

*République Algérienne Démocratique Et Populaire*

*Ministère De L'Enseignement Supérieur*

*Et De La Recherche Scientifique*

*Université Saad Dahlab*

*Faculté De Sciences*

*Département De Mathématiques*

*Mémoire de Fin d'Etudes en vue de l'obtention du diplôme d'ingénieur  
d'état en Mathématiques Appliquées*

*Option : Recherche Opérationnelle*

*Thème*

*La Logique floue avec Optimisation  
par Colonies de Fourmis  
appliquée à la Commande d'Attitude  
d'un Satellite*

*Présenté par*

*Mr : BENNOUR Mahfoud*

*Mr: HADJAMAR Farid*

*Promoteur*

*M. A. BENCHERHALI*

*Promotion 2010/2011*

MA-510-08-1

## REMERCIEMENTS

*Sans l'aide et la grâce de DIEU, ce travail n'aurait jamais vu son terme. Le mot « remercier » n'à pas l'amplitude voulu pour lui exprimer notre GRATITUDE.*

*Nous remercions Monsieur M. A. BENCHERCHALI, notre promoteur pour son encadrement, sa rigueur, son suivi, ses précieux conseils portés à notre égard sans jamais oublier l'objectif rationnel de notre travail.*

*Nous tenons à remercier le jury d'avoir accepté d'être ses membres de juger nous n'oublierons pas les personnes qui nous sont chères, ainsi que ceux et celles qui ont été nos familles d'exception pour leurs soutiens explicites ou implicites.*

*Nous tenons aussi à remercier tous ceux qui ont participé de près ou de loin à notre formation.*

*Nous exprimons notre profonde reconnaissance à Monsieur O. TAMI, notre Chef du Département de MATHÉMATIQUES.*

## DEDICACES

Je dédie ce modeste travail à mes très chers parents, dont le sacrifice, la tendresse, l'amour, la patience, le soutien, l'aide et les encouragements ont été et sont l'essence de ma réussite. Sans eux, je ne serais pas ce que je suis aujourd'hui.

Cette dédicace n'oubliera pas ma petite et grande famille en tentant de rendre hommage à mes chers frères et sœurs, pour leur soutien, leur aide et leurs encouragements ainsi Mohamed afif mon ange et Mohamed read et Madji mokhtar.

A mon compagnon de route, HADJAMAR FARID et toutes sa familles ;

Sans oublier mon ami d'enfance Bahria ahmed

A tout mes enseignants, ainsi que mes amis restés fidèles.

*Mahfoud BENNOUR*

## DEDICACES

Je dédie ce modeste travail à l'esprit de mes très chers parents, dont le sacrifice, la tendresse, l'amour, la patience, le soutien, l'aide et les encouragements ont été et sont l'essence de ma réussite. Sans eux, je ne serais pas ce que je suis aujourd'hui.

Cette dédicace n'oubliera pas ma petite et grande famille en tentant de rendre hommage à mes chers frères et sœurs, pour leur soutien, leur aide et leurs encouragements.

A mon compagnon de route, BENNOUR MAHFOUD et toutes sa familles sans oublier mon ami et frère TCHAMBAZ AHMED et toute sa famille;

A tout mes enseignants, ainsi que mes amis restés fidèles.

HADJAMAR FARID

## ملخص

العمل المعروض في هذه المذكرة يتضمن إيجاد نموذج و التحكم بالمنطق الغامض و يهجن المنطق الغامض مع الخوارزميات الجينية مطبقة على القمر الاصطناعي. في الأول، نبدأ بنظرة عامة على الأقمار، نصف فيها حركتها. بعد ذلك، مبادئ ابتدائية حول المنطق الغامض و الخوارزميات الجينية. الجزء الثالث يعتمد على تطبيق تقنيات التحكم على نوعين من الأقمار (micro satellite et mini satellite) و في الأخير عرضنا نتائج بهدف تطبيق التقنيات المقترحة.

كلمات مفتاحية : حركيات و تحرك القمر الصناعي، المنطق الغامض ، خوارزميات معسكرات النمل.

## Abstract

The work presented in this memory relates to modeling and the order with the hybridization of fuzzy logic with the algorithms of the colonies of ants applied to the artificial satellite; in a first place, highlights the métaheuristiques ones.

Then, of the basic concepts on fuzzy logic and the algorithms of the colonies of ants are given.

The third part is devoted to the application of the techniques of orders caused on a type of satellites to knowing the mini satellite. Results of simulation are also presented in order to validate the techniques suggested and evaluating the performances obtained.

Key words: Modeling of the movement of the artificial satellites, fuzzy controller, optimization, algorithms of the colonies of ants.

## Résumé

Le travail présenté dans ce mémoire concerne la modélisation et la commande par l'hybridation de la logique floue avec les algorithmes des colonies de fourmis appliquée aux satellite artificiel ; dans un premier lieu, un aperçus général les métaheuristiques.

Ensuite, des notions de base sur la logique floue et les algorithmes des colonies de fourmis sont données.

La troisième partie est consacrée à l'application des techniques de commandes suscitées sur un type de satellites à savoir le mini satellite. Des résultats de simulation sont également présentés afin de valider les techniques proposées et évaluer les performances obtenues.

**Mots clés :** Modélisation du mouvement des satellites artificiels, contrôleur flou, optimisation, algorithmes des colonies de fourmis.

# Table des matières

## Chapitre I

## Optimisation :L'état de l'art

---

I.1 Introduction .....	1
I.2 Cadre de l'optimisation combinatoire.....	1
I.2.1 L'optimisation difficile.....	2
I.2.2 Exemples de problèmes .....	3
I.2.3 Structure de voisinage et minimum local.....	4
I.2.4 Méthode exacte et méthode approchée.....	5
I.2.5 Heuristique et metaheuristique .....	5
I.2.6 Classification des metaheuristiques.....	6
I.3 Principales metaheuristiques.....	7
I.3.1 La méthode de descente .....	7
I.3.2 Avantages et inconvénients .....	8
I.3.3 La méthode de recuit simulé.....	8
I.3.4 Avantages et inconvénients .....	9
I.3.5 Applications .....	10
I.3.6 La méthode Tabou .....	10
I.4 Les Algorithmes génétiques .....	10
I.4.1 Principe de l'Algorithme génétique.....	11
I.4.2 Opérateurs de sélection.....	11
I.4.2.1 Sélection par tournois .....	12
I.4.2.2 Sélection par proportionnelle .....	12
I.4.2.3 Opérateurs de croisement.....	13
I.4.2.4 Avantages et inconvénient.....	14
I.5 Les Algorithmes de colonies de fourmis .....	15
I.5.1 Principe de l'algorithme .....	16
I.5.2 Généralisation et variantes .....	18
I.5.3 Avantages et inconvénients .....	18
I.5.4 Stratégies de recherche .....	19
I.5.4.1 intensification et diversification.....	19
I.5.4.2 Dans les algorithmes génétiques.....	19
I.5.4.3 Dans les algorithmes de colonies fourmis .....	20
1.6 Conclusion .....	20

---

II.1	Généralités sur les satellites artificiels .....	21
II.1.1	Classement des satellites .....	21
II.1.2	Domaines d'utilisation des satellites .....	21
II.3.1	Les repères d'attitude .....	23
II.3.2	Capteurs et actionneurs.....	24
II.3.2.1	Les capteurs .....	25
II.3.2.2	Les actionneurs .....	25
II.4	Modélisation dynamique du satellite .....	25
II.4.1	Principaux repères utilisés dans la modélisation.....	26
II.4.2	Couples de gradient de gravité .....	29
II.4.2.1	Modélisation en présence du couple de gradient de gravité.....	29
II.4.2.2	Modélisation du mouvement libre du satellite en présence du couple de gradient de gravité.....	30

---

III.1	Introduction.....	32
III.2	Principe fondamental de la logique floue .....	32
III.3	Les différents domaines d'application .....	33
III.4	Bases générales de la logique floue .....	34
III.4.1	Les sous-ensemble Flou.....	34
III.4.2	Variables floues linguistiques .....	34
III.4.2.1	Formes des fonctions d'appartenance.....	34
III.4.2.2	Considérations générales sur les fonctions d'appartenances ..	36
III.4.3	Déductions floues (Inférences.....	36
III.4.3.1	Inférence avec une seule règle .....	36
III.4.3.2	Inférence avec plusieurs règles.....	37
III.4.4	Opérateurs de la logique floue .....	38
III.4.4.1	Représentation de l'opérateur NON par le complémentaire de A.....	38
III.4.4.2	Opérateur ET .....	38
III.4.4.3	Opérateur OU .....	39
III.5	Structure d'une commande floue .....	40
III.5.1	Généralité .....	40
III.5.2	Avantage et désavantages du réglage par logique floue.....	40
III.5.3	Base de règles et définition.....	41
III.5.4	Procédure de raisonnement flou .....	42
III.6	Conclusion.....	47



VI.1	Introduction .....	48
VI.2	Principe d'optimisation.....	48
VI.3	Problématique du sujet de l'adaptation des colonies de fourmis.....	51
VI.4	Les facteurs d'échelle (gain de normalisation .....	52
VI.5	Optimisation sous contraintes.....	53
VI.6	Description du système .....	54
VI.6.1	Étude du mouvement libre du satellite .....	54
VI.6.2	Étude du système en boucle fermée.....	54
VI.7	Application.....	55
	Conclusion	
	Bibliographie	
	Annexes	

---

## INTRODUCTION

Les métaheuristiques forment un ensemble de méthodes utilisées en recherche opérationnelle pour résoudre des problèmes d'optimisation réputés difficiles. Résoudre un problème d'optimisation combinatoire, c'est trouver l'optimum d'une fonction, parmi un nombre fini de choix, souvent très grand. Les applications concrètes sont nombreuses, que ce soit dans le domaine de la production industrielle, des transports ou de l'économie ainsi que celui de la recherche spatiale et pratiquement partout où se fait sentir le besoin de minimiser des fonctions numériques, dans des systèmes où interviennent simultanément un grand nombre de paramètres.

A ces problèmes de minimisation, les métaheuristiques permettent, dans des temps de calcul raisonnables, de trouver des solutions, peut-être pas toujours optimales, en tout cas très proches de l'optimum ; elles se distinguent en cela des méthodes dites *exactes*, qui garantissent certes la résolution d'un problème, mais au prix de temps de calcul prohibitifs pour nombres d'applications industrielles.

