dans la figure suivante.

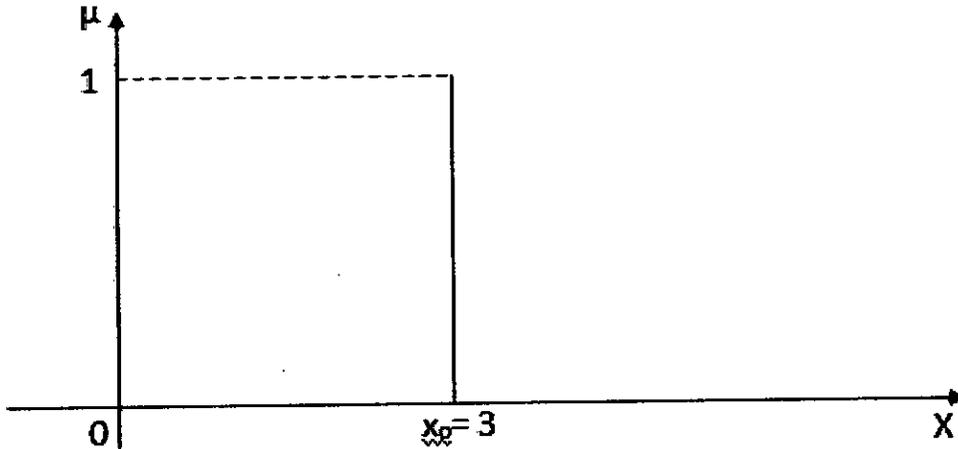


Figure 4 : Fuzzification de la valeur $x_0 = 3$

On dira qu'on a procédé à la fuzzification de la valeur réelle x_0 , d'où l'ensemble flou :

$$x_0 = 0/-6 + 0/-5 + \dots + 0/2 + 1/3 + 0/4 + 0/5 + 0/6 .$$

La composition entre l'ensemble flou x_0 obtenue par fuzzification de la valeur x_0 et de la relation R représentant la signification de la règle, induira l'ensemble flou \tilde{Y}_0 résultat de l'inférence de notre règle "If - then". On aura :

$$\tilde{Y}_0 = x_0 \circ R, \quad (" \circ " \text{ opération de composition })$$

Cela revient à prendre d'abord l'extension cylindrique de l'ensemble flou x_0 , soit $Ce(x_0)$, puis l'intersection de $Ce(x_0)$ avec la relation R , le résultat obtenu sera projeté sur le domaine V de Y , d'où :

$$\tilde{Y}_0 = \text{proj de } (Ce(\tilde{x}_0) \cap R) \text{ sur } V \quad (1.45)$$

3- La projection de $[Ce(x_0) \cap R]$ sur le domaine V :

$$\begin{aligned} \tilde{Y}_0 = & 0/-6 + 0/-5 + 0.33/-4 + 0.66/-3 + 0.66/-2 + 0.66/-1 + 0.33/0 + 0/1 \\ & + 0/2 + \dots + 0/6 . \end{aligned} \quad (1.46)$$

C'est-à-dire, la colonne pour laquelle $x = x_0 = 3$ dans la matrice qui représente la relation flou R .

Pour une autre valeur spécifique x_0 de x , une autre colonne sera sélectionnée dans cette même matrice R .

Graphiquement l'inférence de la règle de production, If X is PM then Y is NS pour une valeur spécifique x_0 de x , se résume aux étapes suivantes :

1) Déterminer le degré d'appartenance de x_0 dans le graphe de μ_{PM} , soit $\mu_{PM}(x_0)$ qui égale à 0.66 dans notre cas ,

2) constituer une version modifiée de μ_{NS} , soit :

$$\begin{aligned} \mu_{NS}(Y) & \quad \text{si} \quad \mu_{NS}(u) \leq \mu_{PM} \\ \mu_{CNS}(Y) & = \mu_{PM}(3) = 0.66 \quad \text{sinon} \end{aligned}$$

ou bien (et c'est exactement pareil) :

$$\begin{aligned} \forall Y \in U : \mu_{CNS}(Y) & = \min(\mu_{PM}(3), \mu_{NS}(Y)) \\ & = \min(2/3, \mu_{NS}(Y)) \end{aligned}$$

Cette version modifiée μ_{CNS} de μ_{NS} , représente l'ensemble flou \tilde{Y}_0 résultat de l'inférence de notre règle, (figure 5).

μ_{CNS} est appelée fonction d'appartenance écrêtée.

CNS ensemble flou écrêté.

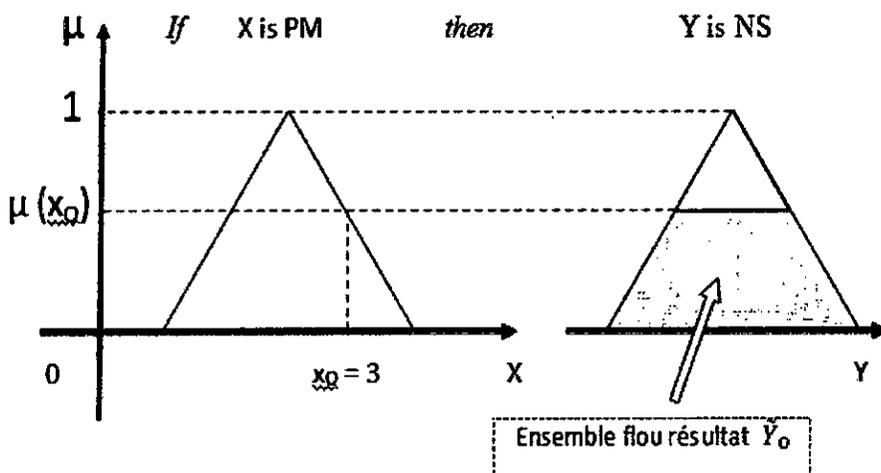


Figure 5 : Représentation graphique de l'inférence d'une simple règle " If - then "

Ainsi dans le cas où on utilise l'implication de Mamdani et pour une valeur réelle donnée x^* de X , on interprète la signification d'une règle comme suit :

$$\begin{aligned}
 \forall Y : \mu_{\text{CNS}}(Y) &= \int_U \mu^*(X) / X \quad \circ \quad \int_{U \times V} \mu_{R_m}(X, Y) / (X, Y) \\
 &= \int_V \min(\mu_{LX}(x^*), \mu_{LY}(Y)) \\
 &= \int_U \mu_{\text{CLU}}(Y),
 \end{aligned} \tag{1.47}$$

Groupe d'équations " * ".

Où " LX " et " LY " sont les valeurs linguistiques prises par X et Y et $\mu^*(X)$ est la fonction d'appartenance obtenue après fuzzification de la valeur donnée x^* ,

$$\forall X : \mu^*(X) = \begin{cases} 1 & \text{si } X = x^* \\ 0 & \text{sinon.} \end{cases} \tag{1.48}$$

Le processus décrit par les équations " * " est appelé " inférence " .

1.7.13. Représentation d'un ensemble de règles :

Considérons le cas d'un système constitué de n règles de productions "If - then" écrites symboliquement comme suit :

$$\text{If } X \text{ is } LX^{(K)} \text{ then } Y \text{ is } LY^{(K)}, \quad K = 1, \dots, n \quad (1.49)$$

Où

<p>$LX^{(K)}$ est la valeur linguistique de X dans la K^{ème} règle , La proposition " X is $LX^{(K)}$ " est interprétée comme :</p> $\tilde{L}X^{(K)} = \int_U \mu_{LX^{(K)}}(X) / X ,$ <p>$LY^{(K)}$ est la valeur linguistique de Y dans la K^{ème} règle , La proposition " X is $LY^{(K)}$ " est interprétée comme :</p> $L\tilde{Y}^{(K)} = \int_V \mu_{LY^{(K)}}(Y) / Y .$ <p>L'interprétation de Mamdani de la K^{ème} règle est alors :</p> $\forall K : \tilde{R}_m^{(K)} = \int_{U \times V} \min (\mu_{LX^{(K)}} (x^*) , \mu_{LY^{(K)}} (Y)) / (X , Y)$ <p>La fonction d'appartenance de la valeur fuzzifiée x^* est :</p> $\forall X : \mu^* (X) = \begin{cases} 1 & \text{pour } X = x^* . \\ 0 & \text{sinon .} \end{cases}$	<p>(1.50)</p>
--	---------------

Alors la signification de l'ensemble des règles est définie comme :

$$\tilde{R}_m = \bigcup_{K=1}^n \tilde{R}_m^{(K)} , \quad (1.51)$$

Ce qui veut dire :

$$\begin{aligned} \forall X, Y : \mu_{R_m}(X, Y) &= \max_K [\mu_{R_m}^{(k)}(X, Y)] \\ \mu_{R_m}(X, Y) &= \max_K \min(\mu_{LX}^{(k)}(x^*), \mu_{LY}^{(k)}(Y)) . \end{aligned} \quad (1.52)$$

Donc :

Alors l'inférence d'un ensemble de règles de productions "If - then", peut être exprimée, en accord avec le groupe d'équation " * " (voir plus haut) , par :

$$\tilde{Y} = \mu^* \circ R_m , \quad (1.53)$$

Donc :

$$\mu_y(Y) = \max_K \min(\mu_{LX}^{(k)}(x^*) , \mu_{LY}^{(k)}(Y)) . \quad (1.54)$$

Cette inférence d'un ensemble de règles par l'opération composition sera appelée " inférence basée sur la composition " .

Un deuxième type d'inférence appelée " inférence basée sur une règle individuelle " consiste à inférer chaque règle individuellement, le résultat global sera un groupe de n ensembles flous écrêtés : $C\tilde{L}U^{(1)}$, $C\tilde{L}U^{(2)}$, ..., $C\tilde{L}U^{(n)}$, un pour chaque règle avec ,

$$\forall Y \text{ et } \forall k : \mu_{C\tilde{L}U}^{(k)}(Y) = \min(\mu_{LX}^{(k)}(x^*) , \mu_{LY}^{(k)}(Y)) . \quad (1.55)$$

On prendra l'union (maximum) de tous ces ensembles flous ce qui nous donnera :

$$\mu_y(Y) = \max_K \mu_{C\tilde{L}U}^{(k)}(Y) = \max_K \min(\mu_{LX}^{(k)}(x^*) , \mu_{LY}^{(k)}(Y)) , \quad (1.56)$$

qui est le même résultat que celui trouvé pour l'inférence basée sur la composition (1.55) .Ces deux inférences représentent les types d'inférences de Mamdani.[12]

CHAPITRE 2

ETAT DE L'ART SUR LE META HEURISTIQUE DE COLONIE DE FOURMIS

2.1.Introduction

Les algorithmes de colonie de fourmis ont été introduits par Colomi et Al. (1991a). L'origine de cette approche repose sur une métaphore où le comportement des fourmis réelles lors de la quête de recherche de nourriture est reproduit. En effet, bien qu'ayant individuellement des capacités cognitives très limitées, les fourmis sont capables de sélectionner collectivement le plus court chemin pour aller de leur nid à une source de nourriture. Pour cela, elles utilisent l'environnement comme support de communication grâce au dépôt et au suivi de pistes de phéromone (une substance chimique volatile). En imitant ce comportement, une famille d'algorithmes d'optimisation a été proposée [13]. L'idée de base consiste à travailler sur une population de solutions, chacune correspondant à une fourmi en utilisant une structure de donnée commune (partagée par toutes les fourmis) qui contient des informations sur les quantités de phéromones accumulées dans l'espace des solutions pour guider leurs recherches.

2.2.Définition de colonie de fourmis

Les fourmis sont des insectes sociaux qui possèdent un certain comportement de collectivité pour se nourrir.

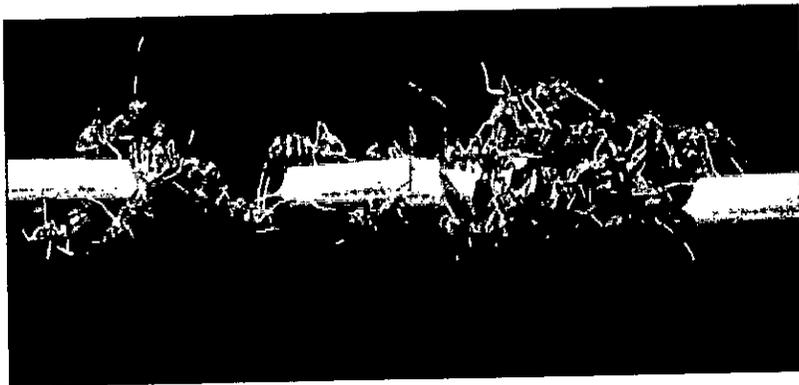


Figure 6 : Comportement des fourmis

Elles ont la particularité d'employer pour communiquer des substances volatiles appelées phéromones qui sont :

- Substances chimiques produites par des glandes, déclenchant des réactions comportementales entre les individus de la même espèce.
- signaux chimiques odorants agissant à grande distance à dose moléculaire.
- Moyen de communication chez les insectes.

Les fourmis utilisent les pistes de phéromones pour marquer leur trajet entre le nid et une source de nourriture. Une colonie est ainsi capable de choisir le plus court chemin vers une source à exploiter sans que les individus aient une vision globale du trajet [14,15].

En effet, les fourmis arrivant le plus rapidement au nid, après avoir visité la source de nourriture, sont celles qui empruntent le chemin le plus court.

Ainsi, la quantité de phéromone présente sur le court trajet est plus importante que celle présente sur le chemin le plus long.

Or une piste présentant une plus grande concentration de phéromones est plus attirante par les fourmis, elle a une probabilité plus grande d'être empruntée.

La piste courte va alors être plus renforcée que la longue et sera choisie par la grande majorité de fourmis.

Cette conduite est le résultat d'un mode de communication indirecte, via l'environnement, qu'on appelle la stigmergie

2.3. La stigmergie

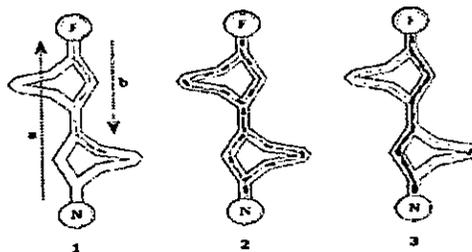


Figure 7 : Recherche du plus court chemin

1. La première fourmi trouve la source de nourriture vers un chemin quelconque (a), puis revient au nid en laissant derrière une piste de phéromone (b).
2. Les fourmis empruntent indifféremment les quatre chemins possibles, mais le renforcement sur la piste rend plus attractif le chemin le plus court.
3. Les fourmis empruntent le chemin le plus court et les partitions longues des autres chemins perdent leurs pistes de phéromones.

2.4.Optimisation par colonie de fourmis et problème de voyageurs de commerce

2.4.1.Définition du problème

Le problème du voyageur de commerce (PVC), ou *travelling salesman problem (tsp)*, est un classique du genre. Rappelons tout de même sa formulation générale

Définition : *Problème du voyageur de commerce.* Un voyageur de commerce doit visiter un ensemble $\{v_1, \dots, v_n\}$ de n villes dont on connaît les distances respectives $d(v_i, v_j), \forall (i, j) \in \{1, \dots, n\}$. Le problème consiste à trouver la permutation σ telle que la séquence $s = (v_{\sigma(1)}, \dots, v_{\sigma(n)})$ minimise la distance totale $D(\sigma)$ parcourue par le voyageur :

$$D(\sigma) = \sum_{i=1}^{n-1} d(v_{\sigma(i)}, v_{\sigma(i+1)}) + d(v_{\sigma(n)}, v_{\sigma(1)})$$

L'espace de recherche est l'ensemble des combinaisons possibles des n villes, soit au total $n!$ Combinaisons. Ce problème, Np-difficile [9], peut être aussi considéré comme la recherche d'un circuit hamiltonien de longueur minimale dans un graphe complet pouvant être antisymétrique dans le cas général $(\exists(i, j) \text{ tel que } d(v_i, v_j) \neq d(v_j, v_i))$.

Ce problème a fait l'objet de la première implémentation d'un algorithme de colonies de fourmis : le "Ant System" (AS) [16].

Le passage de la métaphore à l'algorithme est ici relativement facile et il est intéressant d'approfondir le principe de ce premier algorithme pour bien comprendre le mode de fonctionnement des algorithmes de colonies de fourmis.

Il y a deux façons d'aborder ces algorithmes. La première, la plus évidente qui est celle qui a historiquement mené au "Ant System" original ; dont on va la décrire dans cette section. La seconde est une description plus formelle des mécanismes communs aux algorithmes de colonies de fourmis, elle sera décrite dans les prochaines sections.

2.4.2.Principe général : l'algorithme Ant system

Du côté algorithmique, quelques modifications sont apportées aux capacités des fourmis décrites précédemment :

- Elles possèdent une mémoire.
- Elles ne sont pas totalement aveugles.
- Le temps est discret.

Coloni a introduits trois algorithmes [17], qui mettent à profit ce comportement collectif. Ils les ont appliqué au problème de voyageur de commerce(PVC).

De ces trois algorithmes, on retiendra celui qui a donné naissance à l'algorithme Ant System [18].

Dans cet algorithme, les fourmis sont placées sur les sommets d'un graphe (chaque sommet représentant une ville). Elles se déplacent d'un sommet à l'autre en empruntant les arêtes du graphe. On note par $b_i(t)$ i le nombre de fourmis dans la ville i à l'instant t et soit

$$m = \sum_{i=1}^n b_i(t) , \text{ le nombre total de fourmis .} \quad (2.1)$$

Chaque fourmi possède les caractéristiques suivantes :

- La fourmi dépose une trace de phéromones sur l'arête (i, j) quand elle se déplace de la ville i à la ville j .
- Elle choisit la ville de destination suivant une probabilité qui dépend de la distance entre cette ville, sa position et la quantité de phéromones présente sur l'arête (règle de transition).

– Pour ne pas passer qu'une seule fois par chaque ville, la fourmi ne peut se rendre sur une ville qu'elle a déjà traversée, c'est pour cela que la fourmi doit être dotée d'une mémoire.

Pour éviter qu'une fourmi ne revienne sur ses traces, elle conserve la liste des villes qu'elle a déjà traversées. Cette liste, nommée liste-tabou est remise à zéro chaque fois que la fourmi a terminé un tour. La liste-tabou constitue la mémoire de la fourmi.

Les traces de phéromones sont modélisées par les variables $\tau_{ij}(t)$ qui donnent l'intensité de la trace sur le chemin (i, j) à l'instant t . La probabilité de transition du sommet i vers le sommet j par la fourmi k est donnée par :

$$p_{ij} = \begin{cases} \frac{(\tau_{ij}(t))^\alpha \cdot (v_{ij})^\beta}{\sum_{j \in L_k(t)} (\tau_{ij}(t))^\alpha \cdot (v_{ij})^\beta} & \text{si } j \notin L_k(t) \\ 0 & \text{sinon} \end{cases} \quad (2.2)$$

Où :

- $L_k(i)$ représente la liste- tabou de la fourmi k située sur le sommet i .
- $v_{ij} = 1/d_{ij}$ représente une mesure de la *visibilité*, cette information statique qui correspond à l'inverse de la distance entre les villes i et j , est utilisée pour diriger le choix des fourmis vers les villes les plus proches et éviter les villes trop lointaines.
- α et β sont deux paramètres contrôlant l'importance relative des phéromones.
- $\tau_{ij}(t)$ (intensité de la piste) et de la visibilité v_{ij} .

Après un tour complet, chaque fourmi laisse une certaine quantité de phéromones $\Delta\tau_{ij}^k(t)$ sur l'ensemble de son parcours, la quantité qui dépend de la qualité de la solution trouvée :

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L^k} & \text{si la fourmi } k \text{ est passée par l'arc } (i,j) \\ 0 & \text{sinon} \end{cases} \quad (2.3)$$

Où Q une constante et L^k la longueur du chemin parcouru par la fourmi k .

L'algorithme ne serait pas complet sans le processus d'évaporation des pistes de phéromone. En effet, pour éviter d'être piégé dans des solutions sous optimales, il est nécessaire de permettre au système "d'oublier" les mauvaises solutions. La mise à jour des phéromones est effectuée une fois que toutes les fourmis sont passées par toutes les villes.

Cette mise à jour est donnée par la formule suivante :

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \sum_{k=1}^m \Delta \tau_{ij}^k(t) \quad (2.4)$$

Où ρ est un coefficient représentant l'évaporation des traces de phéromones.

$\Delta \tau_{ij}^k$ Représente le renforcement de l'arc (i, j) pour la fourmi k .

L'algorithme suivant donne la structure générale de AS pour le PVC (noté Ant System-TSP).

Algorithme de colonies de fourmis de base : le "Ant System-TSP" :

Pour $i = 1, \dots, i_{max}$

Pour chaque fourmi $k = 1, \dots, m$

Choisir une ville au hasard

Pour chaque ville non visitée i

Choisir une ville j , dans la liste des villes restantes, selon la formule (1)

Fin Pour

Déposer une piste $\Delta \tau_{ij}^k(t)$ sur le trajet conformément à l'équation (2)

Fin Pour

Evaporer les pistes des phéromones selon la formule (3)

Fin Pour

La valeur initiale de τ_{ij} est τ_0 .

Concernant le nombre de fourmis, il est raisonnablement proposé d'utiliser autant de fourmis que de villes ($m = n$).

La suite de cette section présente un certain nombre d'extensions proposées autour de Ant System.

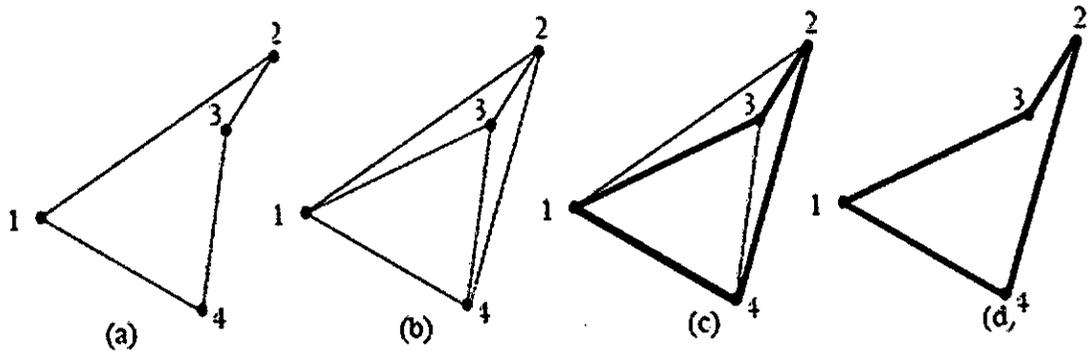


Figure 8 : Le problème du voyageur de commerce optimisé par l'algorithme AS

- Les points représentent les villes.
- L'épaisseur des arêtes représente la quantité de la phéromone déposée.
- (a) Exemple du trajet construit par une fourmi.
- (b) Au début du calcul, tous les chemins sont explorés.
- (c) Le chemin le plus court est le plus renforcé que les autres.
- (d) L'évaporation permet d'éliminer les moins bonnes solutions.

2.4.3. Variantes

2.4.3.1. Ant System & élitisme :

Une première variante du "Ant System" a été proposée dans [18], elle est caractérisée par l'introduction de fourmis " élitistes". Dans cette version, la meilleure fourmi (celle qui a effectué le trajet le plus court) dépose une quantité de phéromone plus grande, dans l'optique d'accroître la probabilité des autres fourmis d'explorer la solution la plus prometteuse.

2.4.3.2. "Ant-Q" :

"Ant-Q" ou Ant System avec apprentissage par renforcement. Dans cette variante de l'Ant System, la règle de mise à jour locale est inspirée de l'algorithme d'apprentissage par renforcement "Q-learning" [19]. Cependant, aucune amélioration par rapport à l'algorithme Ant System n'a pu être démontrée. Cet algorithme n'est d'ailleurs, qu'une préversion du "Ant Colony System".

2.4.3.3.ACS (ant colony system) :

"Ant Colony System" (ACS) a été introduit pour améliorer les performances du premier algorithme sur des problèmes de grandes tailles [6] ACS est fondé sur des modifications de l'Ant System.