Nous nous proposons de fournir un panorama de toutes ces techniques, parfois redoutablement efficaces, qui se développent depuis une vingtaine d'années.

Après avoir donné quelques définitions préalables, et rappelé les énoncés de quelques problèmes traditionnels d'optimisation combinatoire, nous replacerons les métaheuristiques dans leurs contextes d'applications dans le domaine spatial en les appliquant à la commande d'attitude d'un satellite artificiel pour lequel nous dégagerons les principales propriétés.

Une revue des métaheuristiques les plus usuelles nous fera passer dans l'analyse de leur mode sommaire de fonctionnement, en évoquant les évolutions auxquelles elles ont donné lieu, et en soulignant leurs avantages et leurs inconvénients respectifs.

Nous verrons à cette occasion toute la richesse et la variété des procédés mis en jeu dans les métaheuristiques.

## 1. CADRE DE L'OPTIMISATION COMBINATOIRE

En mathématiques, *l'optimisation* recouvre toutes les méthodes qui permettent de déterminer l'optimum d'une fonction, avec ou sans contraintes. En théorie, un problème d'optimisation combinatoire se définit par l'ensemble de ses instances, souvent infiniment nombreuses. Dans la pratique, le problème se ramène à résoudre numériquement l'une de ces instances, par un procédé algorithmique.

A chaque instance du problème est associé un ensemble discret de solutions  $S$ , un sous-ensemble  $X$  de  $S$  représentant les solutions admissibles (réalisables), ainsi qu'une fonction de coût  $f$  (appelée aussi *fonction objectif*).  $f$  associe à chaque solution  $s \in X$  le nombre  $f(s)$ .

Résoudre un problème d'optimisation combinatoire consiste alors à trouver une solution  $s \in X$  optimisant la valeur de la fonction de coût  $f$ . **Formellement, on cherche donc  $s^* \in X$  tel que  $f(s^*) \leq f(s)$  pour tout  $s \in X$ .**

Une telle solution  $s^*$  s'appelle une *solution optimale*, ou un *optimum global*.

**Remarque** : on peut tout aussi bien résoudre des problèmes de maximisation, les principes de résolution restant naturellement les mêmes.

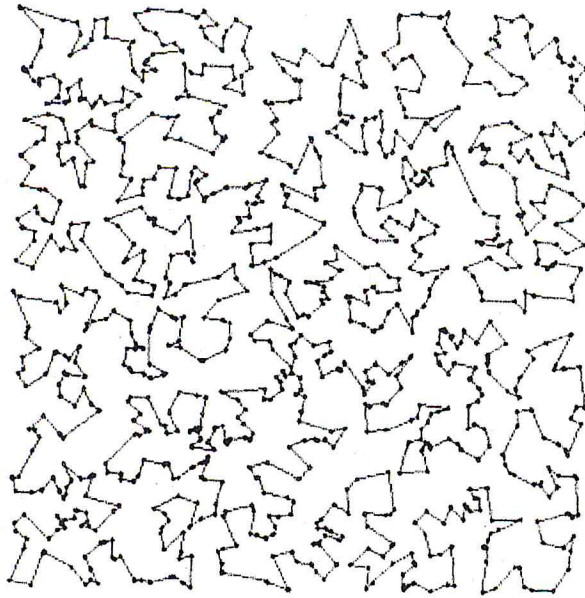


Schéma 1 : représentation du chemin minimal passant par une distribution aléatoire de 1000 points.

Dans le schéma ci-dessus, le problème était de trouver le chemin minimal passant une seule fois par un certain nombre de points (problème du voyageur de commerce, voir paragraphe suivant). Ce graphique correspond à une certaine *instance* du problème. Avec une distribution différente de points, nous aurions eu une autre instance du problème.

Il existe une seconde formalisation du problème d'optimisation combinatoire, plus générale, qui inclue la notion de *contraintes* et d'*affectation* de valeurs à des variables. En voici la définition. Soit :

- un ensemble de variables  $(x_1, x_2, \dots, x_n)$
- un ensemble de domaines de définitions  $D_1, D_2, \dots, D_n$
- un ensemble de contraintes  $C$  sur les variables (par exemple, des inéquations, ou bien l'obligation que toutes les variables prennent des valeurs différentes)
- une fonction objective  $f$  que l'on cherche à optimiser :  $f : D_1 \times D_2 \times \dots \times D_n \rightarrow \mathfrak{R}$

L'ensemble  $S$  des solutions possibles peut alors se définir comme :  
 $S = \{s = \{(x_1, v_1), \dots, (x_n, v_n)\} \text{ tels que } v_i \in D_i, \text{ et tels que } s \text{ satisfasse toutes les contraintes } C\}$

Les problèmes d'affectation sous contraintes possèdent de nombreuses applications pratiques, comme l'affectation de ressources, le groupement, la classification, la planification, l'emploi du temps et l'ordonnancement, dans des domaines très variés...

### 1.1 L'optimisation difficile

L'« optimisation combinatoire » consiste à trouver la meilleure solution entre un nombre fini de choix. Autrement dit, à minimiser une fonction, avec ou sans contraintes, sur un ensemble fini de possibilités. Quand le nombre de combinaisons possibles devient exponentiel par rapport à la taille du problème, le temps de calcul devient rapidement critique.

Ce temps de calcul devient si problématique que pour certains problèmes, on ne connaît pas d'algorithme exact *polynomial*, c'est-à-dire dont le temps de calcul soit proportionnel à  $N^n$ , où  $N$  désigne le nombre de paramètres inconnus du problème, et où  $n$  est une constante entière. Lorsqu'on conjecture qu'il n'existe pas une telle constante  $n$  telle qu'un polynôme de degré  $n$  puisse borner le temps de calcul d'un algorithme, on parle alors d'optimisation difficile, ou de problèmes *NP-difficiles* (c'est tout l'objet de la théorie de la NP-complétude).

## 1.2. Exemples de problèmes

L'un des exemples les plus caractéristiques du champ de l'optimisation combinatoire est le problème du **voyageur de commerce**, dans lequel un représentant doit visiter un certain nombre de villes, avant de retourner à son point de départ : la question est de déterminer le trajet le plus court traversant chaque ville une seule fois, c'est-à-dire la succession de villes qui minimise la tournée du représentant.

De nombreux problèmes d'ingénierie peuvent se ramener au problème du voyageur de commerce (PVC), d'où son intérêt. On le retrouve dans le domaine des réseaux informatiques par exemple, avec les algorithmes de routage.

Par ailleurs, le PVC est un cas particulier d'un problème plus général, celui des **tournées de véhicules**, qui consiste à calculer les meilleurs parcours pour une flotte de véhicules devant livrer un ensemble de produits dans plusieurs destinations à partir d'un entrepôt. Chaque véhicule a une capacité limitée. Le problème des tournées de véhicules trouve ses applications dans le domaine de la distribution bien sûr, mais aussi dans les télécoms.

Un autre problème classique est celui de l'**affectation quadratique**, qui, par rapport au PVC, introduit la notion de *flot* circulant entre les villes, et qui peut s'énoncer ainsi : il s'agit de déterminer comment placer  $n$  objets dans  $n$  emplacements de manière à minimiser la somme des produits *flots* par *distances*. Mathématiquement, cela revient à minimiser :

$$\sum_{i=1}^n \sum_{j=1}^n f_{ij} \cdot d_{ij}$$
 Où  $f_{ij}$  représente le flot entre l'objet  $i$  et l'objet  $j$ , et  $d_{ij}$  les distances entre l'objet  $i$  et l'objet  $j$ .

La répartition de bâtiments dans un espace donné en fonction du nombre de personnes amenées à circuler entre ces bâtiments, ou la façon de répartir des modules électroniques sur une carte en fonction du nombre de connexions les liant les uns aux autres, revient à résoudre le problème de l'affectation quadratique. Enfin, troisième problème combinatoire caractéristique, l'**ordonnement de tâches**, dont l'énoncé classique est le suivant : soit un ensemble  $m$  de machines, un ensemble  $t$  de tâches à réaliser, chaque tâche étant composée d'une certaine suite d'opérations. Une machine ne peut réaliser qu'une seule opération à la fois. Comment ordonner l'exécution des opérations sur les différentes machines de telle façon que le temps mis à faire tout le travail soit fait soit minimal ?

Naturellement, dans la vie courante, les situations sont encore plus compliquées, et il faut rajouter à ces situations-types des contraintes, des particularités, certaines « règles du jeu » qui viennent encore compliquer l'énoncé du problème à résoudre.

### 1.3. Structure de voisinage et minimum local

Soit  $S$  un ensemble de solutions à un problème d'optimisation, et soit  $f$  la fonction objectif. Une structure de voisinage (ou tout simplement un voisinage) est une fonction  $N$  qui associe un sous-ensemble de  $S$  à toute solution  $s \in S$ . Une solution  $s' \in N(s)$  est dite *voisine* de  $s$ .

Une solution  $s \in S$  est un *minimum local* relativement à la structure de voisinage  $N$  si :

$$f(s) \leq f(s') \quad \forall s' \in N(s).$$

Une solution  $s \in S$  est un *minimum global* si  $f(s) \leq f(s') \quad \forall s' \in S$ .

Certaines méthodes d'optimisation, qui partent d'une solution initiale et qui l'améliorent en explorant son voisinage immédiat, présentent l'inconvénient de s'arrêter au premier minimum local trouvé.

Comme nous le verrons plus loin, les métaheuristiques contiennent donc souvent une technique ou une astuce permettant d'éviter de se retrouver piégé dans ces minima locaux, en explorant davantage tout l'espace des solutions, de façon à augmenter la probabilité de rencontrer le minimum optimal, c'est-à-dire le minimum global.