– ACS introduit une règle de transition dépendant d'un paramètre q_0 ($0 \leq q_0 \leq 1$), qui définit une balance diversification/intensification.

Une fourmi k placée dans une ville i choisira une ville j par la règle :

$$j = \begin{cases} \arg \max_{j \in j_k(i)} \{ [\tau_{ij}(t)] [v_{ij}]^\beta \} & \text{si } q \leq q_0 \\ J & \text{sinon} \end{cases} \quad (2.5)$$

Où β est un paramètre servant à moduler la prise en compte des phéromones par rapport à la visibilité.

– Des listes de villes candidates sont utilisées pour accélérer le processus de construction d'un chemin.

– Une heuristique locale est utilisée pour améliorer les solutions générées par les fourmis.

– La mise à jour des phéromones n'est faite qu'à partir du meilleur chemin généré.

– une règle de mise à jour locale des phéromones est utilisée à chaque transition d'une fourmi.

Les résultats obtenus par ACS sur le PVC sont les meilleurs obtenus par les heuristiques à base de fourmis sans toutefois dépasser les meilleures heuristiques dédiées à ce problème.

2.4.3.4.Max Min Ant System :

Cette variante (notée MMAS) est fondée sur l'algorithme Ant System ,elle présente quelques différences notables [20] :

– Seule la meilleure fourmi met à jour une piste de phéromone.

– Les valeurs des pistes sont bornées par τ_{min} et τ_{max} .

– Les pistes sont initialisées à la valeur maximum max .

– La mise à jour des pistes est faite de façon proportionnelle, les pistes les plus fortes étant moins renforcées que les plus faibles.

– Une réinitialisation des pistes peut être effectuée.

Les meilleurs résultats sont obtenus en mettant à jour la meilleure solution avec une fréquence de plus en plus forte au cours de l'exécution de l'algorithme.

L'heuristique ACO :

Toutes les variantes que nous venons d'exposer ont été regroupées sous l'acronyme ACO (Ant Colony Optimization), pour une description plus large afin de faciliter le rapprochement des méthodes entre elles et de se soustraire aux spécificités du PVC [21,22].

2.4.4.Choix des paramètres

Pour l'algorithme AS, les auteurs préconisent que, bien que la valeur de q ait peu d'influence sur le résultat final, cette valeur soit du même ordre de grandeur qu'une estimation de la longueur du meilleur trajet trouvé. D'autre part, la ville de départ de chaque fourmi est typiquement choisie par un tirage aléatoire uniforme, aucune influence significative du placement de départ n'ayant pu être démontrée.

En ce qui concerne l'algorithme ACS, les auteurs conseillent d'utiliser

$$\tau_0 = (n \cdot L_{nn})^{-1} .$$

Où : n est le nombre de villes.

L_{nn} la longueur d'un tour trouvé par la méthode du plus proche voisin.

Le nombre de fourmis m est un paramètre important ; les auteurs suggèrent d'utiliser autant de fourmis que de villes (i.e. $m = n$) pour de bonnes performances sur le problème du voyageur de commerce. Il est possible de n'utiliser qu'une seule fourmi, mais l'effet d'amplification des longueurs différentes est alors perdu, de même que le parallélisme naturel de l'algorithme.

En règle générale, les algorithmes de colonies de fourmis semblent peu sensibles à un réglage fin du nombre de fourmis.

2.5. Autres problèmes combinatoires

Les algorithmes de colonies de fourmis sont très étudiés depuis quelques années et il serait trop long de faire ici une liste exhaustive de toutes les applications et variantes qui ont été produites, même en se restreignant au domaine de l'optimisation. Dans les deux principaux champs d'application (problèmes N_p -difficiles et problèmes dynamiques), certains algorithmes ont cependant donné de très bons résultats. On peut notamment retenir des performances particulièrement intéressantes dans le cas de l'affectation quadratique [23, 24,20], de problèmes de planification [25], de coloriage de graphes [26], d'affectation de fréquence du [27], ou du routage sur réseau [21]. Il existe une littérature importante sur toutes sortes de problèmes : voyageur de commerce, ordonnancement séquentiel, routage de véhicule, affectation généralisé, sac à dos multidimensionnel, satisfaction de contraintes, etc.

2.6. Formalisation et propriétés d'un algorithme de colonie de fourmis

Une description élégante a été proposée [25], qui s'applique aux problèmes (combinatoires) ou une construction partielle de la solution est possible.

Ce cas, bien que restrictif, permet de dégager les apports originaux des méta heuristiques ACO, nous en avons traduit ci-dessous un extrait :

Une métaheuristique de colonie de fourmis est un processus stochastique construisant une solution, en ajoutant des composants aux solutions partielles. Ce processus prend en compte :

- Une heuristique sur l'instance du problème .
- Des pistes de phéromone changeant dynamiquement pour refléter l'expérience acquise par les agents.

Une formalisation plus précise existe [25]. Elle passe par une représentation du problème, un comportement de base des fourmis et une organisation générale du métaheuristique.

Plusieurs concepts sont également à mettre en valeur pour comprendre les principes de ces algorithmes, notamment la définition des pistes de phéromone en tant que mémoire adaptative, la nécessité d'un réglage intensification/diversification et enfin l'utilisation d'une recherche locale.

2.6.1. Formalisation

Représentation du problème :

Le problème est représenté par un jeu de solutions, une fonction objective assignant une valeur à chaque solution et un jeu de contraintes. L'objectif est de trouver l'optimum global de la fonction objectif satisfaisant les contraintes.

Les différents états du problème sont caractérisés comme une séquence de composants. On peut noter que, dans certains cas, un coût peut être associé à des états autres que des solutions.

En théorie des graphes, un graphe simple est noté par $G(X,U)$ où X est l'ensemble des sommets et U est l'ensemble des arêtes reliant éventuellement des couples de sommets.

Dans cette représentation, les fourmis construisent des solutions en se déplaçant sur un graphe $G = (C, L)$, où les nœuds sont les composants de C où l'ensemble L connecte les composants de C .

Les contraintes du problème sont implémentées directement dans les règles de déplacement des fourmis (soit en empêchant les mouvements qui provoquent les contraintes, soit en pénalisant certaines solutions).

Comportement des fourmis :

Les fourmis artificielles peuvent être caractérisées comme une procédure de construction stochastique construisant des solutions sur le graphe G tel que :

$$G = (C; L).$$

En général, les fourmis tentent d'élaborer des solutions faisables mais, si nécessaire, elles peuvent produire des solutions infaisables.

Les composants et les connexions peuvent être associés à des pistes de phéromones (mettant en place une mémoire adaptative décrivant l'état du système) et à une valeur heuristique (représentant une information a priori sur le problème, ou venant d'une source autre que celle des fourmis ; c'est bien souvent le coût de l'état en cours).

Les pistes de phéromone et la valeur de l'heuristique peuvent être associées soit aux composants, soit aux connexions.

Chaque fourmi dispose d'une mémoire utilisée pour stocker le trajet effectué, d'un état initial et de conditions d'arrêt.

Les fourmis se déplacent d'après une règle de décision probabiliste en fonction des pistes de phéromone locales, de l'état de la fourmi et des contraintes du problème.

Lors de l'ajout d'un composant à la solution en cours, les fourmis peuvent mettre à jour la piste associée au composant ou à la connexion correspondante.

Une fois la solution construite, elles peuvent mettre à jour la piste de phéromone des composants ou des connexions utilisées. Enfin, une fourmi dispose au minimum de la capacité de construire une solution du problème.

Organisation du métaheuristique :

En plus des règles régissant le comportement des fourmis, un autre processus majeur qui est en cours tel que l'évaporation des pistes de phéromone.

En effet, à chaque itération, la valeur des pistes de phéromone est diminuée. Le but de cette diminution est d'éviter une convergence trop rapide et le piégeage de l'algorithme dans des minimums locaux, par une forme d'oubli favorisant l'exploration de nouvelles régions.

Selon les auteurs du formalisme ACO, il est possible d'implémenter d'autres processus nécessitant un contrôle centralisé (et donc ne pouvant être directement pris en charge par des fourmis), sous la forme de processus annexes.

En effet, on perd alors la caractéristique décentralisée du système. De plus, l'implémentation de processus annexes est difficilement réalisable dans une formalisation rigoureuse.

2.6.2. Phéromones et mémoire

L'utilisation de la stigmergie est cruciale pour les algorithmes de colonies de fourmis. Le choix de la méthode d'implémentation des pistes de phéromone est donc important pour obtenir les meilleurs résultats. Ce choix est en grande partie lié aux possibilités de représentation de l'espace de recherche. Chaque représentation pouvant apporter une façon différente d'implémenter les pistes, par exemple, pour le problème du voyageur de commerce, une implémentation efficace consiste à utiliser une piste τ_{ij} entre deux villes i et j comme une représentation de l'intérêt de visiter la ville j après la ville i .

Une autre représentation possible, moins efficace en pratique, consiste à considérer τ_{ij} comme une représentation de l'intérêt de visiter i en tant que j éme ville.

En effet, les pistes de phéromone décrivent à chaque pas l'état de la recherche de la solution par le système. Les agents modifient la façon de représentation du problème. Cette information est partagée par le biais des modifications de l'environnement des fourmis, grâce à une forme de communication indirecte qui est la stigmergie.

L'information est donc stockée un certain temps dans le système, ce qui amène certains auteurs à considérer ce processus comme une forme de mémoire adaptative [28], où la dynamique de stockage et le partage de l'information va être cruciale pour le système.

2.6.3. Intensification/diversification

Le problème de l'emploi relatif de processus de diversification et d'intensification est un problème extrêmement courant dans la conception et l'utilisation de métaheuristiques. Par intensification, on entend l'exploitation de l'information accumulée par le système à un moment donné. La diversification est

au contraire l'exploration de régions de l'espace de recherche imparfaitement prises en compte. Bien souvent, il va s'agir de choisir où et quand injecter de l'aléatoire dans le système (diversification). Et/ou améliorer une solution (intensification).

Dans les algorithmes de type ACO, comme dans la plupart des cas, il existe plusieurs façons de gérer l'emploi de ces deux facettes des métaheuristiques d'optimisation. La plus évidente passe par le réglage via les deux paramètres α et β , qui déterminent l'influence relative des pistes de phéromone et de l'information heuristique. Plus la valeur de α sera élevée, plus l'intensification sera importante, car plus les pistes auront une influence sur le choix des fourmis. A l'inverse, plus α sera faible, plus la diversification sera forte, car les fourmis éviteront les pistes. Le paramètre β agit de façon similaire. On doit donc gérer conjointement les deux paramètres, pour régler ces aspects.

On peut également introduire des modifications de la gestion des pistes de phéromone, par exemple l'emploi de stratégies élitistes (les meilleures solutions contribuent plus aux pistes, voir : l'algorithme AS avec élitisme) favorise l'intensification, alors qu'une réinitialisation de l'ensemble des pistes favorisera l'exploration (l'algorithme MMAS).

Ce choix diversification/intensification peut s'effectuer de manière statique avant le lancement de l'algorithme, en utilisant une connaissance a priori du problème, ou de manière dynamique, en laissant le système décider du meilleur réglage deux approches sont possibles : un réglage par les paramètres ou l'introduction de nouveaux processus.

Ces algorithmes montrent l'utilisation de l'auto-organisation, car ces deux approches peuvent être équivalentes. Un changement de paramètre peut induire un comportement complètement différent du système, au niveau global.

2.7. Conclusion

Le mode d'opération de base des algorithmes ACO est comme suit [1] :

Dans chaque itération, une population d'un nombre spécifique de fourmis construit progressivement différentes voies sur un graphe qui représente une solution au problème. Cette solution dépend d'une règle probabiliste de transition qui enchaîne l'information heuristique et les traînées de phéromones.

Ensuite, les traînées de phéromone sont mises à jour par la règle de mise à jour globale, qui se base sur la diminution et le renforcement de phéromones sur les attributs des solutions construites.

La diminution est considérée par un certain facteur constant qui correspond à l'évaporation de phéromones.

Le renforcement est considéré selon la qualité des attributs des solutions construites.

On a proposé plusieurs extensions à ce mode d'opération basique .Leurs principales améliorations consistent à l'utilisation des différentes transitions, de règles de mises à jour, et de l'introduction de nouveaux composants ou par l'ajout de phase de recherche locale [2].

L'approche BWAS [2], est une approche qui améliore l'exécution des modèles ACO ,en utilisant des concepts d'algorithmes évolutionnaires ,comme la règle de mise à jour basée sur l'apprentissage incrémentale [29] (par considération des globales meilleures et plus mauvaises solutions courantes) ou par la mutation de la traînée de phéromone pour introduire la diversité dans la recherche ,et cela est détaillé dans le chapitre suivant .

CHAPITRE 3

Dérivation de la base des connaissances d'un système flou par l'approche hybride (COR -BWAS)

3.1.Introduction

Dans les systèmes basés sur les règles floues (FRBS), la tâche de la définition automatique de la base des connaissances (base des règles et base de données) pour une application concrète est considérée comme un problème difficile.

Plusieurs méthodes ont été proposées dans la littérature pour résoudre ce type de problème. On citera les méthodes Ad Hoc Data Driven [1], les réseaux neurologiques [2], les algorithmes génétiques [2,4] .. Etc.

Les méthodes Ad Hoc Data Driven sont appropriées aux tâches de dérivation des bases de connaissances des systèmes flous. Elles sont connues pour leur efficacité et leur simplicité [3]. Cependant, ces méthodes recherchent habituellement, les règles floues possédant les meilleures performances individuelles, et **l'interaction globale** entre les règles floues de la base n'est pas prise en considération. Ceci peut engendrer des bases de règles floues ayant une **mauvaise coopération** entre les règles qui les composent, entraînant la diminution de la **précision** du système.

Pour remédier aux inconvénients des méthodes Ad Hoc Data Driven, on présente dans ce travail une méthodologie des règles coopératives (COR)(Cooperative rules methodology) permettant d'améliorer la précision du système, avec une meilleure coopération entre les règles.

La méthodologie COR considère la possibilité d'utiliser un conséquent différent du meilleur, car celui-ci permet au système flou, d'être **plus précis (Plus grande exactitude)** en possédant un ensemble de règles floues avec une meilleure coopération, au lieu de sélectionner le conséquent des règles avec la meilleure performance dans chaque sous espace comme dans les méthodes Ad Hoc Data Driven.

Les conditions de mise en œuvre des systèmes flous FRBS sont citées comme suit [3] :

- **Interprétabilité** : elle se rapporte aux possibilités du système flou à exprimer le comportement du système d'une manière compréhensible. C'est une propriété subjective qui dépend du nombre de règles floues à réduire.
- **Exactitude** : elle se rapporte aux possibilités que possède le système flou pour représenter loyalement le système réel.

La vitesse du processus de dérivation des règles floues possède également une importance vitale. En effet un apprentissage rapide possède plusieurs avantages intéressants, telle que la possibilité d'être employé comme un mécanisme antérieur (au préalable) , afin de comprendre les caractéristiques du problème à résoudre [3], et cela aboutit à une première étape d'apprentissage pour raffiner les résultats obtenus avec un post-traitement plus complexe, cela peut aussi être intégré dans un processus de méta-apprentissage ...etc.

Dans cette contribution, nous présentons une nouvelle approche pour le problème de dérivation de la base des connaissances d'un contrôleur flou ,basé sur la méthodologie des règles coopératives **COR** , et on se servira des algorithmes d'optimisation par colonie de fourmis (**ACO**) pour trouver un bon équilibre entre les trois conditions mentionnées au paravent à savoir **l'interprétabilité** , **l'exactitude des systèmes flous obtenus** et **la rapidité de l'apprentissage** .

ACO, comme vu au chapitre 2 est un nouveau paradigme des algorithmes bio-inspirés qui a montré un excellent comportement dans la résolution des problèmes d'optimisations combinatoires (problème de voyageur de commerce PVC).

L'avantage principal de cette technique, est la recherche globale guidé par l'information heuristique et mémoristique avec une manière très efficace.

Pour appliquer les algorithmes ACO à notre problème de dérivation de la base des connaissances , on doit formuler le problème comme un problème d'optimisation combinatoire et introduire les caractéristiques liées à ces algorithmes, tels que l'information heuristique, l'initialisation de phéromone , la fonction d'adéquation: (fitness) , la construction d'une solution et la mise à jour de la traînée de phéromone .

La méthodologie des règles coopératives COR accomplit la tâche d'apprentissage en deux étapes :

- La construction de l'espace de recherche.
 - La sélection de l'ensemble des règles floues les plus coopératives.
- Ce travail sera guidé par une mesure de l'erreur globale (**MSE**).

Puisque l'information heuristique sur la qualité locale du conséquent de chaque règle est disponible, alors les algorithmes ACO s'adaptent correctement à cette méthodologie pour exécuter une meilleure recherche et augmenter la vitesse de convergence, produisant ainsi les systèmes flous interprétables, exactes et avec une grande rapidité.

L'algorithme du meilleur- mauvais système de fournis (**BWAS : Best-Worst-Ant-System**) [3], sera utilisé dans ce travail pour sa bonne efficacité dans la résolution de plusieurs problèmes d'optimisation dans la littérature [1] [2].

Par la suite on décrira la méthodologie de règles coopératives **COR**, avant de présenter très brièvement les algorithmes **ACO** et **BWAS**.

3.2.Méthodologie des règles coopératives COR

La méthodologie COR se divise en deux étapes :

- Construction de l'espace de recherche : où un ensemble de règles candidates est obtenu pour chaque sous espace d'entrée floue.
- Sélection de l'ensemble des règles floues les plus coopératives : où une recherche combinatoire est exécutée parmi ces ensembles recherchant des règles avec la meilleure précision globale [31,32].

Cette approche permet la réduction de l'ensemble de règles floues ,qui est une pratique régulière dans le domaine des systèmes basés sur les règles floues habituellement réalisé par les algorithmes génétiques. [33, 34,35].

Pour faire ainsi, l'élément spécial R_0 (qui signifie – don't care – Néant-pas de règle) est ajouté à l'ensemble des règles candidates correspondantes à chaque sous espace. De cette manière, si un tel élément est sélectionné pour un sous espace spécifique, cela signifiera qu'aucune règle appartenant à ce sous espace ne participera à l'ensemble des règles floues final après apprentissage .Ce

léger changement dans la méthodologie COR implique évidemment l'augmentation de l'espace de recherche, avec le " pour connu " et le "contre".

Des solutions plus précises et plus interprétables peuvent être obtenues mais la difficulté pour trouver de bonnes solutions augmente.

Une description plus large du procédé de génération des règles basé sur la méthode COR incluant le mécanisme de réduction de l'ensemble des règles floues est montrée sur la figure suivante :

Inputs :

- Un ensemble de données entrées – sortie - $E = \{e_1, \dots, e_l, \dots; e_N\}$ avec $e_l = \{x_1^l, \dots, x_n^l, y^l\}$, $l \in \{1, \dots, N\}$, N étant la taille de l'ensemble des données, et n le nombre de variables d'entrée – représentant le comportement du problème à résoudre.
- Une partition floue de l'espace des variables. Dans notre cas, des ensembles flous uniformément distribués sont considérés. Soit A_i l'ensemble des termes linguistiques de la i -ème variable d'entrées avec $i \in \{1, \dots, n\}$ et B l'ensemble des termes linguistiques de la variable de sortie, avec $|A_i|(|B|)$ le nombre de labels de la i -ème variable d'entrée (sortie).

Algorithme :

Construction de l'espace de recherche :

Définissez les sous-espaces d'entrée flous contenant des exemples positifs : pour faire ainsi, nous devrions définir l'ensemble d'exemples positifs ($E^+(S_s)$) pour chaque sous-espace d'entrée flou $S_s = (A_1^s, \dots, A_2^s, \dots, A_n^s)$, avec $A_i^s \in A_i$ un label, $s \in \{1, \dots, N_s\}$ et $N_s = \prod_{i=1}^n |A_i|$ le nombre des sous-espaces d'entrées flous. Deux possibilités se présentent :

$$E^+(S_s) = \{e_l \in E / \forall i \in \{1, \dots, n\}, \forall A_i^s \in A_i, \mu_{A_i^s}(x_i^l) \geq \mu_{A_i^s}(x_i^l)\}.$$

$E^+(S_s) = \{e_l \in E / \mu_{A_1^s}(x_1^l) \dots \mu_{A_n^s}(x_n^l) \neq 0\}$ produit des FA avec $\mu_{A_i^s}(\cdot)$ La fonction d'appartenance associée au label A_i^s .

Parmi tous les sous –espaces flous d'entrées possibles N_s , on considère seulement ceux contenant au moins un exemple positif. Ainsi, l'ensemble des sous-espaces avec des exemples positifs est défini comme :

$$S^+ = \{S_j \mid E^+(S_j) \neq \emptyset\}.$$

Gérez l'ensemble de règles candidates dans chaque sous-espace avec exemples positifs :

L'ensemble des conséquents candidats associé à chaque sous espace contenant au moins un exemple, $S_j \in S^+$, est défini. Il y a encore deux possibilités :

$$C(S_j) = \{B_k \in B \mid \exists e_1 \in E^+(S_j) \text{ avec } \forall B' \in B, \mu_{B_k}(Y^1) \geq \mu_{B'}(Y^1)\}.$$

$$C(S_j) = \{B_k \in B \mid \exists e_1 \in E^+(S_j) \text{ avec } \mu_{B_k}(y^1) \neq 0\}.$$

Alors l'ensemble des règles candidates pour chaque sous-espace est défini comme :

$$CR(S_j) = \{R_k = [\text{IF } X_1 \text{ is } A_1^1 \text{ and ... and } X_n \text{ is } A_n^1 \text{ THEN } Y \text{ is } B_k] \mid B_k \in C(S_j)\}.$$

Pour permettre à la méthodologie COR de réduire le nombre initial des règles floues l'élément spécial R_o (qui signifie (don't care)) est ajouté à chaque ensemble de règles candidates , c'est-à-dire :

$$CR(S_j) = CR(S_j) \cup R_o.$$

Si cet élément est sélectionné, aucune règle n'est employée dans le sous-espace d'entrée flou correspondant.

Sélection de l'ensemble de règles floues le plus coopératif :

Cette étape est exécutée à l'aide d'un algorithme de recherche combinatoire le but étant de trouver la combinaison des règles avec la meilleure exactitude

$$\{R_1 \in CR(S_1), \dots, R_j \in CR(S_j), \dots, R_{|S^+|} \in CR(S_{|S^+|})\}.$$

Un index mesurant le degré de coopération de l'ensemble des règles codées est considéré pour l'évaluation de la qualité de chaque solution.

Dans notre cas, l'algorithme emploie une fonction erreur globale appelée l'erreur de moyenne carrée (MSE), qui est définie comme :

$$MSE = (1/2N) \cdot \sum_{j=1}^N (F(x_1^j, \dots, x_n^j) - y^j)^2, \quad \sum = \text{de } 1 \text{ à } N, \text{ avec :}$$

- (x_1^j, \dots, x_n^j) la sortie obtenue à partir du FRBS (système basé sur les règles floues) lorsque l'exemple e_j est utilisé,
- y^j la sortie désirée connue.

Plus la mesure se rapproche de zéro, plus la performance globale est grande, plus la coopération des règles est meilleure.

Figure 9 : l'Algorithme COR

3.3. Introduction à l'optimisation de colonie de fourmis ACO

Les algorithmes ACO[36], constituent une nouvelle famille d'algorithmes bio-inspirés qui a montré un excellent comportement dans la résolution des problèmes d'optimisation combinatoire. La première proposition d'algorithme de système de fourmis a été appliqué au problème du voyageur de commerce (PVC)[37], de nombreux modèles ont été développés par la suite pour résoudre un ensemble large de problème d'optimisation [38, 39, 40].

Pour résoudre un problème spécifique par un algorithme ACO, on doit suivre les étapes suivantes [41] :

- Représentation de problème : interprétez le problème à résoudre comme un graphe ou une structure similaire facilement parcourable par des fourmis.
- L'information heuristique : définissez la manière d'assigner une préférence heuristique à chaque choix que la fourmi doit faire à chaque étape pour générer la solution.
- Initialisation de phéromone : établissez une manière appropriée d'initialisation de phéromone.
- Fonction d'adéquation(fitness) : définissez une fonction d'adéquation (fitness) à optimiser.
- Schéma d'ACO : choisissez un algorithme ACO et appliquez le au problème.

3.4. Le meilleur mauvais système de fourmis (BWAS)

L'approche BWAS est réalisée en suivant les étapes suivantes :

- Donner une valeur initiale de phéromone.
- Tant que la condition d'arrêt n'est pas satisfaite faites :
 - a. Générez le chemin de chaque fourmi par le procédé de construction de solution.
 - b. Appliquez le mécanisme d'évaporation de phéromone.
 - c. Appliquez le processus de recherche locale sur la meilleure solution courante.
 - d. Mettez à jour $S_{global\ best}$ et $S_{current\ worst}$.
 - e. Appliquez la meilleure mauvaise règle de mise à jour de la trainée de phéromone
 - f. Appliquez la mutation de trainée de phéromone.
 - g. Si (la condition est satisfaite) appliquez alors le relancement

3.5.L'apprentissage du système flou par l'approche BWAS

COR est caractérisé par sa flexibilité d'être employé avec différentes Meta heuristique, néanmoins, ses résultats sont améliorés par l'incorporation de l'information heuristique au processus d'apprentissage. Cette considération guiderait l'algorithme dans la recherche, en le rendant plus efficace à trouver de bonnes solutions.

ACO est un bon support pour une telle intension (l'utilisation inhérente d'information heuristique). Par conséquent, cette section décrit l'utilisation d'ACO, par l'approche BWAS dans la méthodologie COR. Pour faire ainsi on doit suivre les cinq sous sections suivantes :

3.5.1.Représentation du problème pour l'apprentissage des règles floues coopératives

Pour appliquer ACO dans la méthodologie COR, il est commode de voir le problème comme un problème d'optimisation combinatoire avec les possibilités de le représenter sur un graphe dont les connexions possèdent des poids. De cette façon nous pouvons faire face aux problèmes en considérant un nombre fixe de sous-espaces et en interprétant le processus d'apprentissage comme une manière d'assignement des conséquents c.-à-d., labels de la partition floue de sortie – à ces sous-espaces en respectant un critère d'optimalité (c.-à-d., suivant la méthodologie de COR).

Le problème sera représenté d'une manière similaire au problème QAP (problème d'affectation quadratique) [43] avec quelques particularités.

Analogie :

Locations \longrightarrow sous-espaces et équipement \longrightarrow conséquents.

- Contrairement à QAP, dans notre cas, l'ensemble des conséquents possibles pour chaque règle floue peut être différent et on peut assigner un même conséquent à plusieurs règles.
- Selon l'algorithme COR, chaque nœud $S_j \in S^+$ est assigné à chaque conséquent candidat $B_K \in C(S_j)$ et au symbole spécial "don't care "

La figure 10 montre un exemple de processus d'apprentissage.

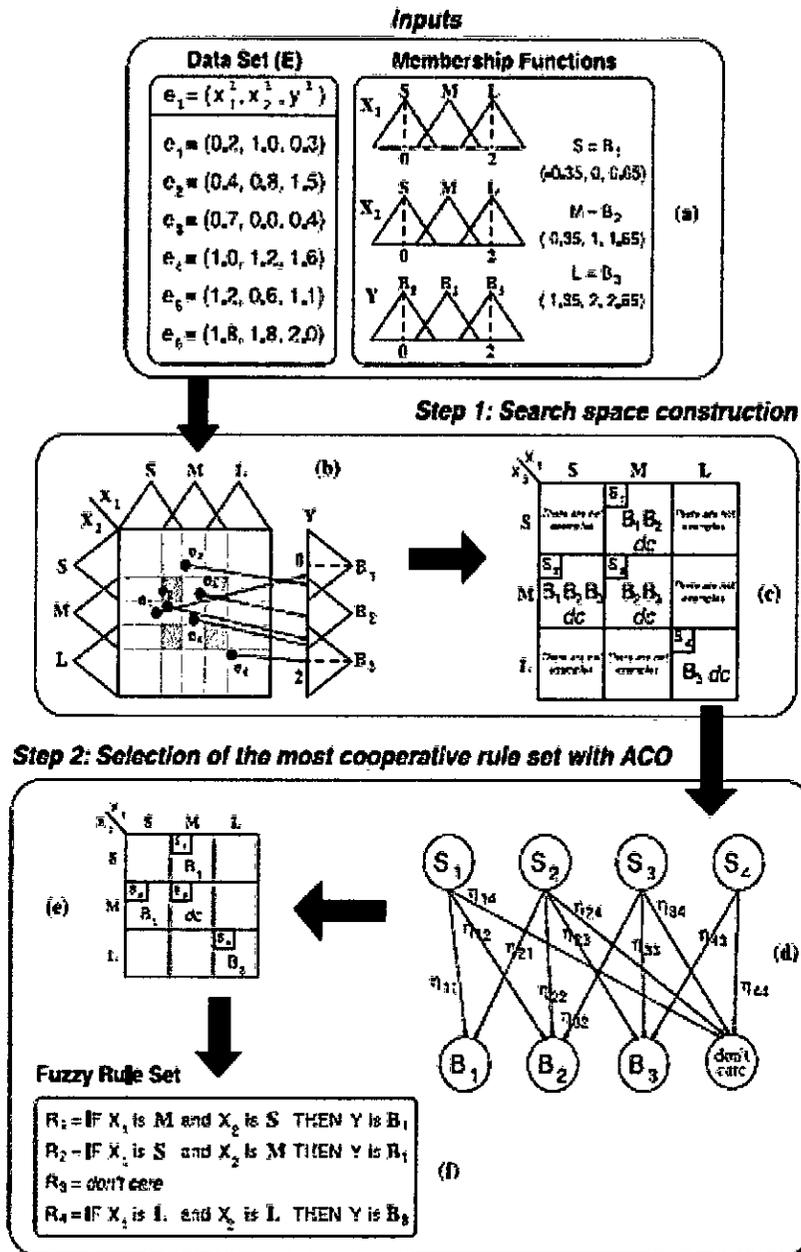


Figure 10 : Exemple de processus d'apprentissage.

Sur la figure 10 (c) les conséquents possibles pour chaque combinaison d'antécédents sont montrés selon l'ensemble de données et les fonctions d'appartenance considérées (figure 10 (a)).

Et pour construire une solution complète, une fourmi va itérativement au-dessus de chaque règle et choisit un conséquent avec une probabilité qui dépend

de la traînée de phéromone et de l'information heuristique associés à chaque décision, (voir la figure 10(d)).

L'ordre de la sélection des règles est non pertinent.

Les figures 10 (e) et 10 (f) montrent l'ensemble des règles floues codées par une solution spécifique.

3.5.2. Information heuristique

L'information heuristique sur la préférence potentielle de la sélection d'un conséquent spécifique B_K dans chaque combinaison d'antécédent (sous-espace) est déterminée, comme décrit sur la figure 11.

Pour chaque sous espace $S_j \in S^+$ faire :

Construire les ensembles $E^+(S_j)$ et $C(S_j)$ comme montré dans la figure 9.

Pour chaque $B_K \in C(S_j)$, utiliser une fonction d'initialisation basée sur un critère de recouvrement pour avoir le degré de préférence heuristique pour chaque choix.

Plusieurs possibilités différentes peuvent être considérées. Dans ce qui suit trois de ces possibilités :

$$\eta_{jk} = H_1(S_j, B_K) = \max_{e_l \in E^+(S_j)} \text{Min} (\mu_{A_j}(x^l), \mu_{B_K}(y^l))$$

$$\eta_{jk} = H_2(S_j, B_K) = (1/|E^+(S_j)|) \sum_{e_l \in E^+(S_j)} \text{Min} (\mu_{A_j}(x^l), \mu_{B_K}(y^l))$$

$$\eta_{jk} = H_3(S_j, B_K) = H_1(S_j, B_K) \cdot H_2(S_j, B_K)$$

$$\text{Avec : } \mu_{A_j}(x^l) = \text{Min} (\mu_{A_{1j}}(x_1^l), \dots, \mu_{A_{nj}}(x_n^l))$$

Pour chaque B_K n'appartenant pas à $C(S_j)$ faire $\eta_{jk} = 0$.

Finalement, pour le symbole « dc », faire ce qui suit :

$$\eta_{j,|B|+1} = \frac{1}{C(S_j)} \cdot \sum_{B_K \in C(S_j)} \eta_{jk}$$

Figure 11 : Processus d'assignement d'heuristique

L'information heuristique est réalisée par chaque combinaison antécédente (en considérant les critères qui sont interprétés graphiquement dans la figure suivante:

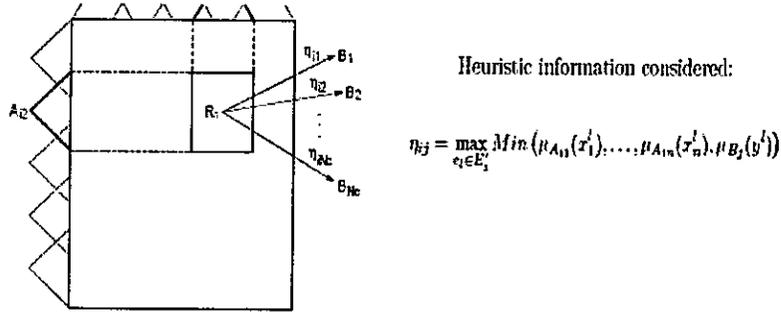


Figure 12 : Assignment heuristique de la règle R_i pour chaque conséquence du système avec deux variables d'entrées et trois conséquences (N_c) de la sortie floue.

La figure13 montre un processus d'apprentissage pour un problème simple avec deux variables d'entrées et quatre règles ($N_r = 4$) et trois conséquences dans la partition de la sortie floue ($N_c = 3$).

On remarque que les liens vers les conséquences dont l'information heuristique est supérieure à zéro sont considérés dans la figure 10b. Et on observe que la conséquence B_3 ne peut pas être attribué à la règle R_1 . Et B_1 ne peut pas être attribué à la règle R_3 . Et B_1, B_2 peuvent pas être attribués a R_4 car leurs informations heuristiques sont nuls.

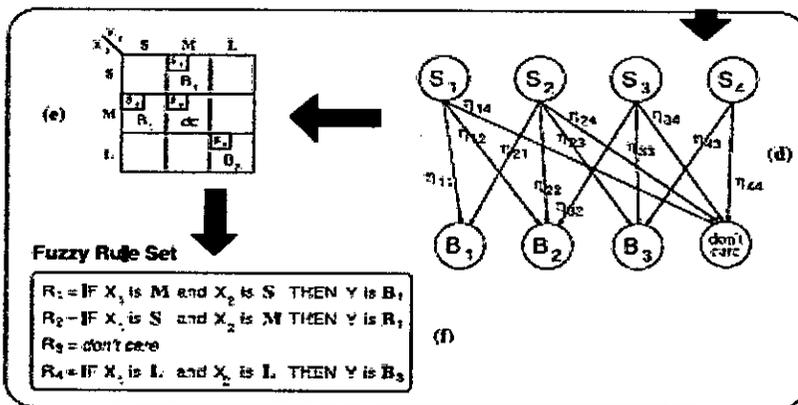


Figure 13: Processus d'apprentissage

3.5.3.Initialisation du phéromone

La valeur initiale de phéromone de chaque assignement est obtenue comme suit :

$$\tau_0 = \left(1/|S+|\right) \cdot \sum \text{Max } \eta_{jk} \quad (3.1)$$

La valeur initiale de phéromone sera la valeur moyenne du chemin construit en prenant le meilleur conséquent dans chaque règle en accordance avec l'information heuristique .

3.5.4.Fonction d'adéquation (fitness)

La fonction d'adéquation établit la qualité de la solution c'est la fonction appelée erreur quadratique moyenne déjà définie sur la figure 9.

MSE (mean square error) qui est définit par :

$$\text{MSE} = (1/2n) \cdot \sum (F(x_1^1, \dots, x_n^1) - y^1)^2 \quad (3.2)$$

Avec : y^1 sortie désirée.

$F_K(x_0^1)$: La sortie obtenue à partir de FRBS en utilisant la base des règles.

La valeur de MSE la plus proche de zéro correspond à la meilleure solution.

3.5.5.L'algorithme du meilleur mauvais système de fourmis BWAS

Une fois que les composantes précédentes ont été définies, un algorithme BWAS (Best Worst Ant System) doit être donné pour résoudre le problème [44].

Les prochaines sous-sections présentent son mode d'opération adapté au problème d'apprentissage des règles floues.

- a. Procédé de construction de solution.

- b. Processus d'évaporation de phéromone.
- c. Processus de recherche locale.
- d. La meilleure mauvaise règle de mise à jour de la traînée de phéromone.
- e. Mutation de la traînée de phéromone.
- f. Relancement du processus de recherche si le critère d'arrêt n'est pas satisfait.

3.5.5.1. Procédé de construction de solution

Dans l'algorithme BWAS, chaque étape de construction de solution est établie par l'assignement d'un conséquent à une règle avec une probabilité donnée par la règle de transition suivante [45] :

$$p(j, k) = \begin{cases} \frac{(\tau_{jk})^\alpha \cdot (\eta_{jk})^\beta}{\sum (\tau_{jk})^\alpha \cdot (\eta_{jk})^\beta} & \text{Si } R_k \in CR(S_j) \\ R_u \in CR(S_j) & \\ 0, & \text{sinon} \end{cases} \quad (3.3)$$

Avec :

τ_{jk} : La phéromone du chemin (j, k). (c.-à-d. La phéromone associée à la décision d'assigner le conséquent B_k au sous-espace S_j).

η_{jk} : L'information heuristique.

α, β : Les paramètres qui déterminent l'importance et l'influence relative de la traînée de phéromone et de l'information heuristique.

Nous devrions noter que dans un problème QAP (le problème d'affectation quadratique), la règle de transition devient une règle d'assignement mais contrairement à ce problème, on n'a pas besoin de la fournir pour garder une liste tabou avec les précédents assignements, puisque le même conséquent peut être assigné à différentes règles.

3.5.5.2. Processus d'évaporation de phéromone

L'évaporation de phéromone est faite selon la formule suivante :

$$\tau_{jk} \leftarrow (1 - \rho)\tau_{jk} \quad , \forall j , \forall k \quad (3.4)$$

Avec :

$\rho \in [0, 1]$ Étant le paramètre d'évaporation de phéromones

3.5.5.3.Processus de recherche locale

Une des manières les plus habituelles d'améliorer l'exécution des algorithmes ACO, est l'utilisation des techniques de recherche locale [46 47]. Cette approche nécessite l'utilisation d'une technique d'optimisation locale pour raffiner les solutions obtenues après une ou plusieurs itérations. L'utilisation des procédures de recherche locale, améliore habituellement l'efficacité de l'algorithme ACO qui augmente aussi le nombre d'évaluations à chaque itération ,et donc le temps d'exécution de la méthode d'apprentissage, ainsi on aura une perte d'efficacité .Donc, nous devons considérer que dans le problème d'apprentissage de règles floues, par rapport à d'autres applications, le temps nécessaire pour évaluer une solution voisine est plus grand que celui requis pour construire une nouvelle solution.

Dans les ACO, la technique de recherche locale est habituellement appliquée à la solution générée par chaque fourmi dans chaque itération. Cependant, en raison des restrictions de temps imposée dans le problème d'apprentissage des règles floues, et pour garder la grande vitesse de l'approche d'apprentissage proposée, le processus de recherche locale sera seulement appliqué à la meilleure solution produite à chaque itération.

Le principe général de ces méthodes est le suivant :

A partir d'une solution initiale $T_{courant}$ dont on connaît la solution correspondante au meilleur chemin trouvé dans l'itération courante ; on cherche la meilleure solution $T_{meilleure}$ dans le voisinage de $T_{courant}$. Si on ne parvient pas à améliorer $T_{courant}$ alors on s'arrête, sinon on remplace $T_{courant}$ par $T_{meilleure}$, et on recommence le processus[48].

Dans notre cas la traînée de phéromone globale est mise à jour d'une manière habituelle et la recherche locale proposée consiste dans le simple algorithme décrit sur la figure 14.

Soient LS_i et LS_n deux valeurs données par la méthode d'apprentissage qui définissent le nombre d'itérations maximum et le nombre de solutions générées à chaque itération.

Faire le suivant :

$$-T_{\text{meilleur}} = \{R_1 \dots R_j, \dots, R_s\}$$

Mettre la solution correspondante à la meilleure recherche dans les itérations de l'ACO courant.

$$-T_{\text{courant}} \leftarrow T_{\text{meilleur}}$$

Pour $h=1 \dots LS_i$, faire :

pour $q=1 \dots LS_n$, faire :

obtenir la solution T_q^1 appliquée au mécanisme de génération de solutions voisines vers T_{courant} . $T_q^1 \leftarrow N(T_{\text{courant}})$. Cette opération rend la sélection spécifique $j \in \{1, \dots, |S^+|\}$ et changer R_j par $R_j^1 \in C_R(S_j)$ donc

$$T_q^1 = \{R_1, \dots, R_j^1, \dots, R_{|S^+|}\}.$$

si $q=1$ alors $T \leftarrow T_1^1$. Sinon, si T_q^1 est meilleur que T on pose

$$T \leftarrow T_q^1$$

si T est meilleur que T_{courant} , on pose $T_{\text{courant}} \leftarrow T$ et continuez. Sinon refaire la boucle.

On pose la meilleure solution de l'optimisation, $T_{\text{meilleur}} \leftarrow T_{\text{courant}}$.

Figure 14 :Processus de recherche locale utilisé dans l'algorithme BWAS

3.5.5.4. La meilleure mauvaise règle de mise à jour de la traînée de phéromone

La mise à jour de traînée de phéromone est faite selon la formule suivante :

$$\tau_{jk} \leftarrow \tau_{jk} + \delta\tau_{jk} \quad (3.5)$$

Avec :

$$\Delta\tau_{jk} = \begin{cases} \frac{1}{\text{MSE}(\text{FRS}_{\text{meilleur global}})} & \text{si } (j, k) \in T_{\text{meilleur global}} \\ 0 & \text{sinon} \end{cases} \quad (3.6)$$

Avec : $T_{\text{meilleur global}}$: étant la meilleure solution globale .

$\text{FRS}_{\text{meilleur global}}$: L'ensemble de règles floues correspondant.

Puis toutes les connexions existantes dans la courante mauvaise solution (de chemin) qui ne sont pas présentes dans le global meilleur sont pénalisées par un autre affaiblissement de la traînée de phéromones associé qui est une évaporation additionnelle-exécutée comme suite :

$$\forall (j, k) \in T_{\text{actuel pire}} \text{ et } (j, k) \notin T_{\text{meilleur global}}, \tau_{jk} \leftarrow (1 - P)\tau_{jk} \quad (3.7)$$

3.5.5.5. Mutation de la traînée de phéromone

Chaque rangée de la matrice de phéromone subit une mutation avec une probabilité P_M en ajoutant ou en soustrayant la même quantité de phéromone à la traînée sélectionnée (une valeur qui dépend de l'itération courante) comme suite :

$$\tau_{jk} = \begin{cases} \tau_{jk} + \text{mut}(it, \tau_{\text{moy}}), & \text{si } a = 0 \\ \tau_{jk} - \text{mut}(it, \tau_{\text{moy}}), & \text{si } a = 1 \end{cases} \quad (3.8)$$

Avec a : est une valeur aléatoire dans $\{0, 1\}$.

it : étant l'itération courante.

τ_{moy} : étant la moyenne de la traînée de phéromone sur les connexions composants la meilleure solution globale avec $\text{mut}(\cdot)$ étant :

$$\text{Mut}(it, \tau_{\text{moy}}) = \frac{it - it_r}{N_{it} - it_r} \sigma \cdot \tau_{\text{moy}} \quad (3.9)$$

Avec N_{it} : est le nombre maximum d'itérations de l'algorithme.

it_r : est la dernière itération où un relancement a été effectué.

Il convient de mentionner que la fonction mut n'empêche pas les valeurs de phéromones d'être négatives .Par conséquent, il est nécessaire de vérifier leurs corrections après chaque application de cette composante.

3.5.5.6. Relancement du processus de recherche

L'algorithme effectuera le relancement en plaçant tous les composants de la matrice de phéromone à τ_0 (la valeur initiale de phéromone) quand la meilleure solution globale n'est pas améliorée pendant un nombre fixe d'itération.

3.6. Conclusion

La méthode de la dérivation de la base des connaissances (base des règles, base de données) étudiée dans ce chapitre a été appliquée à un ballon dirigeable dont les détails de cette application sont présentés dans le chapitre suivant.

CHAPITRE 4

APPLICATION (BALLON DIRIGEABLE)

4.1. Description du ballon dirigeable

Le ballon dirigeable est une enveloppe de mylar de forme ellipsoïde remplie de gaz d'hélium, muni de propulseurs vectorisés pour assurer le contrôle de la force de poussée et de direction, il est doté d'une plate-forme permettant d'emporter le système embarqué de contrôle[49],[50].

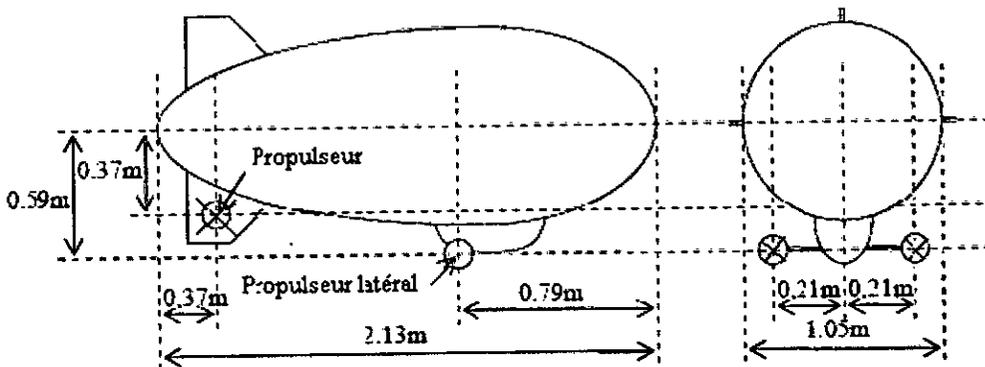


Figure 15 : Dimension du dirigeable et position des propulseurs

4.2. Modélisation du ballon dirigeable

4.2.1. Modélisation cinématique

L'étude et l'analyse du mouvement d'un ballon dirigeable se fait dans deux repères [43], un repère lié au ballon, l'autre lié à la terre que l'on appelle repère référentiel selon la formulation de Fossen [51] :

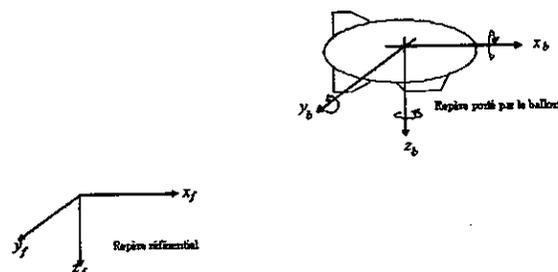


Figure 16 : Représentation des repères

La position et l'orientation du ballon sont définies dans le repère référentiel où la rotation de la terre est négligée ; sa surface est supposée plane.

Ils sont exprimés par le vecteur $\eta = [\eta_1, \eta_2]^T$

Avec : $\eta_1 = [x \ y \ z]^T$ représente les trois composantes de position

Et $\eta_2 = [\phi \ \theta \ \psi]^T$ représente les trois composantes d'orientation que l'on appelle respectivement angle de roulis, de tangage et de lacet

Les vitesses linéaires et angulaires seront définies dans le repère lié au ballon par le vecteur $v = [v_1, v_2]^T$

Où $v_1 = [v_x, v_y, v_z]^T$ représente les trois composantes du vecteur vitesse linéaire.