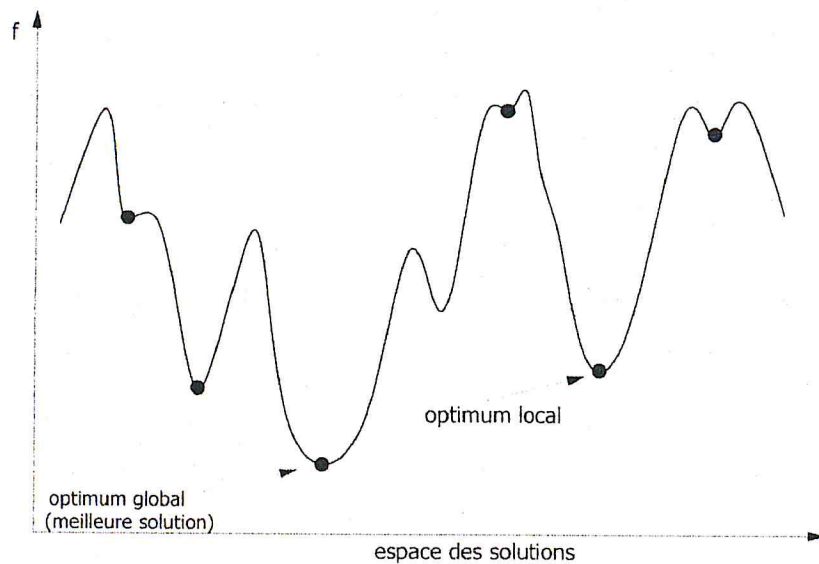


Schéma 2 : analogie entre une fonction numérique à une variable,  
Et la fonction de coût d'un problème combinatoire

Dans le cadre de l'optimisation combinatoire, en pratique, on aura tout intérêt à définir le voisinage en considérant l'ensemble des modifications élémentaires que l'on peut appliquer à une solution  $s$  donnée, par exemple l'ensemble des permutations (si les solutions peuvent s'écrire sous la forme d'une séquence finie d'éléments, comme le cas se présente fréquemment en optimisation combinatoire)

Si cet ensemble est trop grand, on pourra toujours le réduire à un sous-ensemble, aléatoirement, ou en fonction d'un critère précis. Ainsi, dans le cas du problème du voyageur de commerce, et partant d'un trajet donné  $A - B - C - D - E$ , un voisinage pourra se définir comme l'ensemble des inversions de 2 villes sur le parcours (dans notre exemple, une inversion possible serait :  $A - B - D - C - E$ ), ou bien comme l'ensemble des déplacements unitaires (dans notre exemple, un déplacement serait :  $A - B - E - C - D$ )

#### 1.4 Méthode exacte et méthode approchée

Une recherche exhaustive par énumération explicite de toutes les solutions est impensable pour résoudre un problème d'optimisation difficile en raison du temps de calcul induit. Dans le cas du problème du voyageur de commerce, par exemple, l'espace de recherche croît en  $(n-1)!$ , où  $n$  est le nombre de villes à visiter, ce qui dépasse rapidement les capacités de calcul de n'importe quel ordinateur. Avec seulement 50 villes, il faudra évaluer  $49!$  Trajets, soit  $6,08.10^{62}$  trajets. C'est l'explosion combinatoire.

Néanmoins, la résolution d'un tel problème d'optimisation peut se faire de manière exacte, en modélisant soigneusement le problème, puis en appliquant un algorithme ad-hoc, qui écarte d'emblée l'examen de certaines configurations, dont on sait d'ores et déjà qu'elles ne peuvent pas être optimales.

Parmi les méthodes exactes, on trouve la plupart des méthodes traditionnelles (développées depuis une trentaine d'années) telles les techniques de *séparation et évaluation* (branch-and-bound), ou les *algorithmes avec retour arrière* (backtracking). Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable. Mais malgré les progrès réalisés (notamment en matière de programmation linéaire en nombres entiers), comme le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème, les méthodes exactes rencontrent généralement des difficultés avec les applications de taille importante.

Notons qu'il existe ainsi des applications informatiques génériques (AMPL, CPLEX, LINDO, MPL, OMP, XPRESS...) qui permettent à l'ingénieur de résoudre facilement les problèmes pouvant s'écrire sous une forme algébrique en variables binaires ou entières.

Si les méthodes de résolution exactes permettent d'obtenir une ou plusieurs solutions dont l'optimalité est garantie, dans certaines situations, on peut cependant se contenter de solutions de bonne qualité, sans garantie d'optimalité, mais au profit d'un temps de calcul réduit. On utilise pour cela une méthode **heuristique**, adaptée au problème considéré, avec cependant l'inconvénient de ne disposer en retour d'aucune information sur la qualité des solutions obtenues.

#### 1.5 Heuristique et métaheuristique

Afin d'améliorer le comportement d'un algorithme dans son exploration de l'espace des solutions d'un problème donné, le recours à une méthode *heuristique* (du verbe grec *heuriskein*, qui signifie « trouver ») permet de guider le processus dans sa recherche des solutions optimales.

Feignebaum et Feldman (1963) définissent une **heuristique** comme une règle d'estimation, une stratégie, une astuce, une simplification, ou toute autre sorte de système qui limite drastiquement la recherche des solutions dans l'espace des configurations possibles. Newell, Shaw et Simon (1957) précisent qu'un processus heuristique peut résoudre un problème donné, mais n'offre pas la garantie de le faire.

Dans la pratique, certaines heuristiques sont connues et ciblées sur un problème particulier.

La **métaheuristique**, elle, se place à un niveau plus général encore, et intervient dans toutes les situations où l'ingénieur ne connaît pas d'heuristique efficace pour résoudre un problème donné, ou lorsqu'il estime qu'il ne dispose pas du temps nécessaire pour en déterminer une.

En 1996, I.H. Osman et G. Laporte définissaient la métaheuristique comme « un processus itératif qui subordonne et qui guide une heuristique, en combinant intelligemment plusieurs concepts pour explorer et exploiter tout l'espace de recherche. Des stratégies d'apprentissage sont utilisées pour structurer l'information afin de trouver efficacement des solutions optimales, ou presque-optimales ».

En 2006, le réseau Metaheuristics (metaheuristics.org) définit les métaheuristicques comme « un ensemble de concepts utilisés pour définir des méthodes heuristiques, pouvant être appliqués à une grande variété de problèmes. On peut voir la métaheuristique comme une « boîte à outils » algorithmique, utilisable pour résoudre différents problèmes d'optimisation, et ne nécessitant que peu de modifications pour qu'elle puisse s'adapter à un problème particulier ».

Elle a donc pour objectif de pouvoir être programmée et testée rapidement sur un problème. Comme l'heuristique, la métaheuristique n'offre généralement pas de garantie d'optimalité, bien qu'on ait pu démontrer la convergence de certaines d'entre elles. Non déterministe, elle incorpore souvent un principe stochastique pour surmonter l'explosion combinatoire. Elle fait parfois usage de l'expérience accumulée durant la recherche de l'optimum, pour mieux guider la suite du processus de recherche.

## 1.6 Classification des métaheuristicques

Les métaheuristicques n'étant pas, a priori, spécifiques à la résolution de tel ou tel type de problème, leur classification reste assez arbitraire. On peut cependant distinguer :

### a) les approches « **trajectoire** »

Ces algorithmes partent d'une solution initiale (obtenue de façon exacte, ou par tirage aléatoire) et s'en éloignent progressivement, pour réaliser une trajectoire, un parcours progressif dans l'espace des solutions. Dans cette catégorie, se rangent :

- la méthode de descente
- le recuit simulé
- la méthode Tabou
- la recherche par voisinage variable

Le terme de *recherche locale* est de plus en plus utilisé pour qualifier ces méthodes.

### b) les approches « **population** » (ou évolutionnaires)

Elles consistent à travailler avec un ensemble de solutions simultanément, que l'on fait évoluer graduellement. L'utilisation de plusieurs solutions simultanément permet naturellement d'améliorer l'exploration de l'espace des configurations. Dans cette seconde catégorie, on recense :

- les algorithmes génétiques
- les algorithmes par colonies de fourmi
- l'optimisation par essaim particulaire
- les algorithmes à estimation de distribution
- le path relinking (ou chemin de liaison)

Notons d'ores et déjà que ces métaheuristicques évolutionnaires seront probablement plus gourmandes en calculs, mais on peut supposer aussi qu'elles se prêteront bien à leur parallélisations.

Certains algorithmes peuvent se ranger dans les deux catégories à la fois, comme la méthode GRASP, qui construit un ensemble de solutions, qu'elle améliore ensuite avec une recherche locale. La plupart des méthodes développées ces dernières années sont d'ailleurs souvent à

cheval sur ces deux approches. Une autre manière, plus intuitive, de classer les métaheuristiques consiste à séparer celles qui sont inspirées d'un **phénomène naturel**, de celles qui ne le sont pas. Les algorithmes génétiques ou les algorithmes par colonies de fourmi entrent clairement dans la première catégorie, tandis que la méthode de descente, ou la recherche Tabou, vont dans la seconde. On peut également raisonner par rapport à l'usage que font les métaheuristiques de la **fonction objectif**. Certaines la laissent « telle quelle » d'un bout à l'autre du processus de calcul, tandis que d'autres la modifient en fonction des informations collectées au cours de l'exploration – l'idée étant toujours de « s'échapper » d'un minimum local, pour avoir davantage de chance de trouver l'optimal. La *recherche locale guidée* est un exemple de métaheuristique qui modifie la fonction objectif. Enfin, il faut distinguer les métaheuristiques qui ont la faculté de **mémoriser** des informations à mesure que leur recherche avance, de celles qui fonctionnent sans mémoire, en aveugle, et qui peuvent revenir sur des solutions qu'elles ont déjà examinées. On distingue la mémoire à court terme (celles des derniers mouvements effectués), et la mémoire à long terme (qui concerne des paramètres synthétiques plus généraux). Le meilleur représentant des métaheuristiques avec mémoire reste la recherche Tabou. Dans les « sans mémoire », on trouve le recuit simulé par exemple.

## 2. PRINCIPALES METAHEURISTIQUES

Dans ce chapitre, nous passons en revue les principales métaheuristiques, en commençant par celles qui exploitent séquentiellement un seul voisinage (approche « trajectoire »), avant d'étudier celles qui exploitent plusieurs solutions à la fois (approche « population »).

### 2.1 La méthode de descente

Le principe de la méthode de descente (dite aussi *basic local search*) consiste à partir d'une solution  $s$  et à choisir une solution  $s'$  dans un voisinage de  $s$ , telle que  $s'$  améliore la recherche (généralement telle que  $f(s') < f(s)$ ).

On peut décider soit d'examiner *toutes* les solutions du voisinage et prendre la meilleure de toutes (ou prendre la première trouvée), soit d'examiner un sous-ensemble du voisinage.

La méthode de recherche locale la plus élémentaire est la *méthode de descente*. On peut la schématiser comme suit :

Procédure *descente simple* (solution initiale  $s$ )

Répéter :

Choisir  $s'$  dans  $N(s)$

Si  $f(s') < f(s)$  alors  $s \leftarrow s'$

Jusqu'à ce que  $f(s') \geq f(s), \forall s' \in S$

Fin

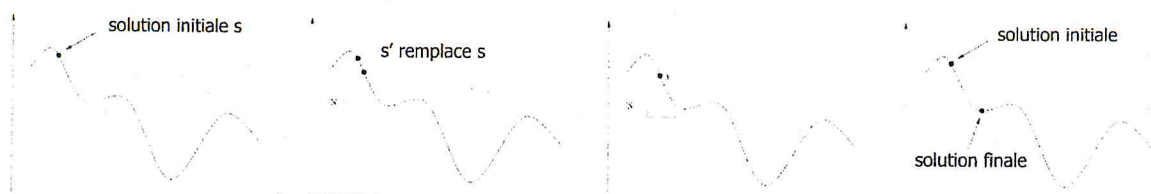


Schéma 3 : évolution d'une solution dans la méthode de descente



On peut varier cette méthode en choisissant à chaque fois la solution  $s'$  dans  $N(s)$  qui améliore le plus la valeur de  $f$ . C'est la *méthode de plus grande descente*.

## 2.2 Avantages et inconvénients

En général, l'efficacité des méthodes de recherche locale simples (descente, ou plus grande descente) est très peu satisfaisante. D'abord, par définition, la recherche s'arrête au premier minimum local rencontré, c'est là leur principal défaut. Pour améliorer les résultats, on peut lancer plusieurs fois l'algorithme en partant d'un jeu de solutions initiales différentes, mais la performance de cette technique décroît rapidement. En revanche, autoriser de temps à autre une certaine dégradation des solutions trouvées, afin de mieux explorer tout l'espace des configurations, a conduit au développement des deux méthodes que nous explorons aux paragraphes suivants, à savoir le recuit simulé et la méthode Tabou. Le principal avantage de la recherche locale simple est évidemment sa grande simplicité de mise en œuvre: la plupart du temps, elle ne fait que calculer  $f(s+i)-f(s)$ , où  $i$  correspond à un déplacement élémentaire, et si cette expression peut se simplifier algébriquement, alors on pourra évaluer très rapidement cette différence. Il est important de remarquer également l'importance du choix de la fonction de voisinage  $N$ : un minimum local pour une certaine structure de voisinage ne l'est pas forcément pour une autre. C'est d'ailleurs ce constat qui est à l'origine de la méthode dite de *recherche par voisinage variable*, qui repose sur la construction de solutions  $s$  parmi plusieurs voisinages  $N_i$ , plutôt que dans un seul.

## 2.3 La méthode du recuit simulé

Le recuit simulé (simulated annealing) est souvent présenté comme la plus ancienne des métaheuristiques, en tout cas, la première à mettre spécifiquement en œuvre une stratégie d'évitement des minima locaux (Kirkpatrick, Gelatt, Vecchi, 1983). Elle s'inspire d'une procédure utilisée depuis longtemps par les métallurgistes qui, pour obtenir un alliage sans défaut, chauffent d'abord à blanc leur morceau de métal, avant de laisser l'alliage se refroidir très lentement (technique du recuit). Pour simuler cette évolution d'un système physique vers son équilibre thermodynamique à une température  $T$ , la méthode du recuit simulé exploite l'algorithme de *Métropolis*. Dans l'algorithme de *Métropolis*, on part d'une configuration donnée, et on fait subir au système une modification élémentaire. Si cette perturbation a pour effet de diminuer la fonction objectif (ou *énergie*) du système, elle est acceptée. Sinon, elle est acceptée avec la probabilité  $\exp(\Delta E/T)$ . En appliquant itérativement cette règle, on engendre une séquence de configurations qui tendent vers l'équilibre thermodynamique. On peut systématiser l'algorithme avec le pseudo-code suivant :

Procédure *recuit simulé* (solution initiale  $s$ )

Poser  $T \leftarrow T_0$

Répéter :

    Choisir aléatoirement  $s' \in N(s)$

    Générer un nombre réel aléatoire  $r$  dans  $[0,1]$

    Si  $r < e^{-\frac{f(s)-f(s')}{T}}$  alors Poser  $s \leftarrow s'$

    Mettre à jour  $T$

Jusqu'à ce que le critère de terminaison soit satisfait

Fin

Et voici l'interprétation de son fonctionnement :

· Si  $f(s') < f(s)$  alors  $e^{-\frac{f(s)-f(s')}{T}} > 1$ , donc  $r$  est toujours inférieur à cette valeur, et on accepte la solution  $s'$  (une meilleure solution est donc toujours acceptée, ce qui paraît logique).

· Si  $f(s') > f(s)$  et  $T$  est très grand, alors  $e^{-\frac{f(s)-f(s')}{T}} \approx 1$ , et on il y a de fortes chances d'accepter  $s'$  (bien que la solution  $s'$  soit plus « mauvaise » que  $s$  !)

· Si  $f(s') > f(s)$  et  $T$  est très petit, alors  $e^{-\frac{f(s)-f(s')}{T}} \approx 0$ , et on va donc probablement refuser  $s'$

Dans un premier temps,  $T$  étant généralement choisi très grand, beaucoup de solutions, même celles dégradant la valeur de  $f$ , sont acceptées, et l'algorithme équivaut à une visite aléatoire de l'espace des configurations. Mais à mesure que la température baisse, la plupart des mouvements augmentant l'énergie sont refusés, et l'algorithme se ramène à une amélioration itérative classique. A température intermédiaire, l'algorithme autorise de temps en temps des transformations qui dégradent la fonction objective. Il laisse ainsi une chance au système de s'extraire d'un minimum local.

Cet algorithme est parfois amélioré en ajoutant une variable qui mémorise la meilleure valeur rencontrée jusqu'à présent (sans cela, l'algorithme pourrait converger vers une certaine solution  $s$ , alors qu'on avait visité auparavant une solution  $s'$  de valeur inférieure à  $f(s)$  !)

### 2.3.1 Avantages et inconvénients

Le recuit simulé présente l'avantage d'offrir des solutions de bonne qualité, tout en restant simple à programmer et à paramétrer. Il offre autant de souplesse d'emploi que l'algorithme de recherche local classique : on peut inclure facilement des contraintes dans le corps du programme.

Aarts, Korst et Laarhoven (1997) ont par ailleurs démontré que, sous certaines conditions de décroissance de la température, l'algorithme du recuit simulé converge en probabilité vers un optimum global lorsque le nombre d'itérations tend vers l'infini.

L'un des inconvénients du recuit simulé est qu'une fois l'algorithme piégé à basse température dans un minimum local, il lui est impossible de s'en sortir tout seul. Plusieurs solutions ont été proposées pour tenter de résoudre ce problème, par exemple en acceptant une brusque remontée de la température, de temps en temps, pour relancer la recherche sur d'autres régions plus éloignées. Il est également possible d'empêcher la température de descendre trop bas : on lui donne une valeur minimale au delà de laquelle on ne change plus de palier de température. Mais si cette valeur est trop grande, l'algorithme passera son temps à augmenter et diminuer son énergie car il acceptera trop de perturbations dégradantes et il n'arrivera pas à explorer à fond une vallée. Ainsi, il est fort possible que l'algorithme arrive à "trouver" la vallée dans laquelle se cache un minimum global, mais il aura beaucoup de mal à l'explorer et donc risque de s'en éloigner sans avoir décelé la solution au problème...

C'est là le principal problème du recuit : le paramétrage de la température peut être rebutant et très empirique. On constate également qu'il faut une diminution de la température « suffisamment lente », donc un certain nombre d'itérations, pour voir la solution s'améliorer.

Corrélativement, le temps de calcul devient parfois excessif avec certaines applications, et l'algorithme exige en particulier le calcul d'une exponentielle.

### 2.3.2 Applications

L'algorithme du recuit simulé a montré son efficacité sur les problèmes combinatoires classiques, surtout sur les échantillons de grande taille. Par exemple, l'expérience a montré qu'il ne devenait vraiment efficace sur le problème du voyageur de commerce qu'au-delà d'environ 800 villes. Il a été testé avec succès sur le problème du partitionnement de graphe. Des analyses ont montré qu'il était efficace avec certaines catégories de problème où l'ensemble des solutions possèdent certaines propriétés particulières. Ceci expliquerait le succès du recuit simulé dans le domaine du placement des circuits électroniques, où il est très employé.

### 2.4 La méthode Tabou

La méthode Tabou est une technique de recherche dont les principes ont été proposés pour la première fois par Fred Glover dans les années 80, et elle est devenue très classique en optimisation combinatoire. Elle se distingue des méthodes de recherche locale simples par le recours à un historique des solutions visitées, de façon à rendre la recherche un peu moins « aveugle ». Il devient donc possible de s'extraire d'un minimum local, mais, pour éviter d'y retomber périodiquement, certaines solutions sont bannies, elles sont rendues « taboues ». A l'inverse du recuit simulé qui génère de manière aléatoire une seule solution voisine  $s' \in N(s)$  à chaque itération, Tabou examine un échantillonnage de solutions de  $N(s)$  et retient la meilleure  $s'$  même si  $f(s') > f(s)$ . La recherche Tabou ne s'arrête donc pas au premier optimum trouvé. Le danger serait alors de revenir à  $s$  immédiatement, puisque  $s$  est meilleure que  $s'$ . Pour éviter de tourner ainsi en rond, on crée une liste  $T$  qui mémorise les dernières solutions visitées et qui interdit tout déplacement vers une solution de cette liste. Cette liste  $T$  est appelée liste Tabou.