Et $v_2 = [w_x, w_y, w_z]^T$ représente les composantes du vitesse angulaire autour des trois axes du repère lié au ballon.

L'étude de l'orientation du ballon dirigeable utilise la formulation des angles d'Euler où le passage d'une configuration angulaire à une autre se fait par la décomposition de la matrice de passage, comme indiquée ci-dessous :

-rotation autour de l'axe y

$$R(y, \theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (4.1)$$

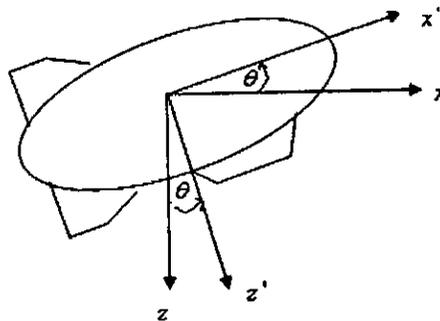


Figure 17 : Vue du profil ; rotation autour de l'axe y (angle de tangage)
-rotation autour de l'axe x :

$$R(x, \varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{bmatrix} \quad (4.2)$$

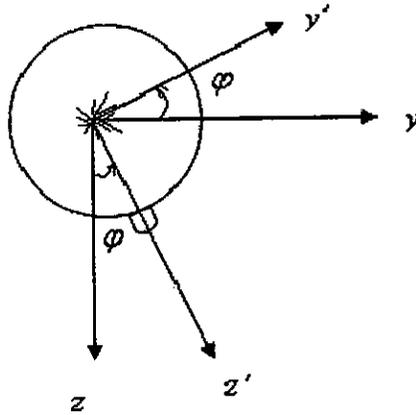


Figure 18 : Vue de l'avant, rotation autour de l'axe x, angle de roulis

Rotation autour de l'axe z :

$$R(z, \psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

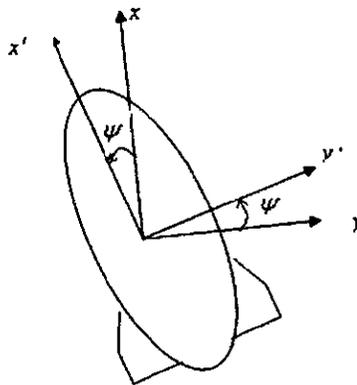


Figure 19: Vue de dessus ; le cap, rotation autour de l'axe z (angle de lacet)

La matrice de passage est donnée par l'expression :

$$R(\varphi, \theta, \psi) = \begin{bmatrix} \cos \psi \cos \theta & -\sin \psi \cos \varphi + \cos \psi \sin \theta \sin \varphi & \sin \psi \sin \varphi + \cos \varphi \cos \psi \sin \theta \\ \sin \psi \cos \theta & \sin \psi \cos \varphi + \sin \varphi \sin \theta \sin \psi & -\sin \psi \sin \varphi + \sin \theta \sin \psi \cos \theta \\ -\sin \varphi & \cos \theta \sin \varphi & \cos \theta \cos \varphi \end{bmatrix} \quad (4.4)$$

4.2.1.1. Transformation des vitesses linéaires

La relation des vitesses linéaires dans les deux repères est donnée par :

$$\dot{\eta} = J_1(\eta_2)v_1$$

Cette relation décrit la position du ballon dirigeable dans le repère inertiel où $J_1(\eta_2) = R(\psi, \theta, \varphi)$ représente la matrice de passage du repère inertiel au repère porté par le ballon dirigeable, c'est une matrice de rotation orthogonale, de déterminant unité ayant pour inverse sa transposée.

4.2.1.2. Transformation des vitesses angulaires

La relation liant les vitesses angulaires dans les deux repères est de la forme suivante :

$$\dot{\eta}_2 = J_2(\eta_2)v_2$$

$$\text{Où } J_2(\eta_2) = \begin{bmatrix} 1 & \sin \varphi \operatorname{tg} \theta & \cos \varphi \operatorname{tg} \theta \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi / \cos \theta & \cos \varphi / \cos \theta \end{bmatrix} \quad (4.5)$$

$$\text{Avec : } \theta \neq \frac{\pi}{2} + k\pi$$

En résumé le modèle cinématique du ballon dirigeable est représenté par la relation suivant :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} [J_1(\eta_2)] & [0_{3 \times 3}] \\ [0_{3 \times 3}] & [J_2(\eta_2)] \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ w_x \\ w_y \\ w_z \end{bmatrix} \quad (4.6)$$

Cette représentation a une singularité pour $\theta=90$

L'intégration de l'équation $\dot{\eta} = J(\eta)v$ nous donne la position et l'orientation du véhicule volant à chaque instant dans le repère inertiel avec :

$$\eta = [x \ y \ z \ \varphi \ \theta \ \psi]^T \quad ; \quad v = [v_x \ v_y \ v_z \ w_x \ w_y \ w_z]^T$$

$$J(\eta) = \begin{bmatrix} [J_1(\eta_2)] & [0_{3 \times 3}] \\ [0_{3 \times 3}] & [J_2(\eta_2)] \end{bmatrix} \quad (4.7)$$

4.2.2. Modélisation dynamique

Les équations qui décrivent la dynamique du ballon dirigeable sont déduites des lois de Newton de la mécanique classique et des lois relatives à la mécanique des fluides. Pour obtenir la dynamique du système, on applique les principes du corps rigide à 6 ddl par rapport au repère référentiel [52], [53] [54] :

Soit un corps rigide de forme quelconque comme centre de gravité le point G, le centre du repère lié à ce corps est le point o (cf. Figure 20)

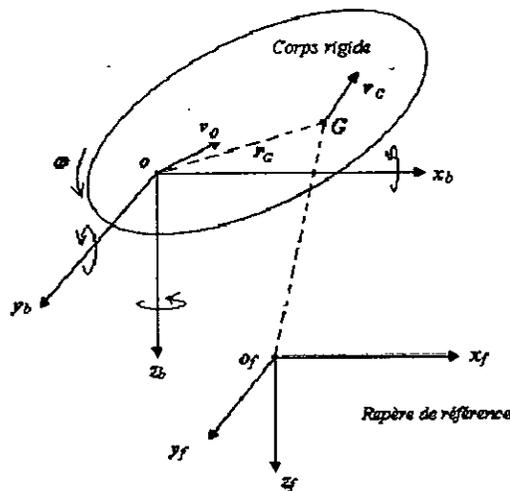


Figure 20 : Représentation en 3ddl d'un corps rigide

Le principe de la dynamique du solide décrit l'ensemble des équations suivantes :

La vitesse et l'accélération du corps sont données par :

$$\begin{aligned} v_G &= v_o + \omega \wedge r_G \\ a_G &= a_o + \omega \wedge v_o + \omega \wedge r_G + \omega \wedge (\omega \wedge r_G) \end{aligned} \quad (4.8)$$

Le moment cinétique par rapport à o est donné par :

$$H = I_0 \omega + m r_G \wedge v_0 \quad (4.9)$$

D'après la loi de la dynamique la force et le couple sont données par :

$$m a_G = f_0 \quad \text{et} \quad \frac{dH}{dt} = M_0 \quad (4.10)$$

Alors, l'équation du mouvement de translation est

$$m(a_0 + \omega \wedge v_0 + \dot{\omega} \wedge r_G + \omega \wedge (\omega \wedge r_G)) = f_0 \quad (4.11)$$

Et l'équation du mouvement de rotation est

$$I_0 \dot{\omega} + \omega \wedge (I_0 \omega) + (m r_G) \wedge (v_0 + \omega \wedge v_0) = M_0 \quad (4.12)$$

Où

$$f_0 = \Gamma_1 = [X \ Y \ Z]^T \text{ forces externes}$$

$$M_0 = \Gamma_2 = [K \ M \ N]^T \text{ moments externes}$$

$$v_0 = v_1 = [v_x \ v_y \ v_z]^T \text{ vitesses linéaires}$$

$$\omega = v_1 = [\omega_x \ \omega_y \ \omega_z]^T \text{ vitesses angulaires}$$

$$r_G = [x_G \ y_G \ z_G]^T \text{ centre de gravité}$$

I_0 est la matrice d'inertie du solide

$$I_0 = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \text{ où } I_0 = I_0^T > 0 \quad (4.13)$$

Si le solide a un plan de symétrie

$$I_0 = \text{diag}[I_{xx} \ I_{yy} \ I_{zz}] \quad (4.14)$$

Pour l'étude de la dynamique du ballon dirigeable, on se base sur les hypothèses suivantes :

- la masse et le volume du ballon sont des grandeurs invariables
- les phénomènes d'aéroélasticités sont négligés
- les phénomènes liés au gaz d'hélium sont négligés
- le centre de poussée est supposé au centre du volume
- le ballon comporte des plans de symétries, le centre de gravité et le centre de flottabilité se trouvent dans l'intersection de ces deux plans
- le centre de gravité coïncide avec le centre du repère lié au ballon ($r_g = 0$)
- le nombre de mach est faible (< 3) et le nombre de réynolds est faible ($\sim 10^4$)
- la viscosité du fluide (de l'air) est supposée constante
- le centre de flottabilité est au-dessus de centre de gravité.

En tenant compte de ces hypothèses les expressions (4.13) et (4.14) deviennent :

$$m(\dot{v}_0 + \omega \wedge v_0) = f_0 \quad (4.15)$$

$$I_0 \dot{\omega} + \omega \wedge (I_0 \omega) = M_0 \quad (4.16)$$

En projetant ces deux équations sur les différents axes du repère lié au dirigeable on obtient :

$$\begin{aligned} m[\dot{v}_x - v_y \omega_z + v_z \omega_y] &= X \\ m[\dot{v}_y - v_z \omega_x + v_x \omega_z] &= Y \\ m[\dot{v}_z - v_x \omega_y + v_y \omega_x] &= Z \\ I_{xx} \dot{\omega}_x + (I_{zz} - I_{yy}) \omega_y \omega_z &= K \\ I_{yy} \dot{\omega}_y + (I_{xx} - I_{zz}) \omega_z \omega_x &= M \end{aligned} \quad (4.17)$$

$$I_{zz}\dot{\omega}_z + (I_{yy} - I_{xx})\omega_x\omega_y = N$$

Ces équations peuvent se mettre sous la forme matricielle selon la formulation de Fossen

$$M_{RB}\dot{v} + C_{RB}(v)v = Y_{RB}$$

Avec :

$$v = [v_x \ v_y \ v_z \ w_x \ w_y \ w_z]^T$$

M_{RB} : Matrice d'inertie du système, c'est une matrice symétrique qui s'exprime ainsi :

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{zz} \end{bmatrix} \quad (4.18)$$

m: masse du ballon

C_{RB} : est la matrice des forces de coriolis et des forces centrifuges qui s'appliquent au ballon dirigeable. C'est une matrice antisymétrique de dimension 6x6 et d'expression :

$$C_{RB} = \begin{bmatrix} 0 & 0 & 0 & 0 & mv_x & -mv_y \\ 0 & 0 & 0 & -mv_z & 0 & mv_x \\ 0 & 0 & 0 & mv_y & mv_x & 0 \\ 0 & mv_z & -mv_y & 0 & I_{zz}\omega_z & -I_{yy}\omega_y \\ -mv_z & 0 & mv_x & -I_{zz}\omega_z & 0 & I_{xx}\omega_x \\ mv_y & -mv_x & 0 & I_{yy}\omega_y & -I_{xx}\omega_x & 0 \end{bmatrix} \quad (4.19)$$

4.2.2.1. Les efforts hydrodynamiques

Le dirigeable est considéré comme un corps submergé dans un fluide, lors de son déplacement il apparaît des efforts hydrodynamiques, ils sont dus aux échanges d'énergie cinétique entre le ballon et l'air environnant. Les efforts s'appliquent au centre de poussée ou de flottabilité, ils dépendent de la densité du fluide et de la géométrie du mobile, on suppose que le dirigeable est submergé

dans un fluide parfait, volume d'air infini, incompressible, non visqueux irrotationnel et à l'état de repos à l'infini.

Soit (x,y,z) les coordonnées d'un point situé sur le véhicule ou dans le fluide, correspondant aux système d'axe lié au véhicule ,on suppose qu'on peut caractériser l'écoulement du fluide par une fonction scalaire $\phi(x,y,z)$ appelée potentiel des vitesses, définie par

$$\mathbf{u} = -\nabla\phi = \left(-\frac{\partial\phi}{\partial x}, -\frac{\partial\phi}{\partial y}, -\frac{\partial\phi}{\partial z} \right) \quad (4.20)$$

Où \mathbf{u} est la vitesse locale du fluide comme le fluide est incompressible l'équation de continuité devient :

$$\nabla^2\phi = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2} = 0 \quad (4.21)$$

Soit \mathbf{n} un vecteur normal à la surface du dirigeable orienté vers le fluide, alors, en tout point de la surface on a :

$$-\nabla\phi \cdot \mathbf{n} = v_n$$

Où v_n est la composante de la vitesse de la surface au point (x, y, z) dans la direction de \mathbf{n} comme le fluide est supposé au repos à l'infini on a :

$$\frac{\partial\phi}{\partial x} = 0, \frac{\partial\phi}{\partial y} = 0, \frac{\partial\phi}{\partial z} = 0 \quad (4.22)$$

Etant donné les hypothèses précédentes, Kirchoff montra qu'il existe des fonctions de (x,y,z) noté $\phi_1, \phi_2, \phi_3, X_1, X_2, X_3$ déterminée d'une façon unique par la configuration de la surface du véhicule tel que :

$$\phi = v_x\phi_1 + v_y\phi_2 + v_z\phi_3 + \omega_x X_1 + \omega_y X_2 + \omega_z X_3 \quad (4.23)$$

En utilisant le théorème de Green, l'énergie cinétique du fluide t_f est donnée par :

$$2T_f = \rho_0 \iiint \left(\left(\frac{\partial \phi}{\partial x} \right)^2 + \left(\frac{\partial \phi}{\partial y} \right)^2 + \left(\frac{\partial \phi}{\partial z} \right)^2 \right) dx dy dz = -\rho_0 \iint \phi (\nabla \phi \cdot n) dS \quad (4.24)$$

Avec : ρ_0 la densité du fluide.

$-\nabla \phi n$ représente la vitesse normale du fluide à la surface.

ρ la pression nécessaire pour générer le mouvement.

Soit le vecteur vitesse du dirigeable v :

$$v = [v_x \quad v_y \quad v_z \quad \omega_x \quad \omega_y \quad \omega_z]^T$$

Le vecteur vitesse du dirigeable

En substituant (4.23) dans (4.24) l'énergie cinétique est

$$T_f = \frac{1}{2} v^T M_a v \quad (4.25)$$

Où M_a une matrice diagonale.

M_a est une matrice diagonale dans le cas où le corps a des plans de symétrie dont les éléments représentent les masses et les inerties de la partie du fluide induite en mouvement avec la même vitesse que le corps du solide et qui dépendent de la forme de la surface de ce corps et de la densité du fluide .

Généralement dans l'étude de la dynamique des corps solides au milieu d'un fluide tel que le sous-marin, torpille, ballon dirigeable, l'existence du fluide est ignorée et la masse et l'inertie du fluide induit en mouvement sont ajoutés à la masse et à l'inertie du corps du solide M_a ; est dite matrice de masses ajoutées. En résumé les efforts hydrodynamiques sont dus aux échanges d'énergie

cinétique entre le corps du dirigeable et les particules du fluide induites en mouvement.

Donc l'énergie cinétique des efforts de la masse ajoutée peut être définie comme suit :

$$T_a = \frac{1}{2} v^T M_a v$$

Les forces et les couples générés par l'échange d'énergie cinétique du fluide sont déterminés par les équations de Kirchoff [55][56]

$$\frac{d}{dt} \left(\frac{\partial T_a}{\partial v_1} \right) + v_2 \wedge \frac{\partial T_a}{\partial v_1} = -f_a \quad \text{force} \quad (4.26)$$

$$\frac{d}{dt} \left(\frac{\partial T_a}{\partial v_n} \right) + v_2 \wedge \frac{\partial T_a}{\partial v_2} + v_1 \wedge \frac{\partial T_a}{\partial v_1} = -n_a \quad \text{Couple} \quad (4.27)$$

Ces deux équations nous donnent l'expression du torseur de la force et du couple induit par l'inertie et la masse ajoutée.

$$\Gamma_a = (M_a \dot{v} + C_a(v)v) \quad (4.28)$$

Où M_a est la matrice d'inertie ajoutée dont les termes sont constants :

$$M_a = \begin{bmatrix} a_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{66} \end{bmatrix} \quad (4.29)$$

C_a est la matrice de force de coriolis et de forces de centrifuges hydrodynamiques et peut se mettre sous la formes :

$$C_a = \begin{bmatrix} 0 & 0 & 0 & 0 & a_{33}v_z & -a_{22}v_y \\ 0 & 0 & 0 & -a_{33}v_z & 0 & a_{11}v_x \\ 0 & 0 & 0 & a_{22}v_y & -a_{11}v_x & 0 \\ 0 & a_{33}v_z & -a_{22}v_y & 0 & a_{66}\omega_z & -a_{55}\omega_y \\ -a_{33}v_z & 0 & a_{11}v_x & -a_{66}\omega_z & 0 & a_{44}\omega_x \\ a_{22}v_y & -a_{11}v_x & 0 & a_{55}\omega_y & -a_{44}\omega_x & 0 \end{bmatrix} \quad (4.30)$$

4.2.2.2. Les forces d'amortissements

Les forces d'amortissement sont générées par un transfert d'énergie entre le fluide et le solide immergé, leur point d'application est le centre de flottabilité du ballon dirigeable.

Elles sont représentées par une matrice $\mathbf{d}(\mathbf{v})$ strictement négative, signifiant que les forces et les couples sont appliqués au corps par l'environnement, ces efforts sont proportionnels à la vitesse

$$\tau_D = D(\mathbf{v}) \cdot \mathbf{v}$$

Ces efforts sont dus principalement aux trois facteurs suivants :

-la force d'amortissement due à la rugosité du solide.

-la force d'amortissement induite lors de la formation des tourbillons à l'arrière d'un solide.

-la force d'amortissement due à la forme du dirigeable, cette force est modélisée comme suit :

$$D_f = -\frac{1}{2} \rho C_D A |\mathbf{v}| \mathbf{v} \quad (4.31)$$

Où

ρ : densité du fluide

C_D : Coefficient de traînée

A : surface du corps en incidence

\mathbf{v} : vecteur vitesse

La matrice d'amortissement $D(\mathbf{v})$ est donnée par [56][57] :

$$D(\mathbf{v}) = \text{diag}\{D_{v_x} + D_{v_x}\} \quad (4.32)$$

Les coefficients d'amortissement sont déterminés expérimentalement

4.2.2.3. Les efforts hydrostatiques, poussée d'Archimède et force de pesanteur

Ces efforts se présentent quand le dirigeable est à l'état de flottement sous l'effet de la combinaison des actions de la force de poussée qui s'applique au centre de poussée et à la force de pesanteur qui agit au centre de gravité

La poussée d'Archimède est égale à la force opposée au poids que subit le volume du fluide

$$F_B = \rho V_f g \quad (4.33)$$

ρ : la densité du fluide

V_f : le volume du fluide déplacé

g : l'accélération gravitationnelle

La force de pesanteur est exprimée par $F_G = mg$ où m est la masse du solide . La résultante des forces hydrostatiques est donnée par : [58][59]

$$g(\eta) = \begin{bmatrix} (W - B) \sin \theta \\ -(W - B) \cos \theta \sin \phi \\ -(W - B) \cos \theta \cos \phi \\ -(y_G W - y_B B) \cos \theta \cos \phi + (z_G W - z_B B) \cos \theta \sin \phi \\ (z_G W - z_B B) \sin \theta + (x_G W - x_B B) \cos \theta \cos \phi \\ -(x_G W - x_B B) \cos \theta \sin \phi - (y_G W - y_B B) \sin \theta \end{bmatrix} \quad (4.34)$$

Avec $W = \|F_G\|$ et $B = \|F_B\|$

4.2.2.4. Synthèse de la dynamique du dirigeable

Finalement l'équation dynamique du ballon dirigeable est la suivante [48],[52] :

$$M\dot{v} + C(v)v + D(v)v + g(\eta) = \tau \quad (4.35)$$

Où

$M = M_{RB} + M_a$: Matrice 6x6 contenant les termes de masse et d'inertie M_{RB}

Est définie en (4.28) et M_a en (4.17)

$C(v) = C_{RB}(v) + C_a(v)$: matrice 6x6 des termes de forces de coriolis et de centrifuges C_{RB} est déterminé en (4.18) et C_a en (4.29)

$D(v)$: représente une matrice contenant des termes de forces

D'amortissements, est donnée par (4.31)

$g(\eta)$: matrice définie en (4.33) englobe les termes des forces

Hydrostatiques

τ : forces et couples des propulseurs

v : torseur cinématique du ballon dirigeable.

Avec $v = [v_x \ v_y \ v_z \ \omega_x \ \omega_y \ \omega_z]^T$

4.3. Contrôle du ballon dirigeable par un contrôleur flou simple

4.3.1. Stratégie du vol du ballon dirigeable

Le pilotage automatique du dirigeable est assuré par deux commandes une à action latérale et l'autre à action longitudinale [59][60].

La stratégie du vol du ballon dirigeable sera divisée en quatre phases :

- a. **Phase de décollage** : Pour le décollage ou le changement d'altitude, le dirigeable est commandé en mode longitudinal, le mouvement se produit dans le plan vertical.
- b. **Phase de maintien** : c'est une étape de préparation a la navigation ou deux régulateurs sont mis en œuvre, l'un pour contrôler l'altitude désirée, l'autre pour maintenir la vitesse constante, ces deux commandes agissent en mode longitudinal.
- c. **Phase de navigation** : On commute vers le mode latéral, la commande agit sur la variation horizontale du dirigeable.
- d. **Phase d'atterrissage** : Pour un capteur désiré, la descente est déclenchée. La commande est commutée vers le mode longitudinal.

Dans notre étude on s'intéresse à la phase de navigation donc au processus latéral.

4.3.2.Synthèse d'un contrôleur flou pour la commande de l'action latérale et longitudinale

Comme vu précédemment, la synthèse d'un contrôleur flou dépend principalement des caractéristiques des modules de **fuzzification**, de la **base de connaissances** (base de données et base de règles) et de **défuzzification** (Voir Chapitre 2).

Pour notre application nous avons opté pour un contrôleur flou à deux entrées et une seule sortie.

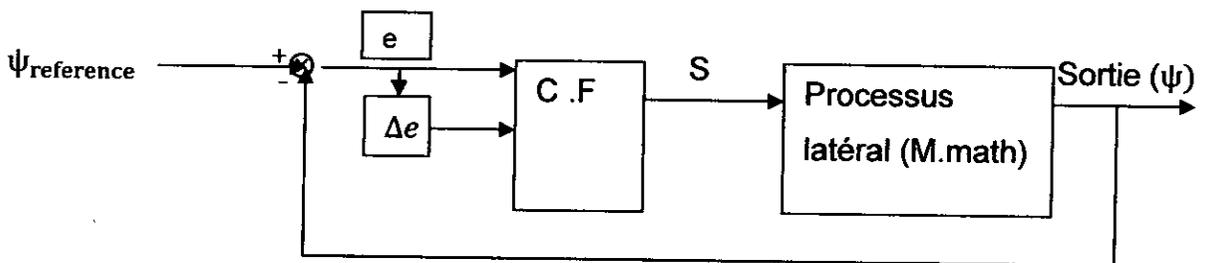


Figure 21: Contrôle flou du processus latéral

Avec :

e : L'erreur entre le signal de sortie de processus et le signal désiré.

Δe : La variation d'erreur

S : La sortie chargée de la commande de l'orientation du dirigeable.

ψ : Angle de lacet.