### 2.5. LES ALGORITHMES GENETIQUES

Avec les algorithmes génétiques, nous passons à une autre catégorie de métaheuristiques, celles des méthodes dites *évolutionnaires*, qui manipulent un ensemble de plusieurs solutions simultanément. La métaheuristique la plus connue dans cette branche est celle reposant sur un algorithme génétique, inspiré du concept de sélection naturelle élaboré par Darwin. Ici, le vocabulaire employé est directement calqué sur celui de la théorie de l'évolution et de la génétique. Nous parlerons donc d'*individus*, pour parler de solutions (complètes, ou partielles). L'ensemble des individus formera une *population*, que nous ferons évoluer pendant une certaine succession d'itérations appelées *générations*, jusqu'à ce qu'un critère d'arrêt soit vérifié. Pour passer d'une génération à une autre, nous soumettrons la population à des *opérateurs de sélection*. Les *opérateurs de variation*, eux, permettront de transformer la population, de façon à favoriser l'émergence de meilleurs individus. Ces opérateurs sont à définir en fonction du problème considéré. Par exemple, dans le cas du voyageur de commerce, une certaine tournée sera représentée par un individu, caractérisé par une certaine succession de villes. Un opérateur transformera cet individu en modifiant les villes qui le constitue, en prenant soin de respecter les contraintes du problème donné, en l'occurrence ici, l'obligation que chaque ville ne soit présente qu'une seule fois dans la tournée.

On définit également la fonction de *performance* (fitness), qui traduit la capacité d'adaptation des individus au problème considéré. Cette fonction, que l'algorithme génétique cherche à maximiser, est étroitement dépendante de la fonction objectif.

### 2.5.1 Principe de l'algorithme génétique

L'algorithme génétique repose sur une boucle qui enchaîne des étapes de *sélections* et des étapes de *croisements*. Dans un premier temps, à partir d'une population de  $\alpha$  individus, on désigne ceux autorisés à se reproduire.

On croise ensuite ces derniers, de façon à obtenir une population d'enfants, dont on peut faire muter aléatoirement certains gènes.

La performance des enfants est évaluée, grâce à la fonction *fitness*, et l'on désigne, dans la population totale résultante parents+enfants, les individus autorisés à survivre, de telle manière que l'on puisse repartir d'une nouvelle population de  $\alpha$  individus.

La boucle est bouclée, et l'on recommence une phase de sélection pour la reproduction, une phase de mutation, et ainsi de suite.

Comme pour les métaheuristiques vues précédemment, un critère d'arrêt permet de sortir de la boucle, par exemple un certain nombre d'itérations sans amélioration notable de la performance des individus.

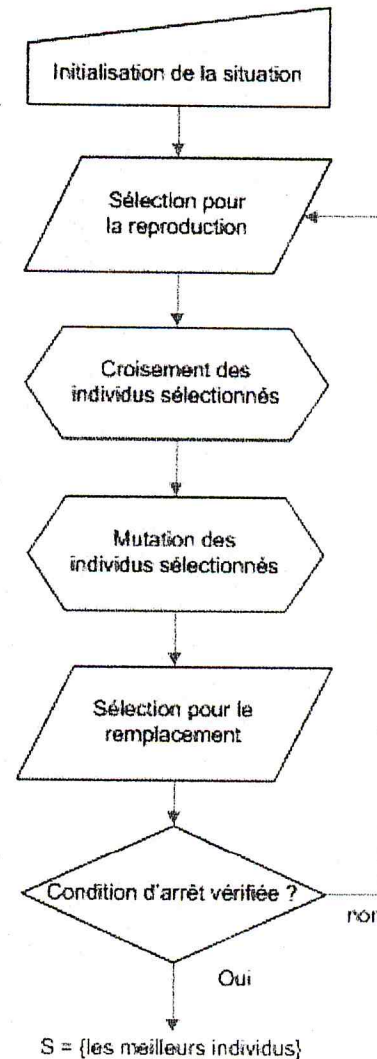


Schéma 4 : principe de fonctionnement d'un algorithme génétique

### 2.5.2 Opérateurs de sélection

Pour déterminer quels seront les  $n$  éléments, parmi les  $\mu$  individus de la population totale qui vont se reproduire entre eux, et former  $\lambda$  enfants, l'opérateur fixe la conduite de différents scénarios, pour décider qu'à chaque génération, les enfants remplacent tous les parents, ou au contraire, décider de faire se conserver la plupart des parents d'une génération à l'autre, et ne faire intervenir qu'un petit nombre d'enfants à chaque fois.

#### 2.5.2.1 Sélection par tournois

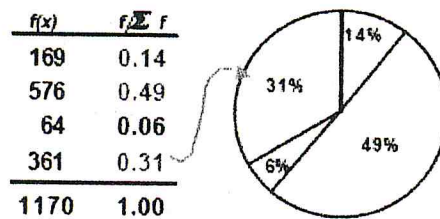
L'idée de cette méthode est de sélectionner aléatoirement un nombre  $k$  d'individus dans la population, et de choisir parmi eux celui qui a la meilleure performance. On organise autant de tournois qu'il y doit y avoir d'individus à sélectionner. On peut décider de faire des tirages avec, ou sans remise. Le tirage sans remise permet d'éviter de favoriser excessivement un

individu. Dans la mesure où seul l'ordre des éléments importe dans la sélection par tournois, il est parfois possible simplifier l'évaluation de la fonction objectif, et donc de gagner en temps de calcul.

**2.5.2.2 Sélection proportionnelle**

Dans la méthode proportionnelle, le nombre espéré de sélections d'un individu est proportionnel à sa performance. Pour expliciter cela, imaginons une roulette de casino, qui comporterait autant de cases que d'individus, et dont la taille des cases serait proportionnelle à la performance des individus. Chaque sélection est alors déterminée par le lancer d'une bille sur la roulette.

Ainsi, avec une population de quatre individus, dont les valeurs de performance seraient de 169, 576, 64 et 361, on obtiendrait la roulette suivante :



Avec cette technique, il faut s'arranger pour que  $f_i$  prenne des valeurs positives, en bornant inférieurement la fonction à 0, ou à une très petite valeur, pour que tout individu, même mauvais, ait quelque chance d'être sélectionné. La sélection proportionnelle risque cependant de favoriser excessivement un très bon individu au détriment des autres. Pour éviter cela, on peut se contenter d'un seul tirage aléatoire, qui fixe le premier élément sélectionné, les autres étant déterminés à partir de la position du premier, selon une succession de points équidistants.

$f_i$	5	4	1	10	4
	*	*	*	*	*

Par exemple : ici, un premier jet a élu l'individu de performance 5 : on a placé une étoile sur le ruban, au niveau de la case 5. Puis, un certain offset a déterminé les quatre autres enfants (on voit sur le schéma que la distance entre chaque étoile est fixe). C'est ainsi que, dans notre exemple, que l'individu de performance 1 n'est pas sélectionné. Cette méthode (*Stochastique Universal Sampling*) a l'avantage de pénaliser les mauvaises solutions, sans mettre trop en avant les meilleures.

**2.5.3 Opérateurs de croisement**

Les opérateurs de croisement, permettent de générer de nouveaux individus à partir de plusieurs (un couple, le plus souvent), et les opérateurs de mutation, qui transforment un seul individu.

Il n'y a pas de règle absolue pour construire des opérateurs de croisement. Il est possible d'apparier au hasard, mais en général, on préférera définir une distance dans l'espace de recherche, et combiner les individus proches selon cette distance. Par exemple, dans un

algorithme génétique appliqué à un problème de régulation de transport, on associera ensemble des individus générant peu de conflits, ou le moins de retards.

Pour pouvoir croiser, il faut déjà représenter les solutions.

C'est dans les années 60 que John Holland, le père des algorithmes génétiques, a mis au point un système de représentation binaire, inspiré du modèle phénotype-génotype du monde vivant. Ce modèle est devenu classique, quoique controversé. Dans ce système, les solutions du problème sont comparables à des chromosomes, et sont constituées d'une série de gènes, chaque gène étant associé à une séquence de bits, à une chaîne binaire codée en fonction d'une règle de conversion particulière. Croiser des individus consiste alors à « croiser » les séquences binaires de deux parents, pour former les « génomes » des enfants.

### Il existe plusieurs méthodes

Dans le croisement par point, le génome des deux enfants est constitué d'une partie de chacun des deux parents, par coupure et échange. Dans le croisement uniforme, on génère un masque binaire, de même longueur que le gène, qui détermine l'opération appliquée sur le gène des enfants (copie intacte, ou croisement des bits, selon la valeur des bits du masque).

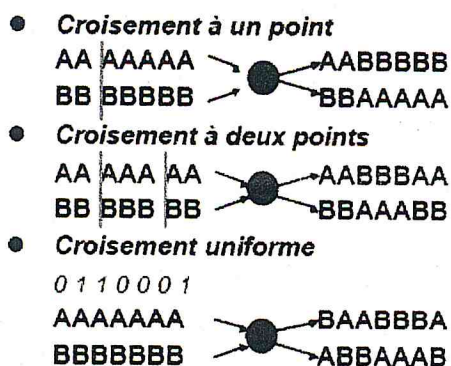
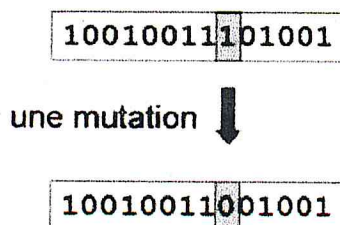


Schéma 5 : les différents types de croisements

Dans certains problèmes, on représente les individus sous forme d'arbre binaire, et le croisement se fera par coupure et inversion des branches : ceci est tout à fait adapté à la résolution de problème de calculs formels par exemple. Dans la mutation binaire, on modifie de temps en temps, avec une faible probabilité, les gènes des enfants : quelques bits, dans le gène d'un individu, sont inversés (un 0 devient un 1, un 1 devient un 0). Ce nombre de bits modifiés peut être fixe, ou choisi aléatoirement.

La mutation permet bien sûr d'introduire de la variété dans la population, et explorer davantage l'espace de recherche, mais si la mutation s'applique trop souvent, l'algorithme génétique revient à faire une marche aléatoire dans l'espace des solutions.