4.3.3.Les équations décrivant la dynamique du vol

On schématise le synoptique du système comme suit :

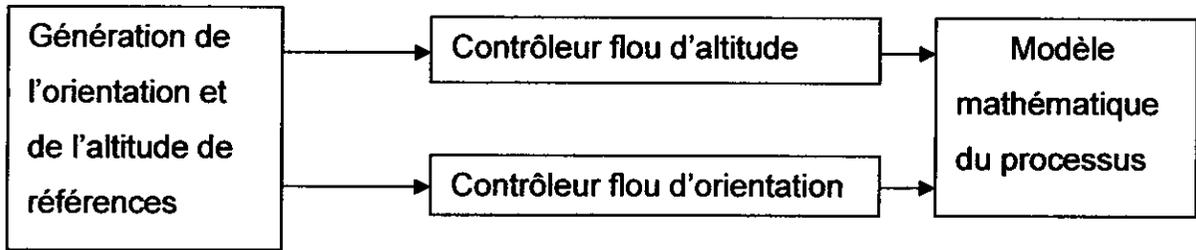


Figure 22 : Schéma synoptique du système

On découple les équations décrivant la dynamique du dirigeable en deux modes et on néglige la rotation autour de l'axe x ; le roulis $\phi = 0$, $\omega_x = 0$, le mode longitudinal et le mode latéral.

a. Mode longitudinal :

Les équations décrivant la dynamique du système en mode longitudinal :

$$\begin{aligned}
 \dot{m}_x v_x &= -m_z v_z \omega_y + m_y v_y \omega_z - d_1 v_x + F \cos \alpha \\
 \dot{m}_z v_z &= m_x v_x \omega_y - m_y v_y \omega_z - d_3 v_z + F \sin \alpha \\
 J_y \dot{\omega}_y &= (m_z - m_x) v_x v_z - d_5 \omega_y + l_y F \\
 \dot{x} &= v_x \cos \theta + v_z \sin \theta \\
 \dot{z} &= -v_x \sin \theta + v_z \cos \theta \\
 \dot{\theta} &= \omega_y
 \end{aligned} \tag{4.36}$$

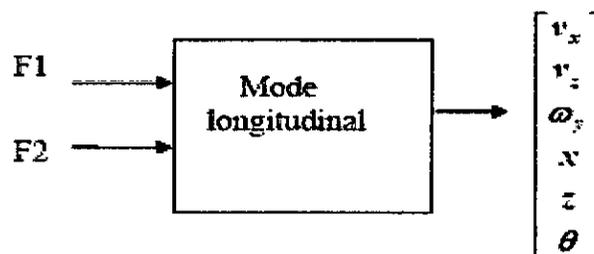


Figure 23 : Synoptique du mode longitudinal

Avec :

$$F_1 = F \cos \alpha$$

$$F_2 = F \sin \alpha$$

b. Mode latéral :

Modèle du processus en mode latéral :

$$m_y \dot{v}_y = +m_x v_x \omega_z - d_2 v_y + F_3$$

$$J_x \dot{\omega}_x = (m_x - m_y) v_x v_y - d_6 \omega_x + F_3 l_x$$

$$\dot{x} = v_x \cos \psi - v_y \sin \psi$$

$$\dot{y} = v_x \sin \psi + v_y \cos \psi$$

$$\dot{\psi} = \omega_z$$

(4.37)

C'est ce modèle qui nous intéresse et qui nous permettra par la suite d'implémenter notre contrôleur flou (voir page 84)

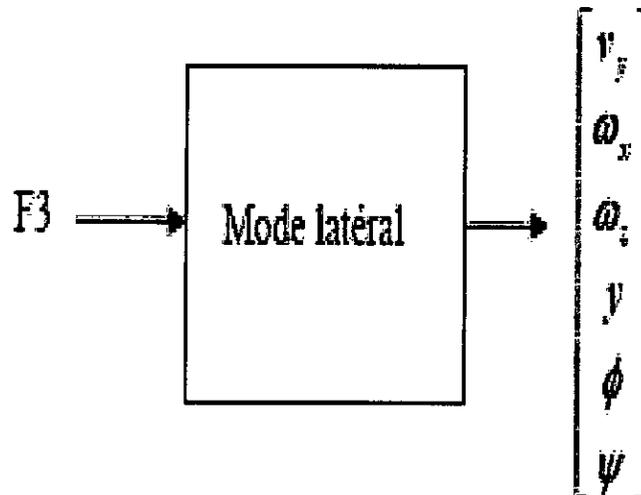


Figure 24: Synoptique du mode latéral

Avec :

$S = F_3$: La sortie chargée de la commande de l'orientation du dirigeable.

F_3 : Force de propulseur attachée à la queue du dirigeable ,elle commande l'orientation du cap (différentielle de force).

4.4.Synthèse d'un premier contrôleur flou

Le contrôleur flou adopté est de type Mamdani à deux entrées et une seule sortie.

Pour un premier essai et une première synthèse de notre contrôleur flou, les caractéristiques, tel que la position, la forme et les ouvertures des variables linguistiques (base de données) qualifiants les entrées et la sortie du contrôleur, ainsi que la base des règles sont choisies d'une façon **intuitive** ou en suivant les directives d'un expert dans le domaine. En se basant sur le fonctionnement de notre système et selon l'objectif désiré on a simulé deux contrôleurs flous (un pour le contrôle de l'orientation, l'autre pour l'altitude) sous **Matlab** en utilisant **simulink**:

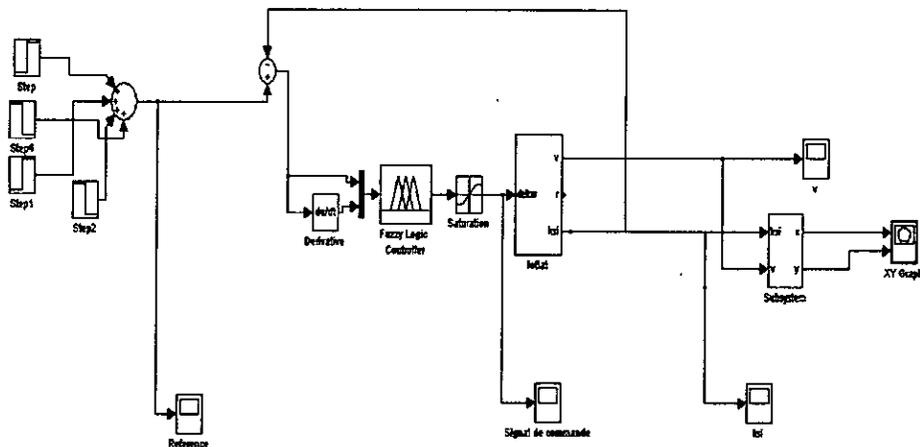


Figure 25 : Simulation de notre système sous simulink

4.4.1.Résultats de la première simulation (C.F simple)

- Première base des règles (première expertise)

e Δe	pb	pm	ps	ze	nb	nm	ns
pb	nb	nb	nb	nb	ze	ns	nm
pm	nb	nb	nb	nm	ps	ze	ns
ps	nb	nb	nm	ns	pm	ps	ze
ze	nb	nm	ns	ze	pm	pm	ps
nb	ze	ps	pm	pb	pb	pb	pb
nm	ns	ze	ps	pm	pb	pb	pb
ns	nm	ns	ze	ps	pb	pb	pm

Figure 26 : Base des règles

Avec les termes linguistiques :

nb : negativ big (negatif grand)

pb : positiv big (positif grand)

nm : negativ medium (negatif moyen)

ns : negativ small (négatif petit)

pm : positiv medium (positif moyen)

ps : positiv small (positif petit)

ze : environ zéro

- Base de données

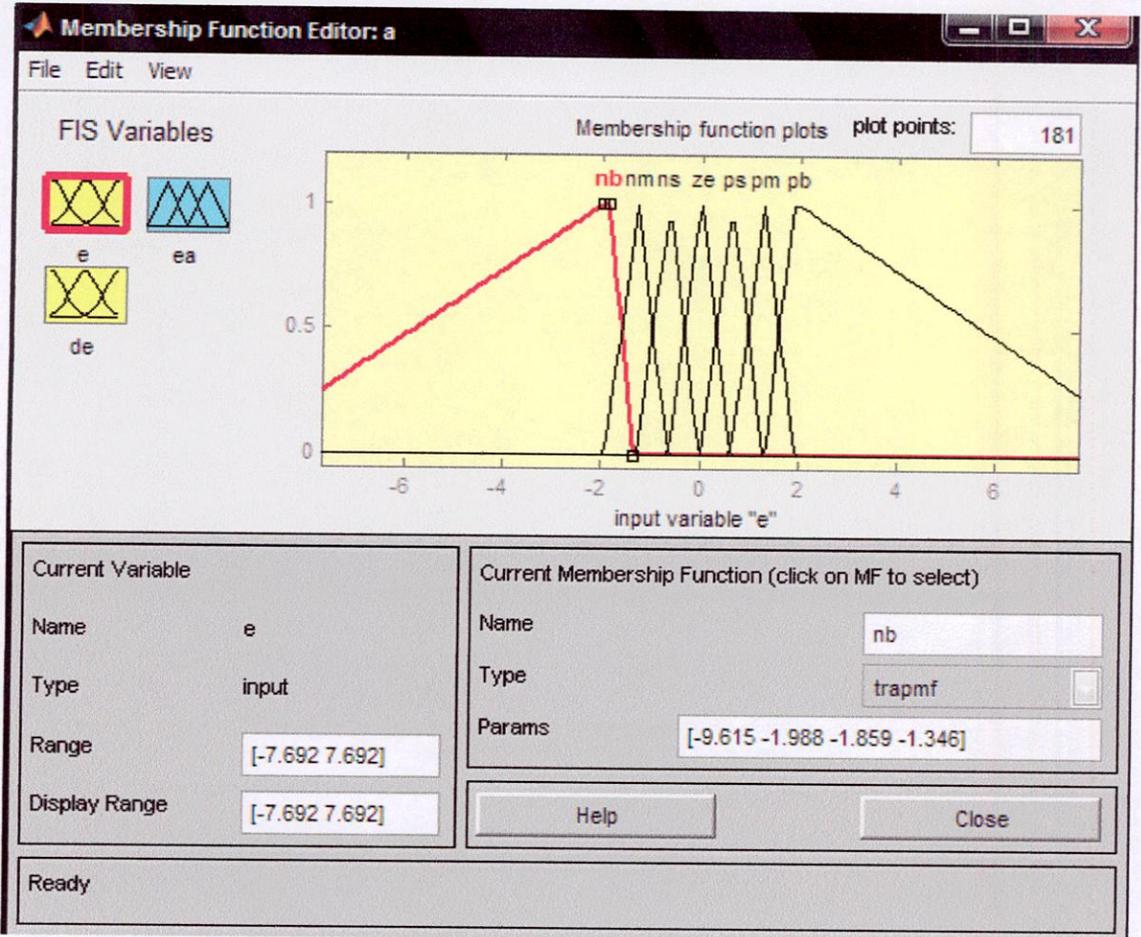


Figure 27 : Fonctions d'appartenances entrée e

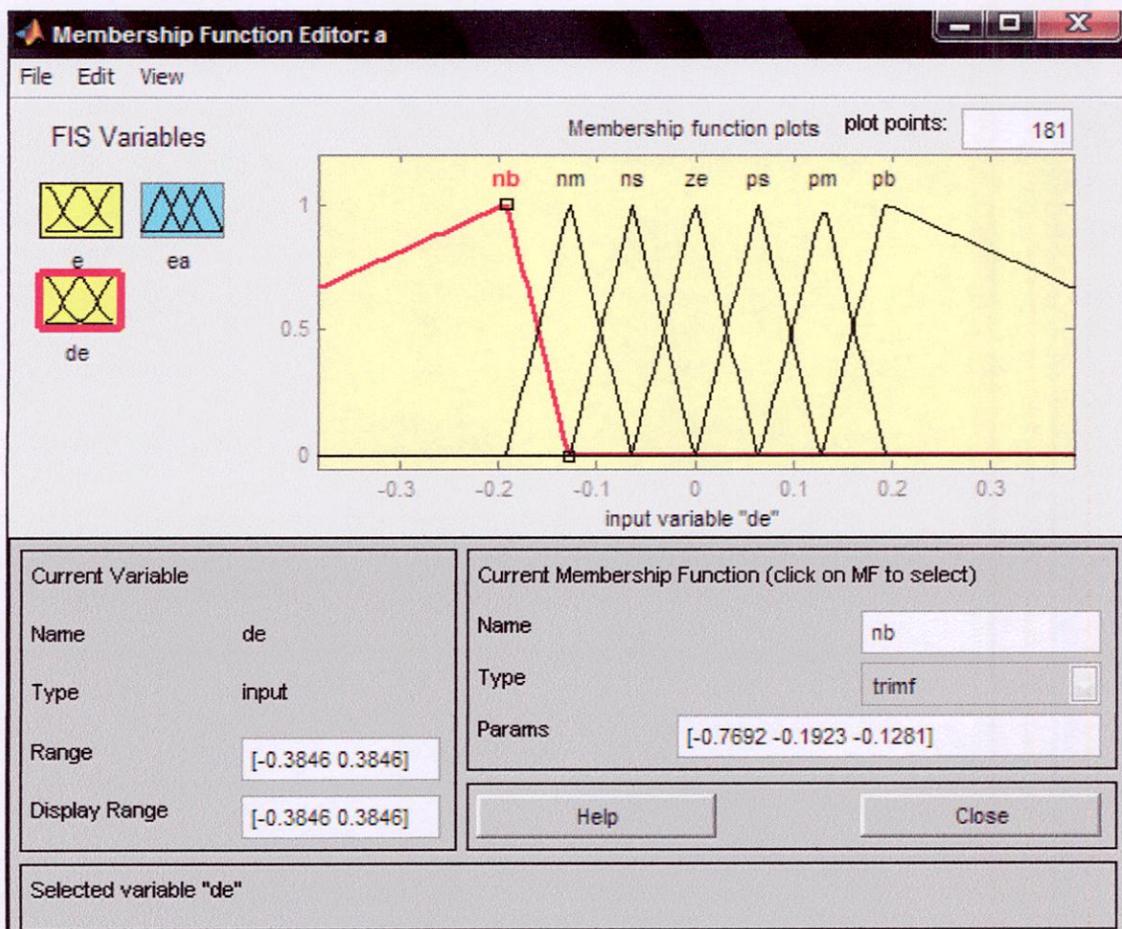


Figure 28 : Fonctions d'appartenances variation d'erreur Δe

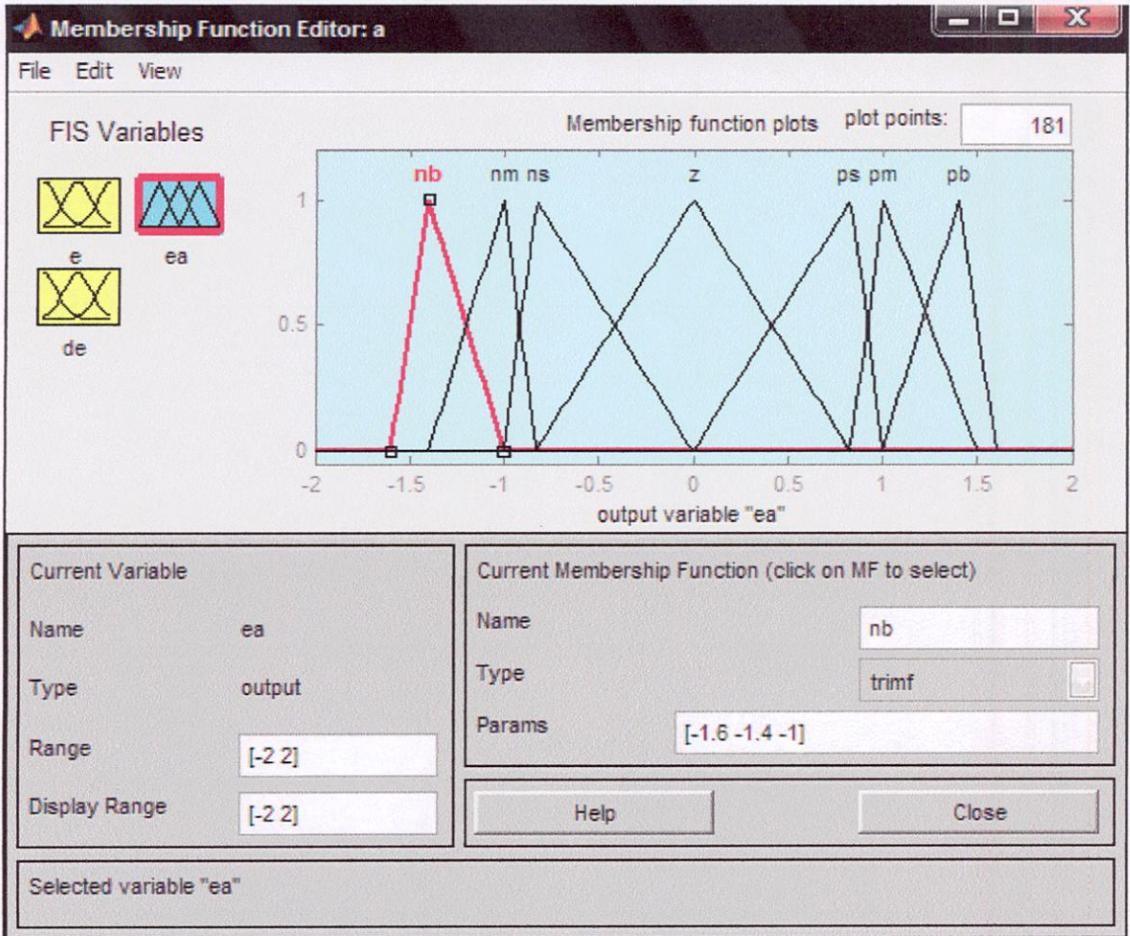


Figure 29 : Sortie du contrôleur (Commande ea)