### 2.5.3.1 Avantages et inconvénients

D'abord, les algorithmes génétiques sont coûteux en temps de calcul, puisqu'ils manipulent plusieurs solutions simultanément. C'est le calcul de la fonction de performance qui est le plus pénalisant, et on optimise généralement l'algorithme de façon à éviter d'évaluer trop souvent cette fonction. Ensuite, l'ajustement d'un algorithme génétique est délicat. L'un des problèmes les plus caractéristiques est celui de la *dérive génétique*, qui fait qu'un bon individu se met, en l'espace de quelques générations, à envahir toute la population. On parle dans ce cas de convergence prématurée, qui revient à lancer à une recherche locale autour d'un minimum... qui n'est pas forcément l'optimum attendu. Les méthodes de sélection proportionnelle peuvent en particulier favoriser ce genre de dérive.

Un autre problème surgit lorsque les différents individus se mettent à avoir des performances similaires : les bons éléments ne sont alors plus sélectionnés, et l'algorithme ne progresse plus. Le choix d'une représentation « intelligente » pour permettre un remplacement générationnel efficace est un autre aspect de la question, et l'efficacité d'un algorithme génétique dépend beaucoup de la façon dont on opère le croisement des individus. Ainsi, dans le cas du problème du voyageur de commerce, on peut envisager d'apparier selon la méthode des coupures, telles que nous l'avons vu plus haut. Considérons les deux tournées suivantes, et opérons une coupure entre la troisième et la quatrième ville, puis inversons les gènes :

Parent 1 :     A – F – B – C – E – D  
 Parent 2 :     C – B – A – F – D – E

On obtient alors les successeurs :

Enfant 1 :     A – F – B – C – D – E  
 Enfant 2 :     C – B – A – F – E – D

Ici, des trajets initiaux, on n'a gardé que les trois premières villes (A – F – B), et on a placé dans l'enfant 1 les villes qui ne sont pas présentes dans la première partie (C – D – E), dans l'ordre où elles apparaissent dans le parent 2. On a fait le contraire pour l'enfant 2.

Mais on pourrait également recomposer en fonction du nombre d'adjacences de villes communes entre les deux parents. Si deux tournées possèdent des villes adjacentes en commun, il peut être intéressant que les trajets « enfant » les contiennent aussi. Des méthodes spécifiques de remplacement générationnel ont ainsi été élaborées, comme le « edge-3 » de Mathias et Whitley, qui ont montré leur grande efficacité. Mais il faut du temps pour se consacrer à l'étude et à la mise en place de bons opérateurs génétiques, face à un problème donné, et un algorithme génétique est sans doute plus délicat à faire fonctionner du premier coup qu'un algorithme du recuit simulé.

Le grand avantage des algorithmes génétiques est qu'ils parviennent à trouver de bonnes solutions sur des problèmes très complexes, et trop éloignés des problèmes combinatoires classiques pour qu'on puisse tirer profit de certaines propriétés connues. Ils doivent simplement déterminer entre deux solutions quelle est la meilleure, afin d'opérer leurs sélections. On les emploie dans les domaines où un grand nombre de paramètres entrent en jeu, et où l'on a besoin d'obtenir de bonnes solutions en quelques itérations seulement – dans les systèmes de régulation de transport en temps réel par exemple. Par ailleurs, les algorithmes génétiques se prêtent bien, du fait de leur traitement simultané de solutions, à la

recherche d'**optimum multiples** : en créant une fonction de coût partagée, dont la valeur dépend partiellement de la distance entre les individus, on voit se former graduellement des sous-populations d'individus, qui se stabilisent autour des différents pics de la fonction objectif. C'est la technique du *nichage par la méthode du partage*.

## 2.6 LES ALGORITHMES DE COLONIES DE FOURMI

Comme les algorithmes génétiques, les algorithmes de colonies de fourmi font évoluer une population d' « agents », selon un modèle stochastique.

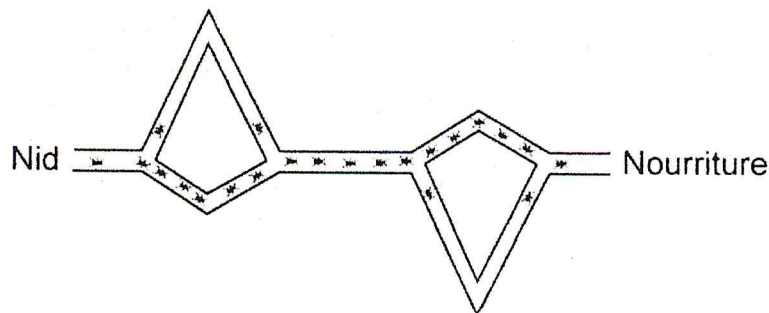


Schéma 6 : recueil de ressources par des fourmis

Cet algorithme est encore inspiré de la nature et de son organisation, et a été mis au point par Dorigo au début des années 90. Son principe repose sur le comportement particulier des fourmis qui, lorsqu'elles quittent leur fourmilière pour explorer leur environnement à la recherche de nourriture, finissent par élaborer des chemins qui s'avèrent fréquemment être les plus courts pour aller de la fourmilière à une source de nourriture intéressante.

Chaque fourmi laisse en effet derrière elle une traînée de phéromone à l'attention de ses congénères ; les fourmis choisissant avec une plus grande probabilité les chemins contenant les plus fortes concentrations de phéromones, il se forme ainsi ces « autoroutes » à fourmis, qui sillonnent le paysage. Ce mode de communication particulier, qui fait intervenir des modifications dans l'environnement, est appelé stigmergie.

### 2.6.1 Principe de l'algorithme

Le premier algorithme conçu selon ce modèle était destiné à résoudre le problème du voyageur de commerce, et c'est sous cette forme que nous le présentons dans les lignes suivantes. Nous dirons ensuite quelques mots de la généralisation à laquelle cet algorithme a donné lieu.

Le principe consiste à « lancer » des fourmis, et à les laisser élaborer pas à pas la solution, en allant d'une ville à l'autre. C'est donc un algorithme qui repose sur la *construction* progressive de solutions, un peu comme dans la méthode GRASP, qui inclue également une phase de construction. Afin de ne pas revenir sur ses pas, une fourmi tient à jour une liste Tabou, qui contient la liste des villes déjà visitées.



Soit A un ensemble de k fourmis :

Répéter :

Pour i=1 à k  
Construire le Trajet(i)

Fin Pour

Mettre à Jour Phéromones ()

Jusqu'à ce que le critère de terminaison soit satisfait.

Dans la procédure Construire Trajet (i), chaque fourmi se construit une route en choisissant les villes selon une règle de transition aléatoire très particulière :

Si  $p_{ij}^k(t)$  est la probabilité qu'à l'itération t la fourmi k choisisse d'aller de la ville i vers la ville j, alors on a :

$$p_{ij}^k(t) = \frac{(\tau_{ij}(t))^\alpha \cdot (\eta_{ij})^\beta}{\sum_{l \in J_i^k} (\tau_{il}(t))^\alpha \cdot (\eta_{il})^\beta} \quad \text{si } j \in J_i^k$$

$$p_{ij}^k(t) = 0 \quad \text{si } j \notin J_i^k$$

Où :  $\tau_{ij}(t)$  désigne le taux de phéromone sur la route ij à l'itération t

$\eta_{ij}$  Désigne l'inverse de la distance séparant les villes i et j.

$\alpha$  et  $\beta$  sont deux paramètres contrôlant respectivement l'influence du taux de phéromone sur le trajet ij, et l'influence de la distance sur le trajet ij

En d'autres termes, plus il y a de phéromone sur le trajet reliant deux villes, plus la probabilité est grande que la fourmi emprunte ce trajet. Mais ceci est contrebalancé par la longueur du trajet.  $\alpha$  et  $\beta$  permettent de régler l'effet de ces paramètres.

Lorsque toutes les fourmis ont construit une solution, la procédure *Mettre à Jour Phéromones* modifie les taux de phéromone  $\tau$  sur les routes en fonction des trajets effectivement empruntés par les fourmis, selon la formule :

$$\Delta \tau_{ij}^k(t) = \frac{Q}{L^k(t)} \quad \text{Si le trajet (i,j) est dans la tournée de la fourmi } k$$

$Q$  est une constante, et  $L^k(t)$  est la longueur totale de la tournée de la fourmi  $k$ . On constate donc que plus la route suivie par la fourmi a été courte, plus grande est donc la quantité de phéromone laissée derrière elle.

Pour éviter que des chemins ne se forment trop vite, et ne convergent trop rapidement vers des optima locaux, on introduit le concept d'évaporation des pistes de phéromone, au travers du paramètre  $\rho$  ( $0 < \rho \leq 1$ ) dans la formule complète de mise à jour suivante :

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t)$$

Avec  $\Delta\tau_{ij}(t) = \sum_{x=1}^k \Delta\tau_{ij}^x(t)$ , et  $k$  le nombre de fourmis.

Il est également possible d'introduire des modifications des taux de phéromone par un processus démon, indépendamment des itérations des fourmis. Ceci permet de biaiser l'évolution du système, afin d'introduire le résultat de certaines recherches locales, ou pour éviter des convergences prématurées. Les spécialistes conseillent de démarrer l'algorithme avec autant de fourmis qu'il y a de villes. L'initialisation du taux de phéromone initial est souvent donnée par la formule :

$\tau_0 = (n \cdot L_m)^{-1}$  Avec  $n$  le nombre de villes et  $L_m$  la longueur d'un tour trouvé par la méthode du plus proche voisin.