- La sortie du processus du ballon dirigeable et le suivi de la trajectoire

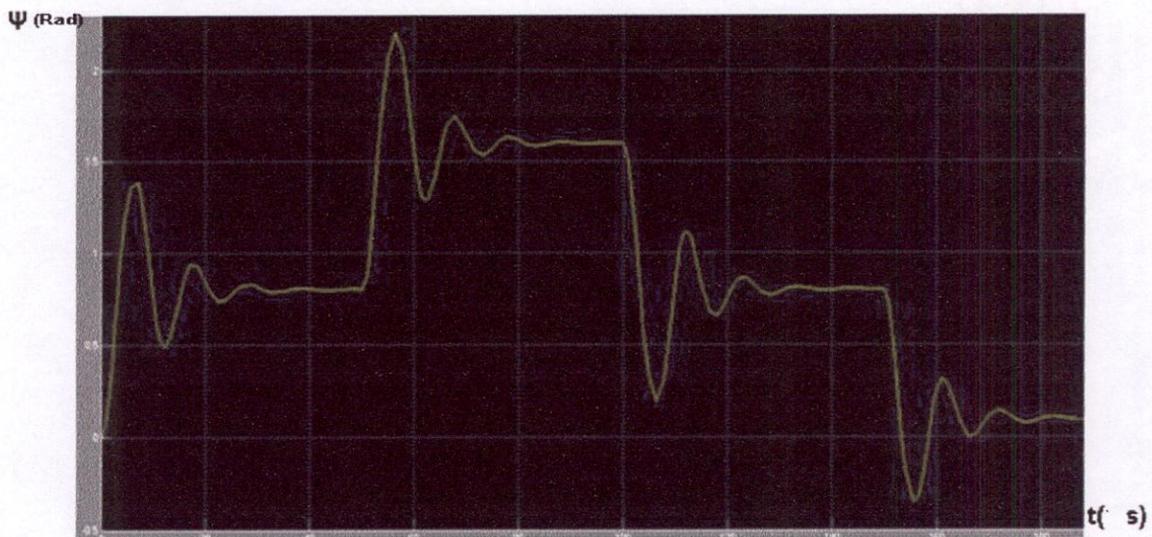


Figure 30 : La sortie du processus

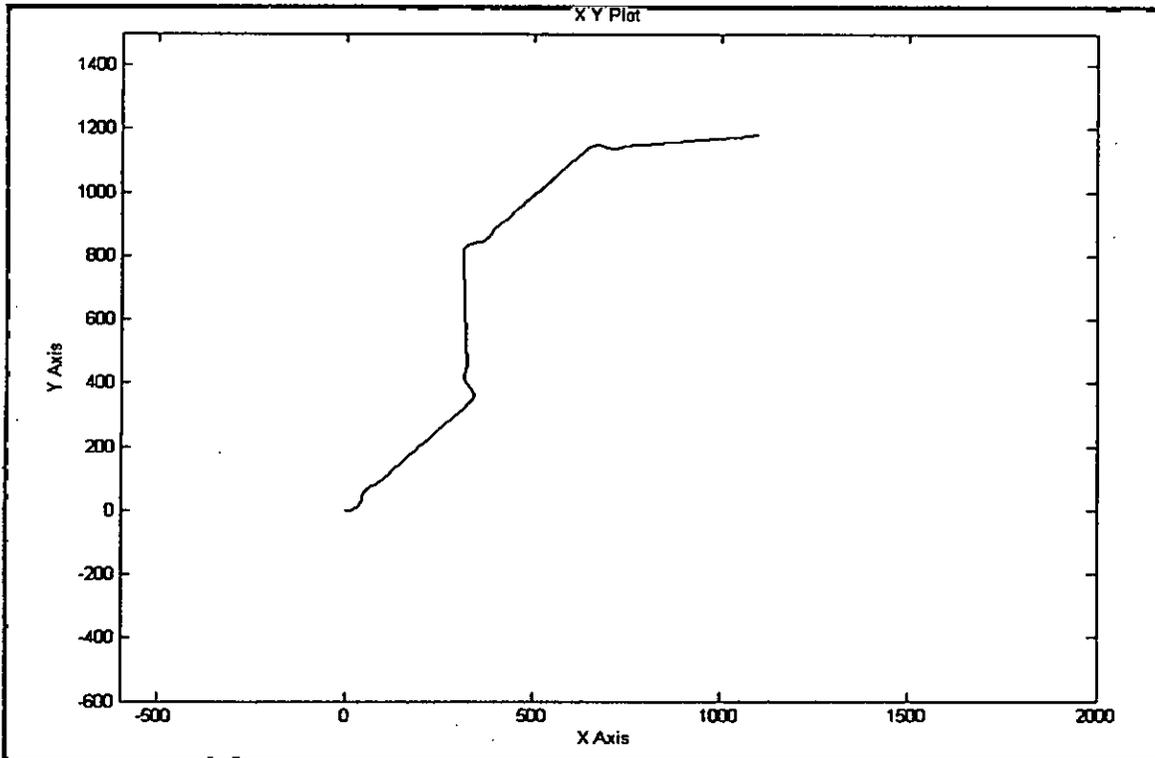


Figure 31 : Suivi de la trajectoire

- Superposition du signal de sortie du processus et le signal de référence

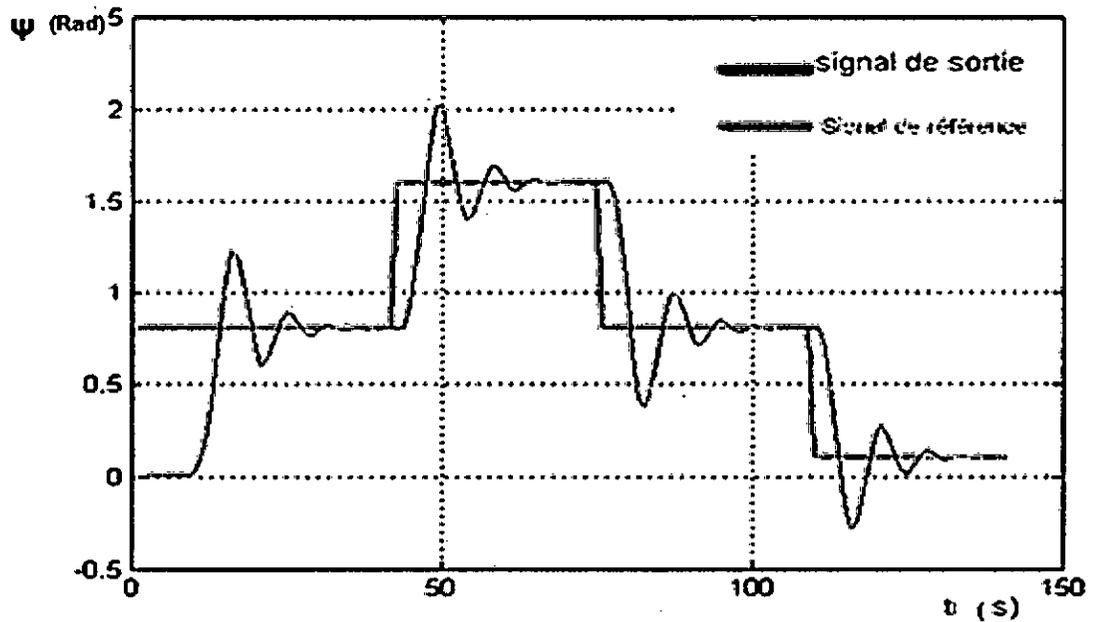


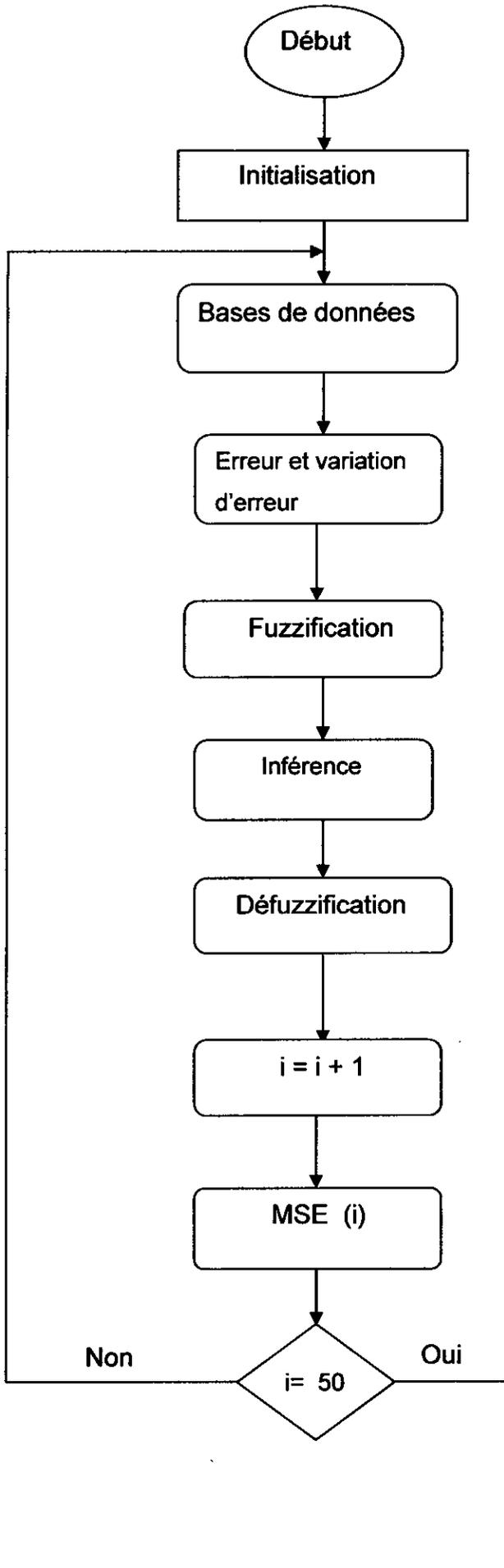
Figure 32 : Superposition des deux signaux (Sortie du processus – Signal de référence)

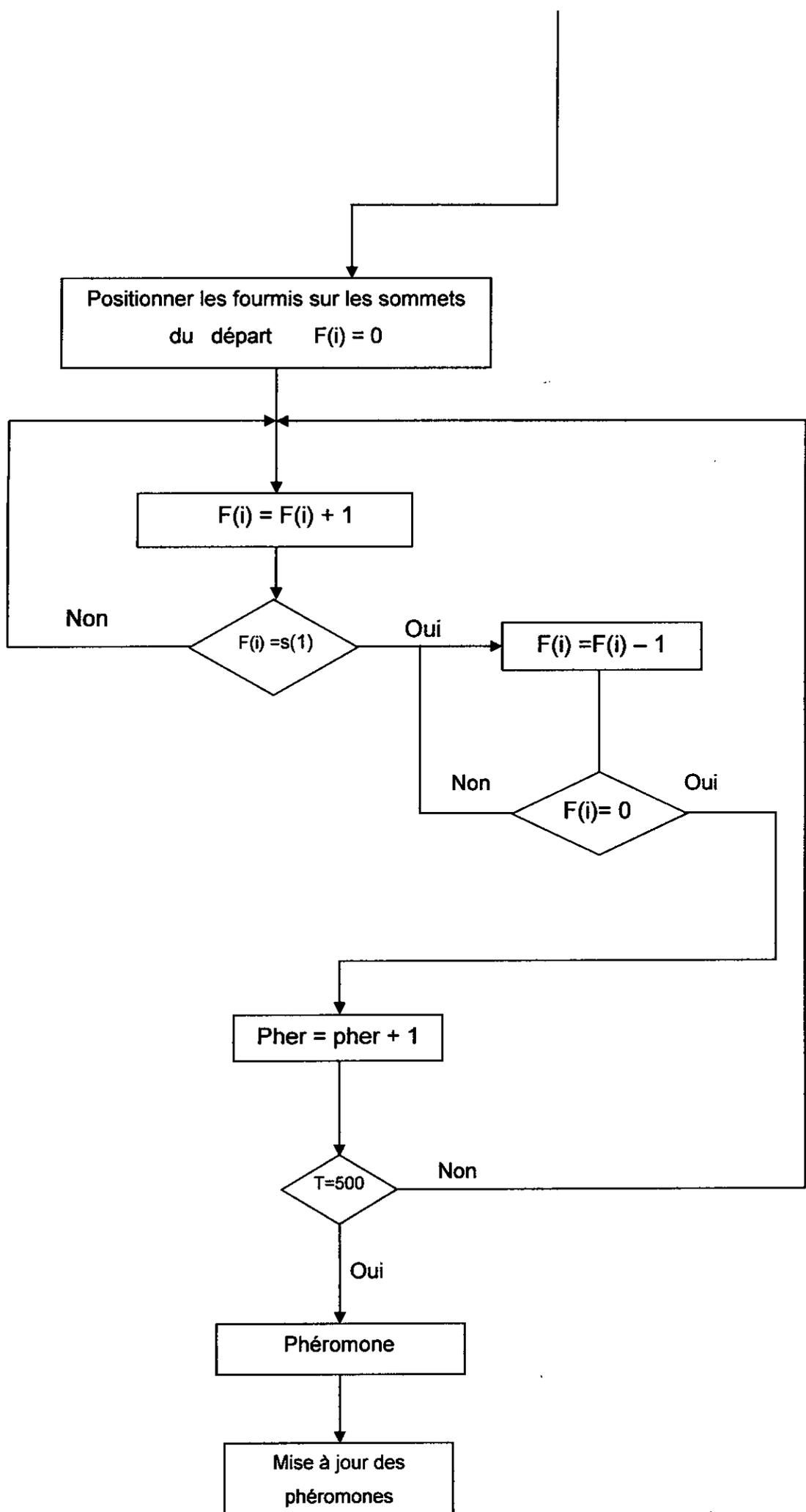
Conclusion :

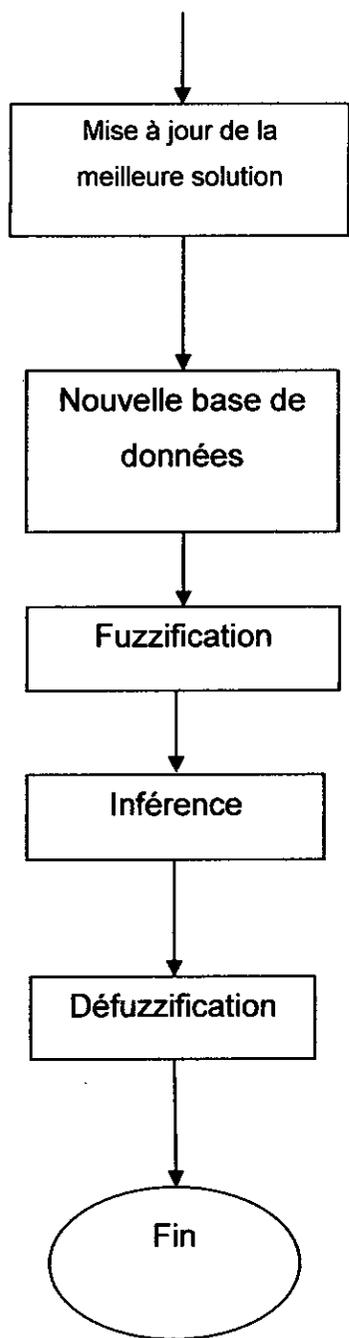
Les résultats de la première synthèse des deux contrôleurs flous montrent que le dirigeable avance avec des dépassements assez importants, suivi de petites oscillations. Ceci nous incite à penser qu'une optimisation du rendement de nos contrôleurs s'impose. En effet, la base de connaissances des contrôleurs flous, constituées par un ensemble de règles et leurs données relatives traduisant une stratégie de contrôle basé sur les informations fournies par un expert, demeure un problème important dans la conception des contrôleurs.

L'influence de l'écriture des règles de contrôle en pratique est très grande. On est en droit de se demander si telle base de connaissances, optimise le rendement du contrôleur flou ? N'y aurait-il pas une base de connaissances qui optimiserait d'avantage ce rendement d'une façon sensible ?

Pour répondre a cette question, une écriture d'une meilleure base de connaissances est introduite par l'accès d'un contrôleur hybride basé sur la méthodologie COR et l'approche BWAS.

4.5. Organigramme de l'approche COR - BWAS :





Explication de l'organigramme :

L'organigramme du système contient deux parties importantes :

- La première partie contient les différentes étapes de la méthodologie des règles coopératives COR qui permet de générer plusieurs règles floues ensuite les sélectionner par une fonction qui s'appelle MSE (Erreur Quadratique moyenne) qui permet de tirer la meilleure solution avec un MSE très faible dans chaque sous-espace de recherche .Le MSE est le calcul de la différence entre le signal de référence et le signal de commande issu du système basé sur les règles floues (FRBS) .
- La deuxième partie de l'organigramme est réservée à l'approche d'optimisation BWAS (Best Worst Ant System). L'algorithme permet de tirer la meilleure des mauvaises solutions avec une forte concentration de phéromone, d'où la solution optimale.

Avec :

- ✓ $i = i+1$ (c'est une incrémentation qui génère un certain nombre de solution choisi aléatoirement ex :50)

Si on atteint 50 solutions on crée une matrice de fourmis $F(i)$.

- ✓ $F(i)=F(i)+1$ (Déplacement des fourmis)
- ✓ $F(i)=S(i)$ (Atteindre la solution : Nourriture)
- ✓ $F(i)=F(i)-1$ (Retour au nid)
- ✓ Pher : Matrice de concentration de phéromones.
- ✓ $Pher=Pher+1$: (Trace de phéromone laissée par la fourmis)
- ✓ $T=500$ (Période : on a choisit 500 cycles d'exécution)

C'est une Période de foncer les pistes de phéromones.

Donc la meilleure solution est la solution de plus forte concentration de phéromone .

- ✓ Mise à jour de phéromone : Matrice qui contient pour chaque piste sa concentration de phéromone après élimination des mauvaises solutions par le mécanisme d'évaporation de phéromone.
- ✓ Mise à jour de la meilleure solution : c'est-à-dire on choisit le chemin de plus forte concentration qui est le chemin de la solution optimale

et pour cette dernière on tire la nouvelle base de données ou la base de données idéale.

4.6. Résultats de la simulation d'un contrôleur flou optimisé

- Après 7 itérations

✓ Base des règles (deuxième expertise) :

$\Delta e/e$	pb	pm	ps	ze	nb	nm	ns
pb	nm	nm	nm	nm	ze	ns	nm
pm	nm	nm	nm	nm	ps	ze	ns
ps	nm	nm	nm	ns	Don't care	ps	ze
ze	ns	ns	ns	ze	Don't care	Don't care	ps
nb	ze	ps	Don't care	pb	pb	pb	pb
nm	ns	ze	ps	Don't care	pb	pb	pb
ns	ns	ns	ze	ps	pb	pb	Don't care

43 règles (Réduction du nombre de règles)

Figure 33 : Nouvelle base de règles

✓ Base de données :

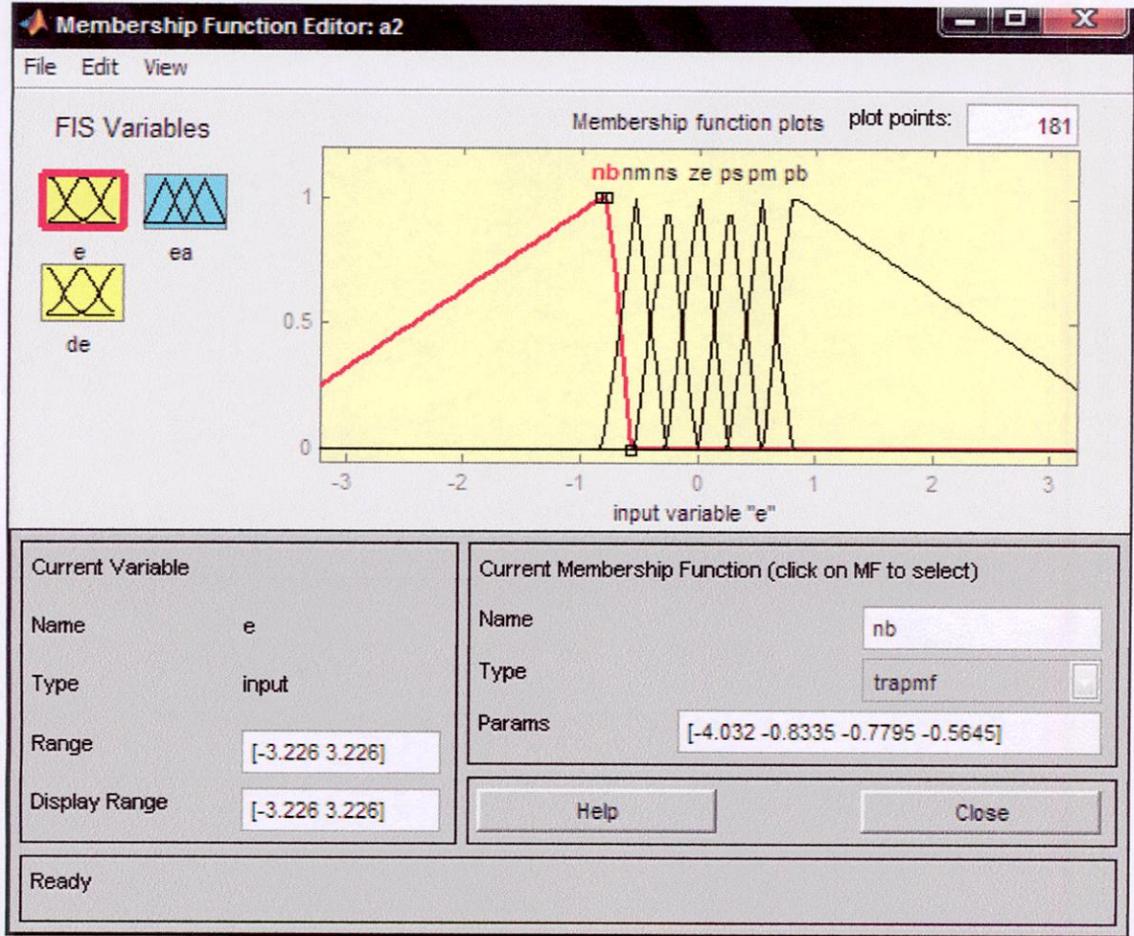


Figure 34: Fonctions d'appartenance entrées e

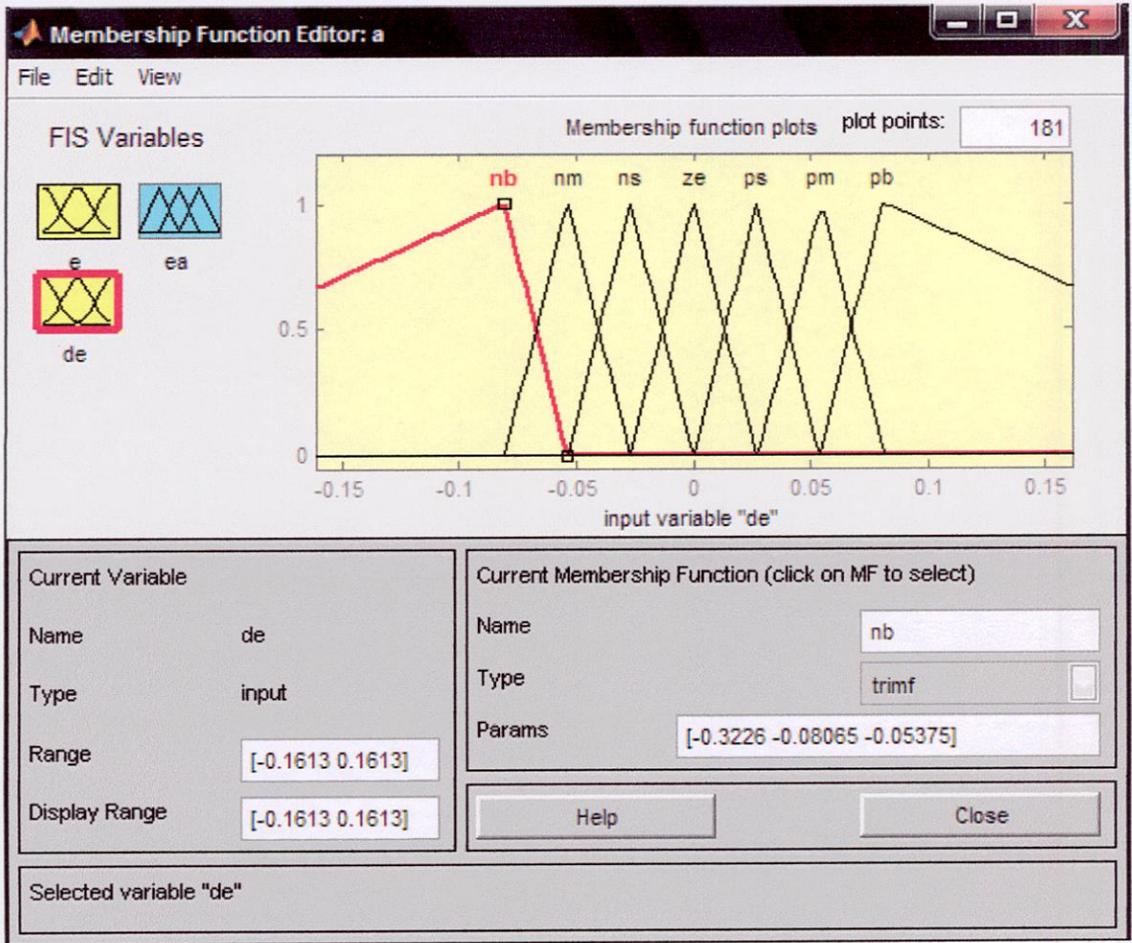


Figure 35: Fonctions d'appartenances variation d'erreur Δe

✓ Sortie de la commande :

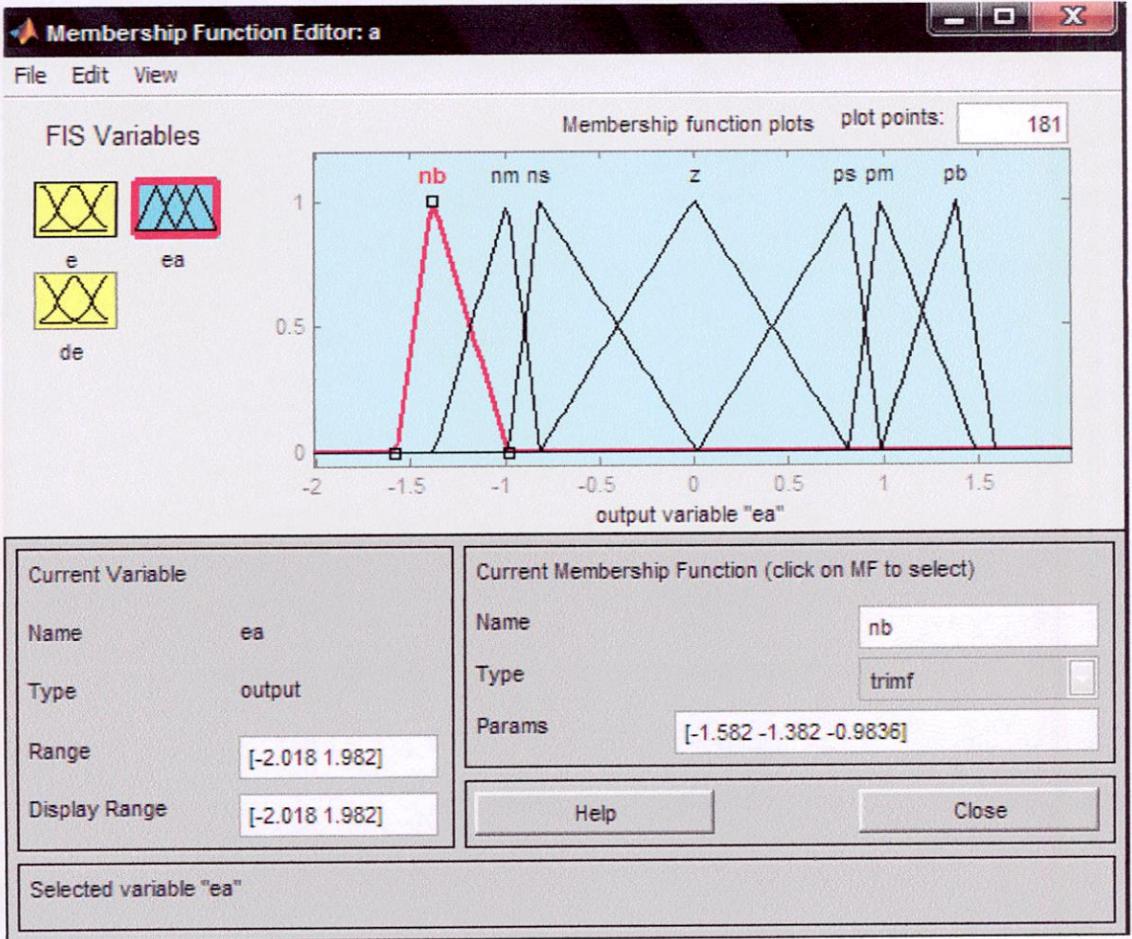


Figure 36: Sortie de la commande floue (deuxième contrôleur flou)

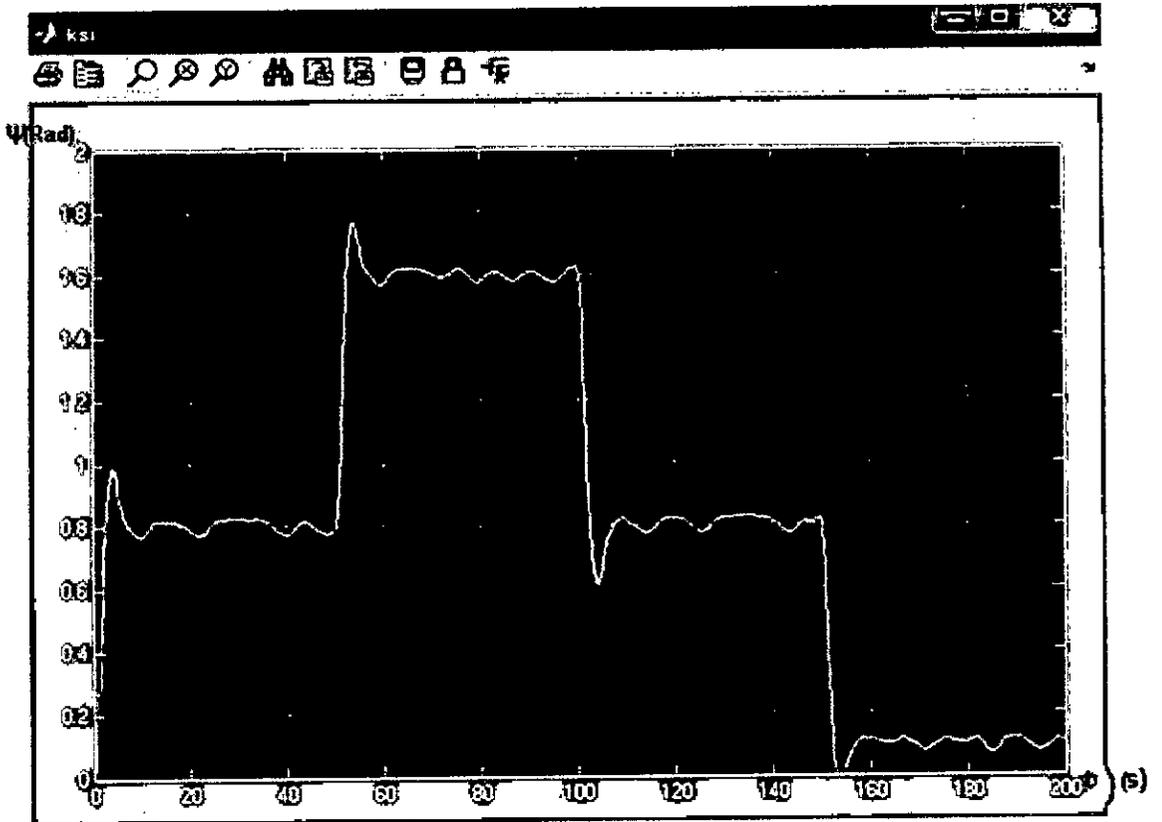


Figure 37 :Signal de sortie du processus

On remarque qu'après 7 itérations, il y a une petite amélioration avec un léger dépassement suivi par des petites oscillations.

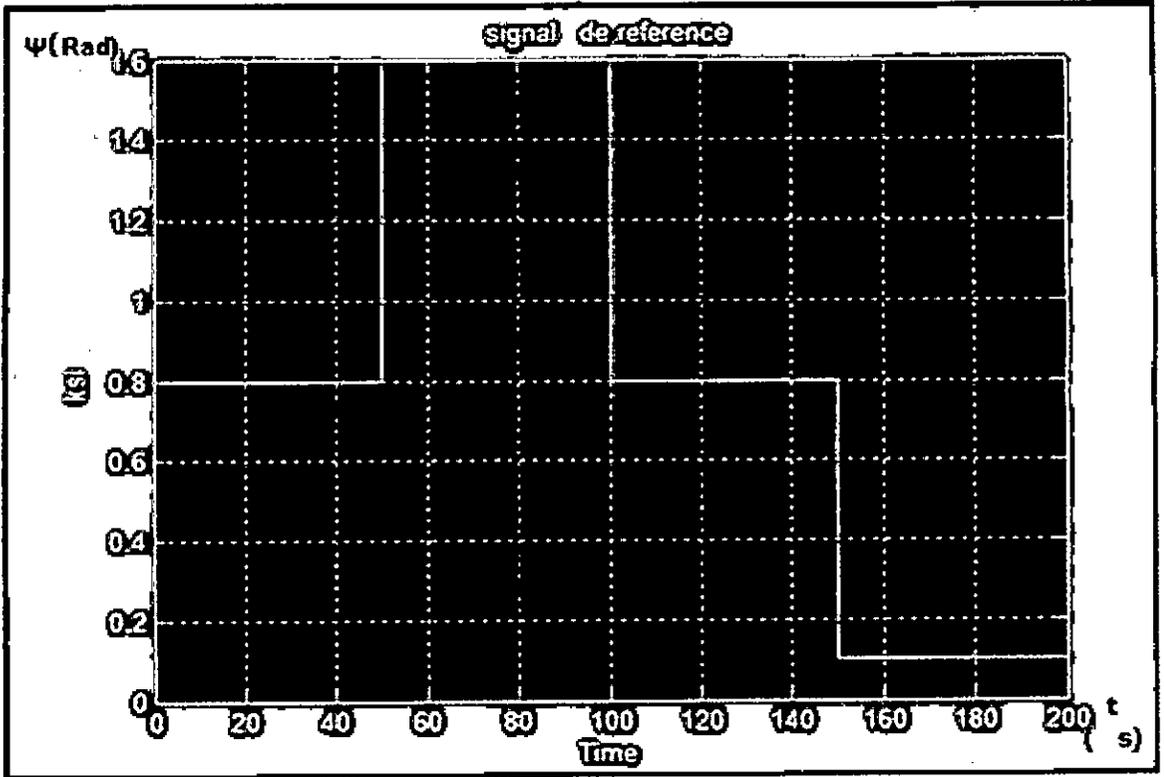


Figure 38 :Signal de référence

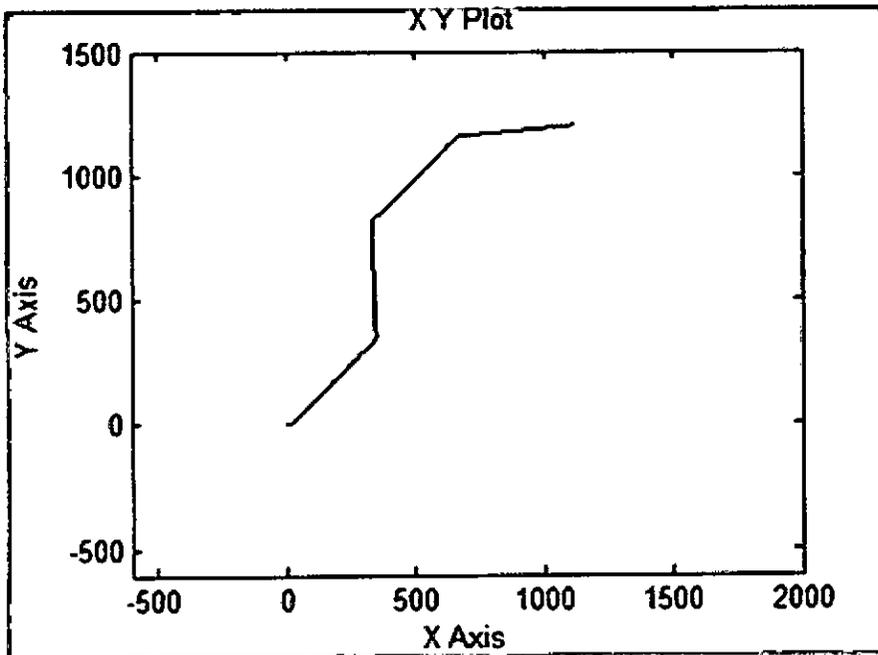


Figure 39: Suivi de la trajectoire

- Après 50 itérations :
 - ✓ Base des règles (troisième expertise)

$\Delta e \backslash e$	pb	pm	ps	ze	nb	nm	ns
pb	ns	ns	ns	ns	Don't care	ns	ns
pm	ns	ns	ns	ns	ps	Don't care	ns
ps	ns	ns	ns	ns	Don't care	ps	Don't care
ze	ns	ns	ns	Don't care	Don't care	Don't care	ps
nb	Don't care	ps	Don't care	pb	pb	pb	pb
nm	ns	Don't care	ps	Don't care	pb	pb	pb
ns	ns	ns	Don't care	ps	pb	pb	Don't care

36 règles (Réduction du nombre de règles)

Figure 40 : Base des règles

- ✓ Base de données :

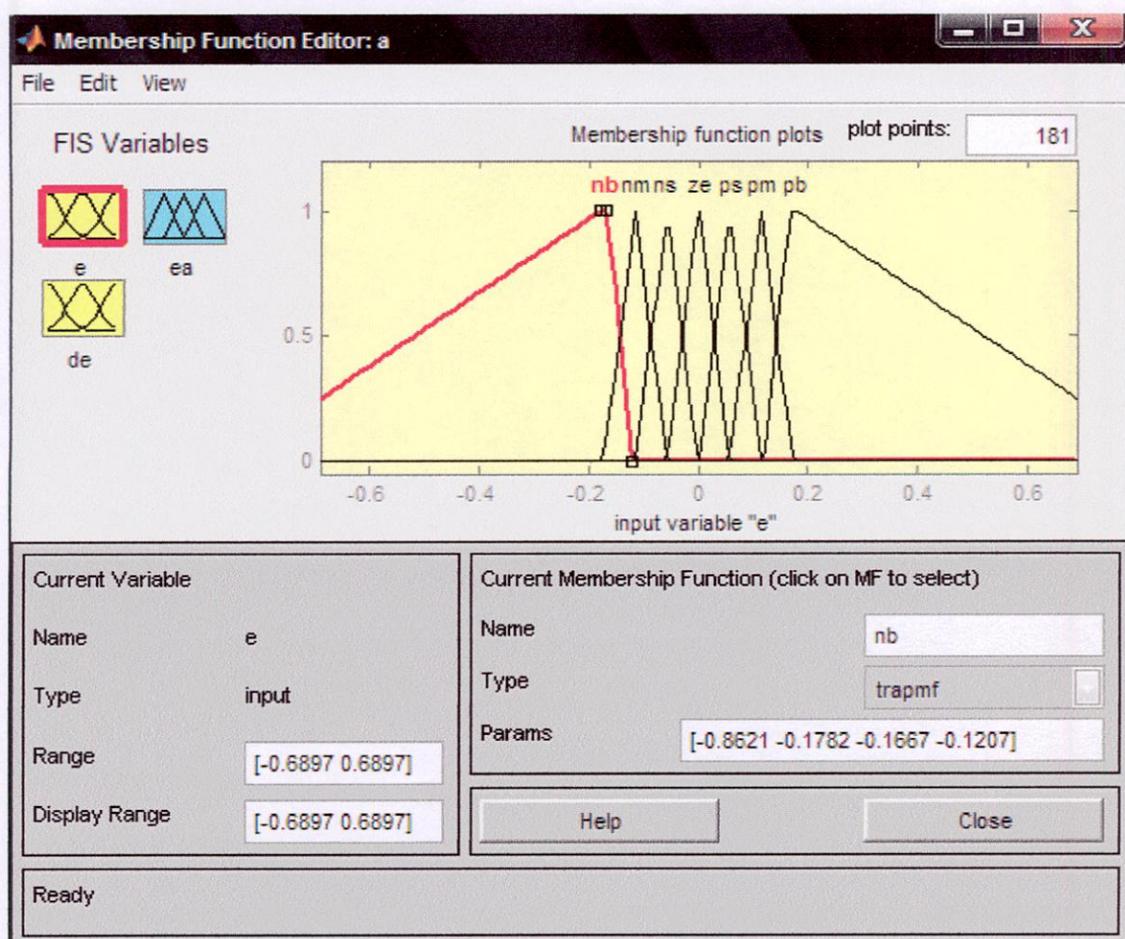


Figure 41: Fonctions d'appartenance entrée e

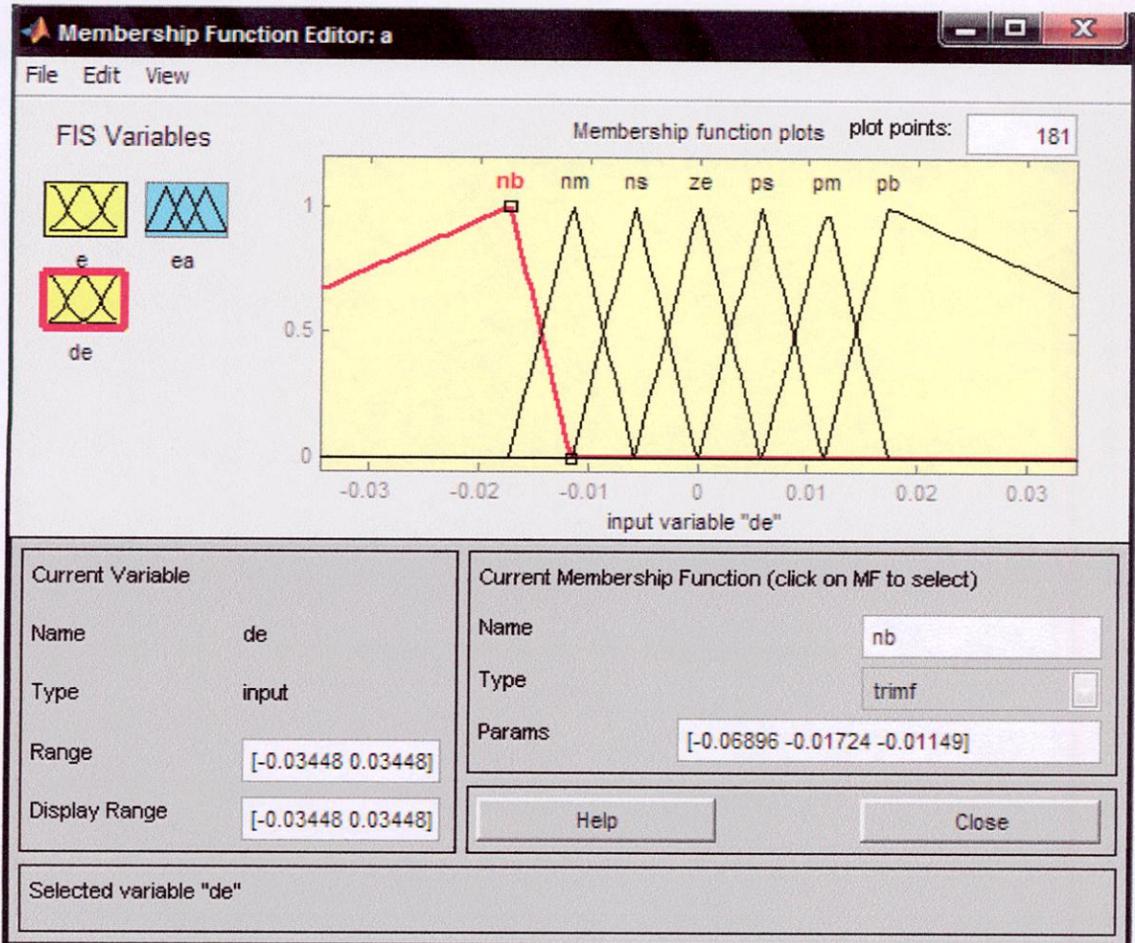


Figure 42: Fonctions d'appartenances variation d'erreurs Δe

✓ Sortie de la commande :

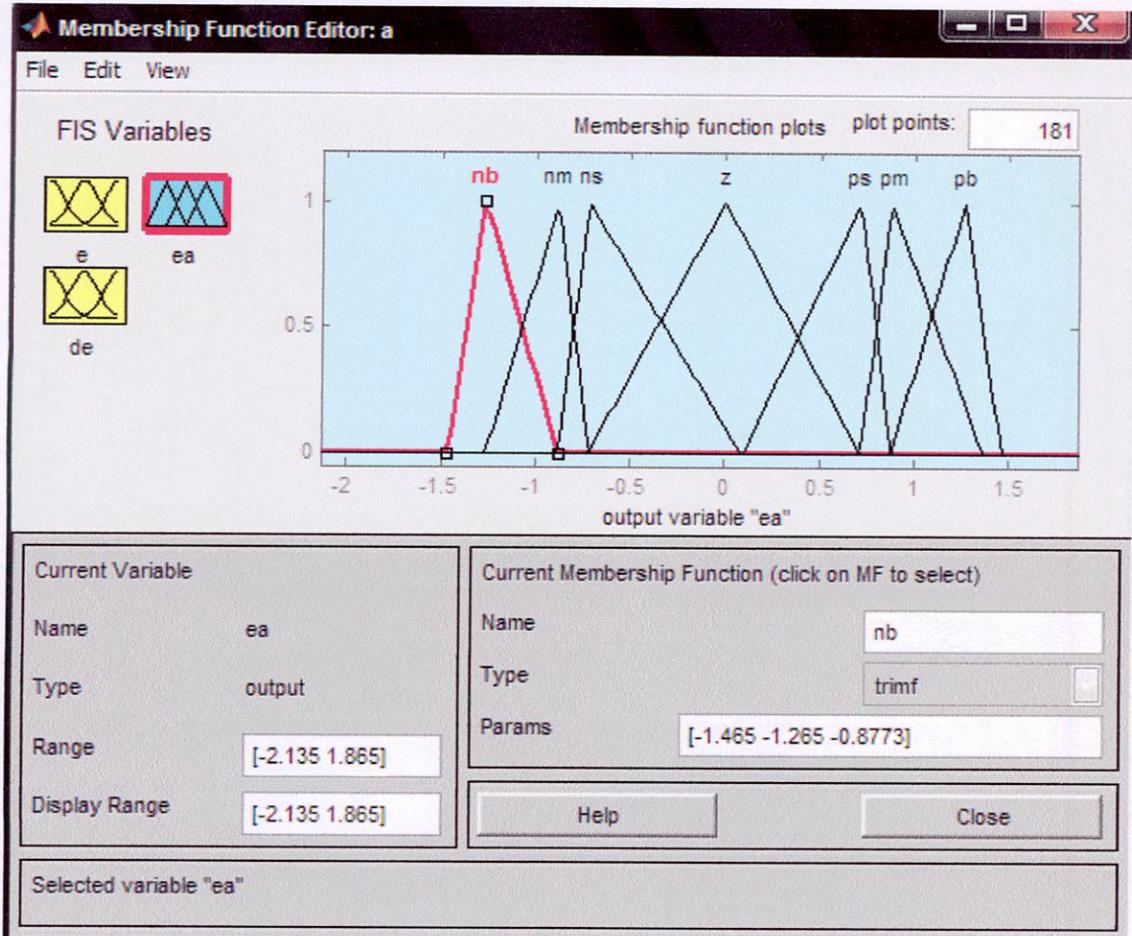


Figure 43: Sortie du contrôleur flou



Figure 44: Sortie du signal optimal du système

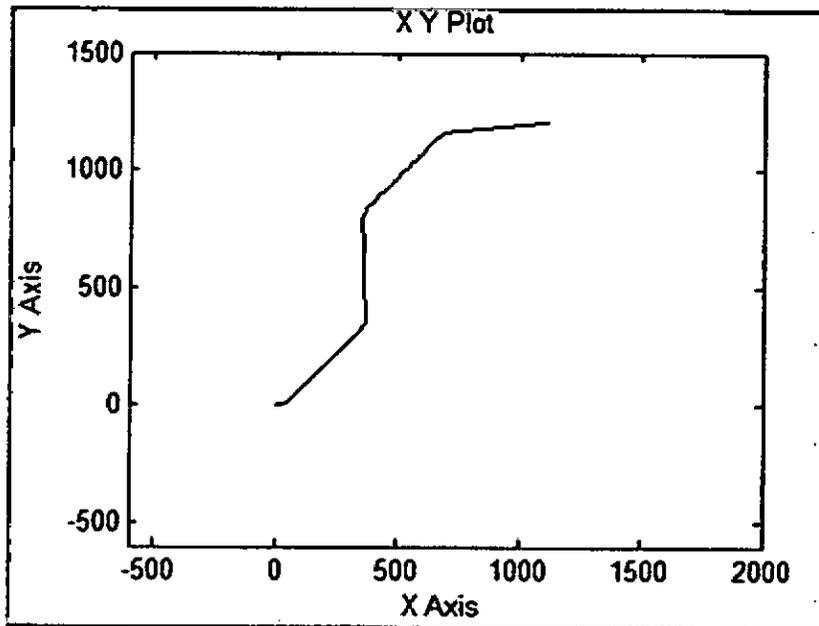


Figure 45: Suivi de la trajectoire

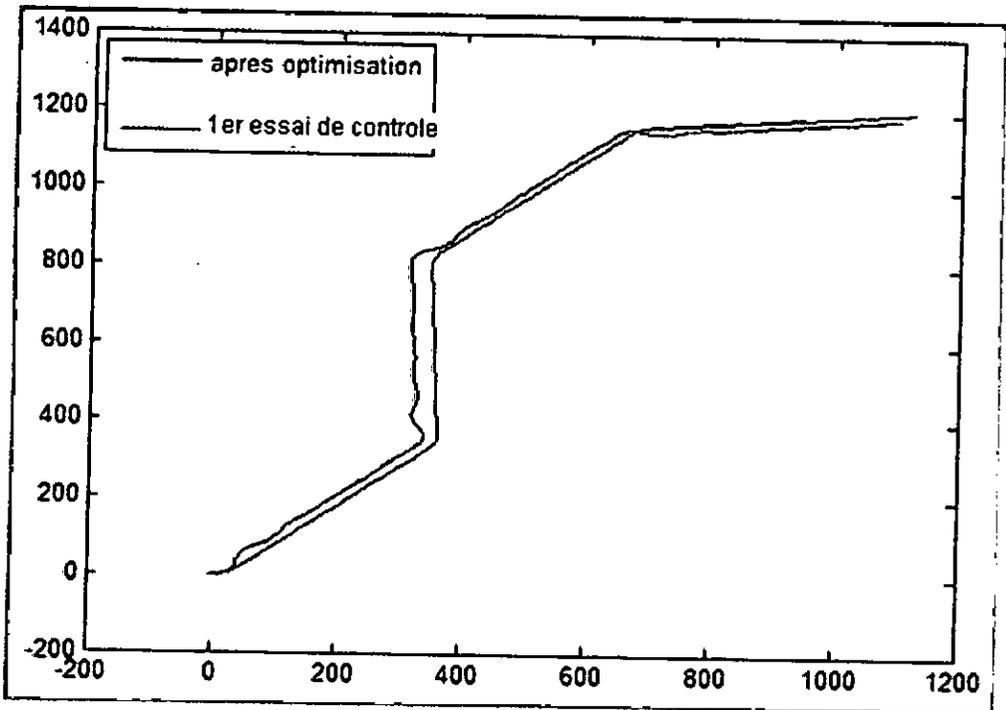


Figure 46 : Superposition des suivis de la trajectoire .

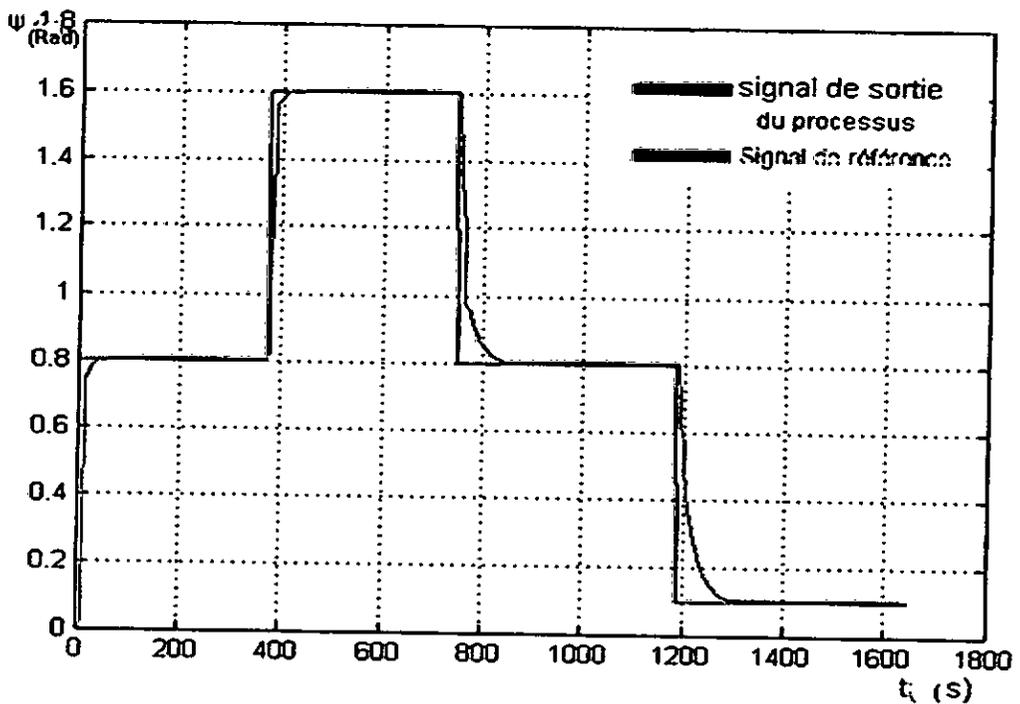


Figure 47: Superposition des deux signaux (sortie du processus optimisée et de référence)

On remarque dans cette figure, l'absence des petites oscillations en régime établi et des dépassements en régime transitoire. Ceci nous incite à dire que dans

cette deuxième partie du travail, où les bases de connaissances des deux contrôleurs, ont été déduites grâce à l'approche (COR-BWAS), les résultats obtenus sont nettement meilleurs par rapport à ceux obtenus dans le premier essai et le rendement des contrôleurs s'est amélioré très sensiblement. Donc, l'approche en question constitue une solution très acceptable au problème d'écriture des bases de connaissances des systèmes flous.