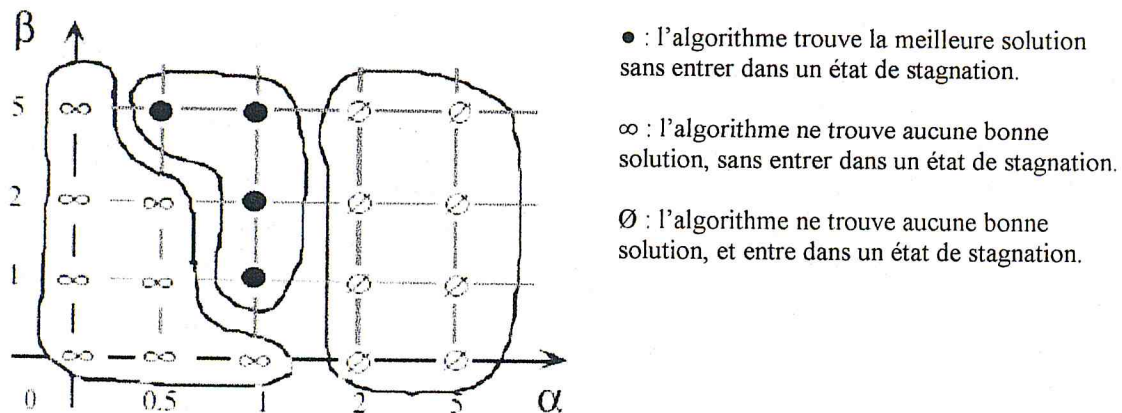


Schéma 7 : exemple d'influence des paramètres  $\alpha$  et  $\beta$

Le schéma ci-dessus montre qu'avec  $\alpha$  trop grand, les pistes de phéromone prennent trop d'importance, toutes les fourmis s'engagent systématiquement sur le même chemin et ne s'écartent plus : l'algorithme stagne.

## 2.6.2 Généralisation et variantes

L'algorithme de colonies de fourmi a pu s'adapter à d'autres problèmes que le voyageur de commerce, en attribuant une signification plus générale aux facteurs  $\tau$  et  $\eta$ .

$\eta_d$  Est la *force gloutonne* : plus cette valeur est grande, plus il semble intéressant de faire le choix  $d$ . En général, cette valeur est directement proportionnelle à la qualité de la solution partielle obtenue en prenant la décision  $d$ .

$\tau_d(t)$  Représente l'intérêt *historique* qu'a la fourmi de prendre la décision  $d$ . Plus cette quantité est grande, plus il a été intéressant dans le passé de prendre cette décision  $d$ . Cette grandeur  $\tau$  joue donc un rôle analogue à celui de la mémoire dans la recherche Tabou.

Dans le cadre du problème du voyageur de commerce,  $\tau_{ij}$  représente généralement l'intérêt qu'il y a de visiter la ville  $j$  après la ville  $i$ . Cet intérêt, nous l'avons vu, est naturellement évalué au regard de l'amélioration de la fonction objectif constatée après emprunt du trajet  $ij$ .

L'algorithme de colonies de fourmi a débouché sur une généralisation appelée *Ant Colony System*, plus efficace sur les problèmes de grande taille.

Elle introduit un certain nombre de modifications de fonctionnement :

La règle de transition aléatoire  $p_{ij}^k(t)$ , comme nous l'avons définie ci-dessus, n'est plus systématiquement appliquée. On laisse ainsi de temps en temps une fourmi choisir des arêtes uniquement parmi celles déjà visitées. Ceci permet de contraindre les fourmis à explorer davantage les régions où elles se trouvent.

La gestion des pistes de phéromone est également modifiée, et se fait de deux manières différentes :

- globalement, à la fin de la construction d'une solution
- pas à pas, de ville en ville

Une autre variante est l'algorithme *Max-Min*, qui introduit quelques particularités, dont les plus remarquables sont :

- seule la meilleure fourmi met à jour une piste de phéromone
- les valeurs des pistes sont bornées
- la mise à jour d'une piste phéromone est inversement proportionnelle à sa valeur : les pistes les plus fortes sont donc moins renforcées que les plus faibles

Ceci permet d'éviter que la probabilité de sélectionner certaines solutions ne finisse par devenir négatives, tout en évitant le surinvestissement de certaines pistes.

### 2.6.3 Avantages et Inconvénients

Bien que conçu au départ pour le problème du voyageur de commerce, l'algorithme de colonies de fourmi offre finalement beaucoup de souplesse, et il a été possible de l'adapter à tous les grands problèmes combinatoires classiques. Par exemple, avec le problème des tournées de véhicules, on utilise deux colonies de fourmi simultanément, chacune gérant ses propres pistes de phéromone : une colonie minimise le nombre de véhicules utilisés, et une autre minimise le coût total des tournées.

Par ailleurs, l'algorithme de colonies de fourmi se parallélise de façon très naturelle, en affectant par exemple un processus différent pour traiter la marche de chaque fourmi, et un autre pour mettre à jour les pistes de phéromones.

Il est également possible, nous l'avons vu, d'hybrider l'algorithme, en tenant compte du résultat de certaines recherches locales menées indépendamment de la marche des fourmis.

Une méthode d'hybridation avec un algorithme génétique a même été tentée récemment.

L'algorithme, de par son dynamisme intrinsèque, s'adapte aussi très bien aux espaces de solutions qui varient dynamiquement dans le temps.

Les algorithmes de colonies de fourmi ont été appliqués avec succès sur les problèmes d'affectation quadratique, de routage et de planification, et ils ont été la source d'inspiration de nouvelles métaheuristiques, comme celles des essais particuliers.

### 3. STRATEGIES DE RECHERCHE

Un certain nombre de propriétés se retrouvent dans toutes les métaheuristiques, et jouent un rôle prépondérant dans leur efficacité face à un problème donné.

#### 3.6.1 Intensification et diversification

Si les concepts d'intensification et de diversification ont d'abord été introduits par Glover au sujet de la recherche Tabou, ils n'en sont pas moins applicables à toute métaheuristique. Par intensification, on entend l'exploitation de l'information rassemblée par le système à un moment donné. C'est ce qui permet l'**amélioration de la valeur** d'une solution trouvée dans un certain voisinage. Par diversification, on entend au contraire l'**exploration de l'espace** de recherche encore inconnu. Elle est généralement obtenue par des processus stochastiques.

#### 3.6.3 Dans les algorithmes génétiques

Dans un sens, ils fonctionnent naturellement sur le versant « diversification », puisque l'utilisation d'une population de solutions tend déjà à améliorer l'étendue de l'espace de solutions examinée.

C'est l'opérateur de mutation qui permet d'accentuer la diversification de la recherche, en injectant une perturbation aléatoire dans la marche des cycles évolutionnaires. D'ailleurs, plus la mutation est sollicitée, plus la recherche équivaut à une exploration aléatoire de l'espace des solutions. Corrélativement, si l'on décide qu'à chaque nouvelle génération, on produit beaucoup d'enfants, et que l'on ne garde que peu de parents de la génération précédente, on va diversifier la recherche.

L'intensification est rendue possible par hybridation, en améliorant la valeur des individus par une recherche locale (on parle alors d'algorithmes *mémétiques*) ou bien en mettant en œuvre une stratégie de recombinaison « intelligente » des individus entre eux, comme nous l'avons vu avec le problème du voyageur de commerce, en sélectionnant des individus ayant des portions de trajets communs, et en recomposant le génome des trajets « enfant » par sélection des segments de trajets « parent ».

#### 3.6.4 Dans les algorithmes à colonies de fourmi

Ainsi que nous l'avons vu, la probabilité de choisir un certain trajet dépend de la formule :

$$p_{ij}^k(t) = \frac{(\tau_{ij}(t))^\alpha \cdot (\eta_{ij})^\beta}{\sum_{l \in J_i^k} (\tau_{il}(t))^\alpha \cdot (\eta_{il})^\beta}$$

Plus  $\alpha$  est grand, plus l'intensification est importante, car plus les pistes de phéromone ont d'influence sur le choix des trajets pris par les fourmis : la recherche est centrée autour des pistes de phéromone principales, là où on peut espérer trouver de bonnes solutions. Au contraire, plus  $\alpha$  est faible, plus la diversification sera forte, car c'est l'heuristique sur la longueur des arcs qui va alors plutôt guider les fourmis, réglée par le coefficient  $\beta$ .

L'algorithme Max-Min, avec les bornes qu'il impose sur les pistes de phéromones, permet d'éviter que certaines très bonnes solutions ne monopolisent la recherche, ou que certains trajets ne se voient affecter une probabilité négative et ne soient complètement écartés : ceci permet d'améliorer la diversification.

L'emploi de stratégies élitistes (qui font que les meilleures solutions contribuent davantage aux pistes) favorise l'intensification, alors qu'une réinitialisation de l'ensemble des pistes favorise l'exploration.

## CONCLUSION

Les métaheuristiques constituent une classe de méthodes approchées adaptables à un grand nombre de problèmes d'optimisation combinatoire. Mais, si l'on a pu constater leur grande efficacité sur de nombreuses classes de problèmes, il existe en revanche très peu de résultats permettant de comprendre la raison de cette efficacité, et aucune méthode particulière ne peut garantir qu'une métaheuristique sera plus efficace qu'une autre sur n'importe quel problème. Concrètement, certaines métaheuristiques présentent l'avantage d'être simples à mettre en œuvre, comme nous l'avons vu avec le recuit simulé ; d'autres sont plutôt bien adaptées à la résolution de certaines classes de problème, très contraints, comme le système de colonies de fourmis. La qualité des solutions trouvées par les métaheuristiques dépend de leur paramétrage (il faut éviter que les algorithmes ne convergent trop rapidement vers un optimum local), et de l'équilibre à trouver entre un balayage de tout l'espace des solutions (diversification) et une exploration locale poussée (l'intensification). Le choix d'une bonne représentation, d'un bon voisinage, sont également, nous l'avons dit, des facteurs influençant grandement l'efficacité de la méthode choisie, quelle qu'elle soit. Depuis une dizaine d'années, l'hybridation des métaheuristiques est devenue un enjeu important, à tel point que toutes les différentes métaheuristiques que nous avons examinées, recuit simulé, Tabou, algorithmes génétiques, sont maintenant perçues comme des points de départ au développement de nouveaux algorithmes d'optimisation. Les différentes caractéristiques de ces métaheuristiques (mémoires, listes de candidats, populations d'agents, transformations de la fonction objectif, etc.) sont recomposées entre elles pour former de nouvelles méthodes, dont la liste s'allonge. Et c'est bien là toute la difficulté que rencontre l'ingénieur non-spécialiste face à un problème d'optimisation combinatoire : doit-il consacrer son temps à passer en revue les abondants travaux de recherche qui existent sur le sujet, jusqu'à trouver l'hybridation optimale et le meilleur paramétrage connus à ce jour pour résoudre son problème ? Ne pourrait-il pas plutôt se concentrer sur la mise au point d'une heuristique spécifique, adaptée à sa situation ? Ou bien d'une métaheuristique classique – un « simple » recuit simulé par exemple, ou une recherche Tabou avec une implémentation judicieuse des mémoires à court et à long terme ?