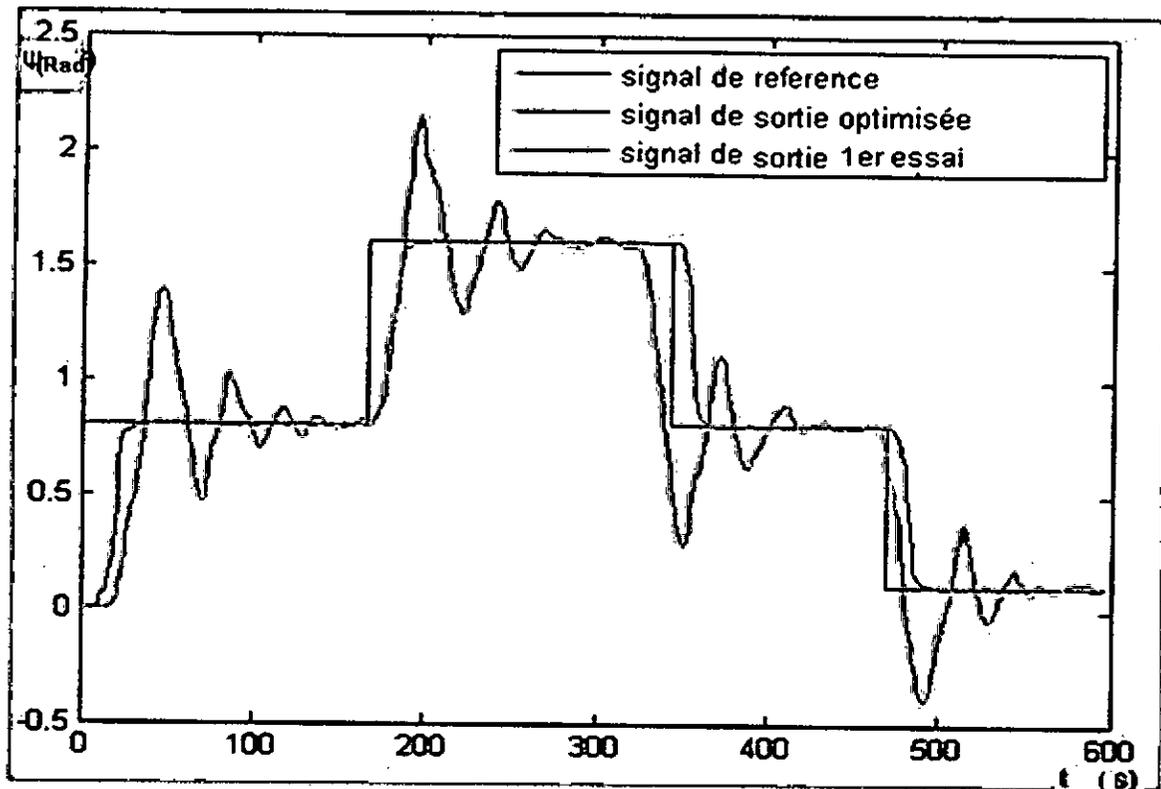


Figure 48 : Superposition des trois signaux (Référence –sortie du processus (1^{er} essai) –sortie du processus optimisée)

CONCLUSION

Dans les systèmes basés sur les règles floues (**FRBS**), la tâche de la définition automatique de la base des connaissances (base des règles et base de données) pour une application concrète, est considérée comme un problème dur.

La multitude de méthodes, efficaces et simples, proposées en littérature pèchent par le fait qu'elles recherchent habituellement, les règles floues possédant les meilleures performances individuelles, et que **l'interaction globale** entre ces règles floues de la base n'est **jamais prise en considération**. Ceci engendre des bases de règles floues possédant une **mauvaise coopération** (entre les règles qui les composent), ce qui entraîne une diminution sensible de la **précision** du système.

Pour remédier à ces inconvénients, on a présenté dans ce travail, une méthodologie des règles coopératives (**COR**) dans le but d'avoir **une amélioration sensible de la précision du système, avec une meilleure coopération entre les règles**. En effet, la méthodologie **COR**, nous donne la possibilité d'utiliser des conséquents des règles qui sont différents des meilleurs, d'où un ensemble de règles floues avec **une meilleure coopération** et une **plus grande exactitude** du système.

En outre, dans cette contribution, on s'est servi d'un algorithme spécifique d'optimisation par colonie de fourmis, appelé **BWAS : Best-Worst-Ant-System** (L'algorithme du meilleur mauvais système de fourmis), pour sa bonne efficacité pour la résolution d'un grand nombre de problèmes d'optimisation dans la littérature [1] [2]. Cet algorithme nous a aidé à trouver un excellent équilibre entre **l'interprétabilité, l'exactitude des systèmes flous obtenus et la rapidité de l'apprentissage des règles**.

L'approche ainsi définie (**COR-BWAS**), a été appliquée au problème de dérivation des bases des connaissances optimales (bases des règles et bases de données) de deux contrôleurs flous pour la conduite d'un ballon dirigeable par la vision.

Dans ce contexte, pour un premier essai et une première synthèse de nos deux contrôleurs flous, les bases de connaissances ont été écrites en suivant les directives d'un expert dans le domaine et les résultats obtenus montrent la présence de petites oscillations en régime établi et des dépassements assez importants en régime transitoire. Ceci nous a incité à penser que la stratégie de

contrôle basée sur les informations fournies par l'expert est traduite par les deux bases de connaissances des contrôleurs, n'était pas optimale.

Dans la deuxième partie de notre travail, les deux bases de connaissances des deux contrôleurs, ont été déduites grâce à l'approche **(COR-BWAS)** et les résultats obtenus sont nettement meilleurs par rapport à ceux obtenus dans le premier essai, ce qui nous fait dire que le rendement des contrôleurs s'est amélioré très sensiblement et que l'approche en question constitue une solution très acceptable au problème d'écriture des bases de connaissances des systèmes flous.

Les perspectives de ce travail sont relativement nombreuses, tant du point de vue théorique que pratique. On pourrait citer, par exemple :

- L'utilisation de l'approche **(COR-BWAS)** dans le cadre générale des problèmes d'optimisation difficile.
- Envisager une approche **(COR-API)** pour les problèmes d'optimisation combinatoire (très nombreux en sciences de l'ingénieur).
- On peut envisager aussi des adaptations de l'algorithme de fourmis BWAS aux problèmes d'optimisation avec contraintes, à l'optimisation multicritère et l'optimisation dynamiques.

Remarque : **API** (algorithme, déduit de la modélisation du comportement de fourragement d'une population de fourmis primitives (*Pachycondyla APIcalis*)).

APPENDICE A
LISTE DES SYMBOLES ET DES ABREVIATIONS

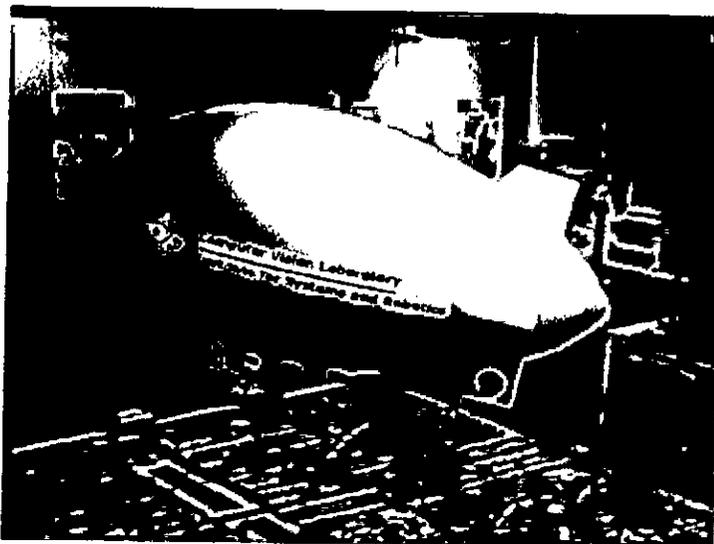
η	: Vecteur de configuration
η_1	: Vecteur des positions
η_2	: Vecteur d'orientations
v	: Vecteur vitesses
v_l	: Vecteur de vitesses linéaires
v_a	: Vecteur de vitesses angulaires
ϕ	: Angle de roulis
θ	: Angle de tangage
ψ	: Angle de lacet
ω	: Vecteur de vitesses angulaires
ω_x	: Vitesse angulaire autour de l'axe x
ω_y	: Vitesse angulaire autour de l'axe y
ω_z	: Vitesse angulaire autour de l'axe z
R	: Matrice de rotation
J_1	: Matrice de transformation des vitesses linéaires
J_2	: Matrice de transformation des vitesses angulaires
J	: Matrice de transformation
r_G	: Distance du centre de gravité
v_o	: Vecteur vitesses linéaires du centre du repère
v_G	: Vecteur vitesses linéaires du centre de gravité
m	: Masse
a_o	: Vecteur accélérations du centre du repère
a_G	: Vecteur accélérations du centre de gravité
H	: Moment cinétique
f_o	: Forces externes
M_o	: Moments externes
I_o	: Matrice d'inertie
M_{ED}	: Matrice d'inertie du système
C_{KB}	: Matrice des forces de Coriolis et des forces centrifuges
u	: Vitesse locale du fluide
n	: Vecteur normal à la surface du dirigeable orienté vers le fluide
ϕ	: Potentiel des vitesses
ρ_o	: La densité du fluide
M_a	: Matrice de masses ajoutées.
T_f	: Energie cinétique du fluide
T_a	: Energie cinétique des efforts de la masse ajoutée

- F_a : Force
 n_a : Couple
 Γ_a : Couple induit par l'inertie et la masse ajoutées.
 C_a : Matrice de force de Coriolis et de forces de centrifuges hydrodynamique
 D : Matrice des forces et les couples appliquées au corps par l'environnement
 D_f : Force d'amortissement due à la forme du dirigeable
 τ_D : Force d'amortissement
 C_D : Coefficient de traînée
 A : Surface du corps en incidence
 V_f : Volume du fluide déplacé
 F_B : Poussée d'Archimède
 F_G : Force de pesanteur
 W : Force de pesanteur
 B : Poussée d'Archimède
 M : Matrice 6x6 contenant les termes de masse et d'inertie
 C : Matrice 6x6 des termes de forces de Coriolis et de centrifuges
 D : Matrice contenant des termes de forces d'amortissements
 g : Matrice contenant les termes des forces hydrostatiques
 τ : Forces et couples des propulseurs

APPENDICE B

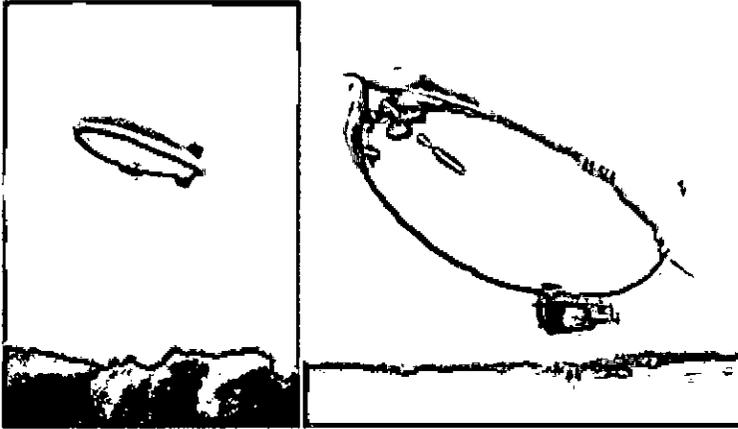
Les différents ballons dirigeables utilisés en laboratoire :

Les ballons d'intérieur :





Les ballons d'extérieur :



REFERENCES

1. Jorge casilas , Oscar Cardon , Inaki Fernandez de viana ,Fransisco Herrera «Learning Cooperative Linguistic Fuzzy rules Using the Best-Worst Ant System Algorithm » Department of computer science and artificial intelligence , University Of Granada , E-18071 Granada , Spain
2. Casillas J, Cordón O, Herrera F, Magdalena L, editors. Interpretability issues in fuzzy modeling. Heidelberg: Springer; 2003.
3. Jorge casilas , Oscar cardon , Fransisco Herrera “COR :methodology to improve ad hoc data driven linguistic rule learning methods by inducing cooperation among rules “.IEEE transactions on systems , MAN,and cybernetics –part B cybernetics , Vol 32 , No 4 , August 2002 .
4. Casillas J, Cordón O, Herrera F, Magdalena L, editors. Accuracy improvements in linguistic fuzzy modeling. Heidelberg: Springer; 2003.
5. Cordón O, Herrera F. Athree-stage evolutionary process for learning descriptive and approximate fuzzy logic controller knowledge bases from examples. *Int J Approx Reason* 1997; 17:369– 407.
6. Dorigo M, Stützle T. Ant colony optimization. Cambridge, MA: The MIT Press; 2004.
7. Cordón O, Fernández de Viana I, Herrera F. Analysis of the best–worst ant system and its variants on the TSP. *Mathware Soft Comput* 2002;9:177–192.
8. Cordón O, Fernández de Viana I, Herrera F. Analysis of the best–worst ant system and its variants on the QAP. In Dorigo M, Di Caro G, Sampels M, editors. *Ant algorithms*. LNCS 2463. Heidelberg: Springer-Verlag; 2002. pp 228–234.

9. Wang L-X, Mendel JM. Generating fuzzy rules by learning from examples. *IEEE Trans Syst Man Cybern* 1992;22:1414–1427.
10. Wei WU «Synthèse d'un contrôleur flou par Algorithme Génétique :Application au réglage dynamique des paramètres d'un système»thèse doctorat de l'Université de Lille 1,1998
11. M.Hadj sadok <<Méthodologie de contrôle basée sur la logique flou» Thèse de magister Université de Blida Algérie 1995
12. L.A.ZADEH. A Theory of approximate reasoning. In J.E.Hayes,D Michie and L.I.Mikulich,ed.Machine Intelligence,Vol 9.,pp.149 194.Elsevier,Amsterdam,1979.
13. A. Bárdossy and L. Duckstein, *Fuzzy Rule-Based Modeling With Application to Geophysical, Biological and Engineering Systems*. Boca Raton, FL: CRC, 1995.
14. W. Pedrycz, Ed., *Fuzzy Modeling: Paradigms and Practice*. Norwell, MA: Kluwer, 1996.
15. L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning," *Inform. Sci.*, vol. 8, pp. 301–357, 1975.
16. L.A.ZADEH, PRUF-A Meaning Representation language for Natural Languages , Int.J.Man-Machine Studies 10 ,pp.395-460,1978.
17. Guenounou, O., A. Belmehdi et B. Dahhou (2006). Optimisation des contrôleurs flous par algorithmes génétiques hiérarchisés-application à la commande d'un bioprocédé.In: *Seventh International Conference on Sciences and Techniques of Automatic control STA 2006*. Hammamet,Tunisia. pp. 1–10.
18. Dorigo, M. & Stützle, T., 2004. *Ant Colony optimization*. MIT Press Deneubourg, J. L., Goss, S., Franks, N., Detrain, C., Chretien, L.," The dynamics of collective sorting: robot-like ant and ant-like robots." In Meyer and Wilson, (1990), 356–365.

19. Gutjahr, W. J., "A graph-based Ant System and its convergence." *Future Generation Computer Systems*, tome 16, No 8, (2000), 873-888.
20. Lee, M. A. et H. Takagi (1993). Dynamic control of genetic algorithms using fuzzy logic techniques. In: *Proceedings of International conference on Genetic Algorithms*. San Mateo, CA. pp. 76–83.
21. Colomi, A., Dorigo, M. & Maniezzo, V., "An investigation of some properties of an ant algorithm." In (Männer and Manderick) ,(1992), 509–520.
22. Colomi, A., "Distributed optimization by ant colonies", *Proc. 1st European Conf. on Artificial Life*, Paris, Elsevier, (1991), 134-139.
23. Dorigo, M., Maniezzo, V. & Colomi, A., "The Ant System: optimization by a Colony of Cooperating Agents." *IEEE Trans. Syst. Man Cybern*, tome B, No 26,(1996), 29-41.
24. Gambardella, L.M., Dorigo, M., "Solving symmetric and asymmetric TSPs by ant colonies", *Proc. IEEE Conf. on Evolut. Computa. ICEC*, New Jersey, (1996), 622-627.
25. Yubazaki, N., M. Otani, T. Ashida et K. Hirota (1995). Dynamic fuzzy control method and its application to positioning of induction motor. In: *Proceedings of Fourth IEEE International Conference on Fuzzy Systems*. Japan. pp. 1095–1102 ;
26. M.Hadj sadok « Api, une métaheuristique pour l'optimisation difficile. Application à l'identification de la machine asynchrone » thèse de doctorat université de Blida, Algerie ,Dec2007.
27. Stützle, T., "Parallelisation Strategies for Ant Colony Optimization." In (Eiben et al., 1998a).

28. Maniezzo, V., Colorni, A., and Dorigo, M., "The Ant System Applied to the Quadratic Assignment Problem." Technical Report 94-28, IRIDIA, universit  Libre de Bruxelles, Belgium, (1994).
29. Gambardella, L.M., Taillard, E.D., & Dorigo, M., "Ant colonies for the quadratic assignment problem", *Journal of the Operational Research Society*, Vol. 50, No. 2, (Feb. 1999), 167-176.
30. Dorigo, M., St tztle, T., "Handbook of Metaheuristics", tome 57 de International series in operations research and management science, chapitre "The ant Colony Optimization Metaheuristics : Algorithms, Applications and Advances." Kluwer Academic Publishers, Boston Hardbound, (2003).
31. Costa, D., Hertz, A., "Ants can color graphs", *Journal of the Operational Research Society*, Vol. 48, No. 3, (Mar. 1997), 295-305.
32. Maniezzo, V., Carbonaro, A., "An ANTS heuristic for the frequency assignment problem." *Future Generation Computer Systems*, 16(8), (2000), 927-935.
33. Taillard, E., "FANT : Fast Ant System." Technical Report 46-98, IDSIA, Lugano, (1998).
34. Baluja S, Caruana R, Removing the genetics from standard genetic algorithm .In: Prieditis A, Russell S , Editors .Proc Int Conf on machine learning .San Mateo, CA : Morgan Kauf - mann ; 1995 .PP 38-46.
35. Zadeh LA. Concept of a linguistic variable and its application to approximate reasoning, 1, 2, and 3. *Inform Sci* 1975;8:199-249, 301-357; 1976;9:43-80
36. Dorigo M, St tztle T. The ant colony optimization metaheuristic: Algorithms, applications, and advances. In: Glover F, Kochenberger GA, editors. *Handbook of metaheuristics*. Norwell, MA: Kluwer Academic; 2003. pp 251-285..
37. Sugeno M, Yasukawa T. A fuzzy-logic-based approach to qualitative modeling. *IEEE Trans Fuzzy Syst* 1993;1(1):7-31.

38. Krone A, Krause P, Slawinski T. A new rule reduction method for finding interpretable and small rule bases in high dimensional search spaces. In: Proc Ninth IEEE Int Conf on Fuzzy Systems, San Antonio, TX, 2000. pp 693– 699.
39. Cordón O, Herrera F, Villar P. Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base. *IEEE Trans Fuzzy Syst* 2001;9:667– 674.
40. Glover F, Kochenberger GA, editors. *Handbook of metaheuristics*. Norwell, MA: Kluwer Academic; 2003.
41. Dorigo M, Stützle T. The ant colony optimization metaheuristic: Algorithms, applications, and advances. In: Glover F, Kochenberger GA, editors. *Handbook of metaheuristics*. Norwell, MA: Kluwer Academic; 2003. pp 251–285.
42. Cordón O, Herrera F, Stützle T. A review on the ant colony optimization metaheuristic: Basis, models and new trends. *Mathware Soft Comput* 2002;9:141– 175.
43. Dorigo M, Di Caro G. The ant colony optimization meta-heuristic. In: Corne D, Dorigo M, Glover F, editors. *New ideas in optimization*. New York: McGraw-Hill; 1999. pp 11–32.
44. Cordón O, Herrera F, Fernández de Viana I, Moreno L. A new ACO model integrating evolutionary computation concepts: The best–worst ant system. In: Proc Second IntWorkshop on Ant Algorithms, Brussels, Belgium, 2000. pp 22–29.
45. Dorigo M, Gambardella LM. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans Evol Comput* 1997;1:53– 66.
46. Dorigo M, Gambardella LM. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans Evol Comput* 1997;1:53– 66.

47. Dorigo M, Stützle T. Ant colony optimization. Cambridge, MA: The MIT Press; 2004.
48. Vincent Gardeux « Conception d'heuristiques d'optimisation pour les problèmes de grande dimension application à l'analyse de données de puces à ADN » thèse de doctorat ,université de Paris-est-creteil, 2011.
49. Nozaki K, Ishibuchi H, Tanaka H. A simple but powerful heuristic method for generating fuzzy rules from numerical data. *Fuzzy Set Syst* 1997;86:251–270.
50. Cordón O, Fernández de Viana I, Herrera F. Analysis of the best–worst ant system and its variants on the TSP. *Mathware Soft Comput* 2002;9:177–192..
51. Cordón O, Fernández de Viana I, Herrera F. Analysis of the best–worst ant system and its variants on the TSP. *Mathware Soft Comput* 2002;9:177–192.
52. Stützle T, Hoos H. Max-min ant system. *Future Generat Comput Syst* 2000;8:889–
53. Baluja S. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical Report CMU-CS-94- 163. Pittsburgh, PA: Carnegie Mellon University; 1994.
54. Filipe Manuel da Silva Metelo, Luís Ricardo Garcia Campos« Vision based control an autonomous blimp (vidéoblímp) ».Thèse ; Université Technique de Lisbonne 2003.
55. Technique de l'ingénieur A1661« Mécanique générale, cinématique générale »par Jean-Pierre BROSSARD ;Professeur de Mécanique à l'Institut National des Sciences Appliquées (INSA) de Lyon.
56. Fossen, Thor I., «Guidance and Control Of Ocean Vehicles, John Wiley & Sons, 1994.
57. Jean- François Lots «Application of visual servoing to the dynamique positioning of underwater vehicle»thesis for the degree of de doctor Heriot-watt university.

58. Tomas SALGADO JIMENEZ <<Contribution a la commande d'un robot sous-marin autonome de type torpille>>.Thèse de doctorat , Université de Montpellier II France .2004.
59. Jonwoo.Kim Jim Keller Vijay Kumar. «Design and verification of controllers for airship» Proceedings of the 2003 IEEE/RJS.Int.Conference on Intelligent Robots and systems.
60. Emmanuel Hygounec, Il-Kyun Jung, Philippe Soueres, Simon Lacroix «The Autonomous Blimp Project of LAAS-CNRS:Achievements in Flight Control and Terrain mapping». LAAS/CNRS.France 2004.