Chaque situation est différente, et c'est en fonction du degré d'exigence de performance, et en fonction de contraintes économiques, c'est-à-dire selon le temps dont dispose l'ingénieur pour modéliser, programmer et tester son système, que se fera le choix de telle ou telle solution.

## II.1 Généralités sur les satellites artificiels

Les techniques spatiales ont modifié profondément les domaines de télécommunication, transmission des données, météorologie, géodésie et de positionnement précis à la surface de la terre. Avec l'envoi des satellites dans l'espace, les hommes découvrent leur planète de l'extérieur.

Le 26 août 1957 les soviétiques ont lancé grâce à une fusée, le Spoutnik 1, le premier satellite artificiel de la terre. En 1965 Mariner 4, le satellite Américain s'approchait de la planète Mars et en prenait les premières photos. En 1988, la sonde Pioneer 10 explore les confins du système solaires à plus de 15 milliards de Km.

Un *satellite artificiel* est un objet de fabrication humaine gravitant autour d'un corps céleste tel que la terre, le soleil ou tout autre planète. Il est généralement placé sur une orbite et est animé d'un mouvement périodique. Sa conception fait appel à des logiciels spécialisés. En général, on utilise pour sa construction des alliages légers à base d'aluminium, de magnésium, de titane ou des matériaux composites. Tous les modes de fabrication (usinage mécanique, moulage, forgeage) ou d'assemblage (soudage, rivetage, collage) sont utilisés.

### II.1.1 Classement des satellites [17]

Le classement des satellites est généralement établi en fonction du type d'orbite :

- **Les satellites géostationnaires** sont **des satellites géosynchrones** dont le plan de l'orbite est confondu avec celui de l'équateur, à une altitude de 36000 Km environ. Néanmoins **les satellites géosynchrones** sont des satellites dont la période de rotation autour de la terre est de 24 h, c'est-à-dire que leur vitesse angulaire de rotation est égale à celle de la terre ( $15^\circ/\text{Hr}$ ).
- **Les satellites à défilement** ont en général **des orbites basses** et survolent un grand nombre de régions du globe.
- Les satellites à **orbite héliosynchrone** ont des orbites dont la ligne des nœuds reste fixe par rapport au soleil. Leur période de rotation est d'environ 90 mn et leur inclinaison de  $95^\circ$ . L'utilité de telles orbites se justifie pour des satellites ayant besoin d'un éclairage particulier ou constant pour leurs panneaux solaires.
- Les satellites à **orbite semi-synchrone** ont une période de rotation de 12 h et sont surtout utilisés pour des missions d'aide à la navigation. leur inclinaison est de  $55^\circ$ .

### II.1.2 Domaines d'utilisation des satellites

- **Satellites d'astronomie** : Ils servent à l'observation permanente de l'espace, non seulement en lumière visible, mais également dans tout le spectre des longueurs d'onde électromagnétique. Ils permettent, par exemple, de mesurer la densité, la température d'un corps céleste. [6]
- **Satellites de navigation** : Ces satellites permettent de localiser des appels de détresse, de déterminer la position d'un navire ou d'établir la carte des courants marins d'une région donnée. [6]
- **Satellites météorologiques** : Classés sous deux catégories, *les satellites géostationnaires* et *les satellites à défilement*, ils sont destinés aux météorologues. Donnant une vision globale et permanente de la surface de notre planète ainsi que de son atmosphère, ils fournissent de précieuses indications sur la répartition des masses nuageuses et du rayonnement thermique terrestre pour comprendre les relations qui les lient. Ils permettent de prévoir à court ou moyen terme l'évolution du temps. [19]

- **Satellites de télécommunication** : Ces satellites sont utilisés pour relier différents points de la terre, par des communications de type téléphoniques, de transmission de données ou de diffusion de programmes télévisés.
- **Satellites militaires** : Ils sont utilisés pour la surveillance, la reconnaissance, l'alerte, l'aide à la navigation aérienne ou maritime.
- **Satellites de télédétection** : Ces satellites sont destinés à des fins scientifiques, économiques ou militaires. Leur vaste spectre d'observation permet de suivre par exemple l'évolution de la température des mers ou des océans, de suivre l'évolution des manteaux neigeux, comprendre les mécanismes de la sécheresse, prospector les ressources naturelles, élargir les domaines agricoles, etc. [19]
- **Satellites biologiques** : Ils sont destinés aux recherches bio-médicales, en rapport avec les vols cosmiques. Un satellite biologique est aménagé pour emporter des animaux et d'autres organismes vivants (plantes, bactéries, etc.).

## II.2 Structure d'un satellite [9]

Un satellite est composé en général de deux parties :

- a) **La charge utile** qui comprend les équipements nécessaires à la mission principale, la prise d'image. Elle a pour objectif de relayer, par des signaux radios, différents points de la surface terrestre en vue d'assurer une mission de durée donnée (téléphonie, télévision, transmission de donnée, etc.), appelée **durée de vie**, qui dimensionne la configuration, le type et les performances de la charge utile en relation avec les procédés de transmission fixés par les normes internationales.
- b) **La plate-forme** quand à elle, correspondait originellement à la structure destinée à supporter une ou plusieurs charges utiles et est équipée pour leur fournir les ressources nécessaires à leur fonctionnement dans des conditions requises. Elle est conçue pour assurer les fonctions suivantes :
  - *Le contrôle thermique* dont le rôle est le maintien de la température des équipements dans une plage spécifique et ce pendant toute la durée de vie du satellite afin de garantir leurs performances.
  - *La Protection contre les rayonnements et les corpuscules sachant que la structure porteuse et les mécanismes* comprennent les ensembles mécaniques assurant les fonctions d'interface avec le lanceur (statique et dynamique), de support des équipements stables dimensionnellement en orbite, après mise en forme (déploiements), et d'écran.
  - *Le maintien précis de l'orbite*. Du fait de forces parasites exercées sur le satellite (pression de radiation solaire par exemple), la trajectoire du satellite pourrait changer et il serait alors nécessaire d'effectuer cycliquement (ou non) des petites corrections à l'aide de micropropulseurs pour garder le satellite sur son itinéraire.
  - *L'alimentation électrique* qui distribue l'énergie nécessaire au fonctionnement de tous les systèmes électroniques embarqués.
  - *Le contrôle d'attitude* qui consiste en l'orientation angulaire du satellite dans l'espace par rapport à un système de référence.
  - *Le traitement, le stockage et la gestion des données à bord*.

## II.3 Contrôle d'attitude [10]

Un satellite artificiel est généralement doté d'un **système de contrôle d'attitude et d'orbite** (SCAO). Le **système de contrôle d'attitude** (SCA) désigne l'ensemble des équipements et algorithmes mis en œuvre de manière autonome sur un satellite artificiel afin de lui permettre de contrôler précisément son attitude, c'est-à-dire son orientation dans

l'espace autour de son centre de gravité. Ce contrôle se distingue du contrôle d'orbite qui consiste à maintenir le satellite sur l'orbite prévue. Néanmoins, le mouvement angulaire du satellite, n'a quasiment pas de conséquence sur l'orbite. L'orbite du satellite est définie par la vitesse initiale que lui communique le lanceur (Ariane), puis par les petites corrections effectuées régulièrement à l'aide de micropropulseurs.

### II.3.1 Les repères d'attitude

On peut distinguer les repères de la modélisation d'attitude suivants :

- **Le repère lié au satellite** : Ce repère est fixé au satellite (voire figure I.5), en fait l'attitude du satellite est définie par l'orientation de ce repère par rapport à celui de référence, donc la connaissance de son orientation est pour le SCAO, pour un bon pointage il faut que ce repère se confonde avec celui de référence.

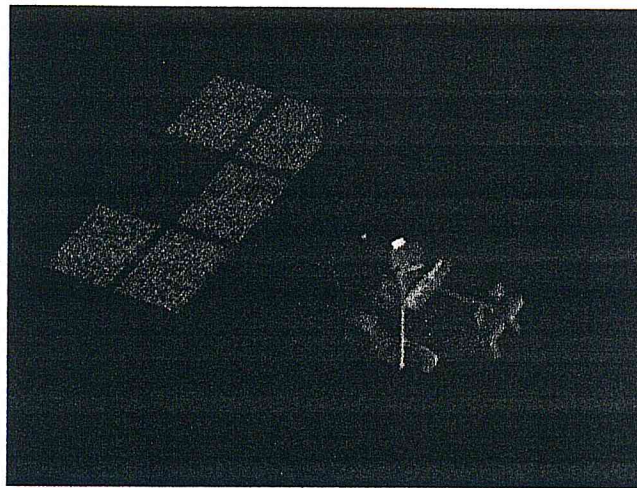


Figure I.5 : Axes du satellite.

- **Les repères de référence** : dont on cherche par rapport auxquels est caractérisé l'orientation du corps considéré. Pour ce contexte on cite trois repères :

#### a) Repères orbitaux locaux [25]

Un satellite stabilisé 3-axes, est asservi sur une orientation donnée pour chacune des trois directions de l'espace. L'une d'elles correspond à la direction satellite-centre terre, appelée "**direction géocentrique**"; la deuxième lui est perpendiculaire et est dans la direction de la vitesse du satellite; la troisième est perpendiculaire aux deux premières. Toutes les trois forment un trièdre direct et définissent le « **repère orbital local** ».

Le repère orbital local est défini en chaque point de l'orbite par les trois vecteurs unitaires construits à partir du **vecteur position** et du **vecteur vitesse** du satellite (figure I.6). En effet, le vecteur  $\vec{L}$  qui définit l'axe de *lacet* est colinéaire au vecteur position  $\vec{P}$  (sur l'axe centre terre-satellite). Le vecteur  $\vec{T}$  est perpendiculaire au plan de l'orbite formé par l'ensemble  $(\vec{L}, \vec{V})$  matérialise l'axe de *tangage*, tandis que  $\vec{R}$  complète le trièdre. Il appartient au plan  $(\vec{L}, \vec{V})$  pour définir l'axe de *roulis*. Il ne coïncide pas exactement avec le vecteur vitesse à cause de l'excentricité de l'orbite.



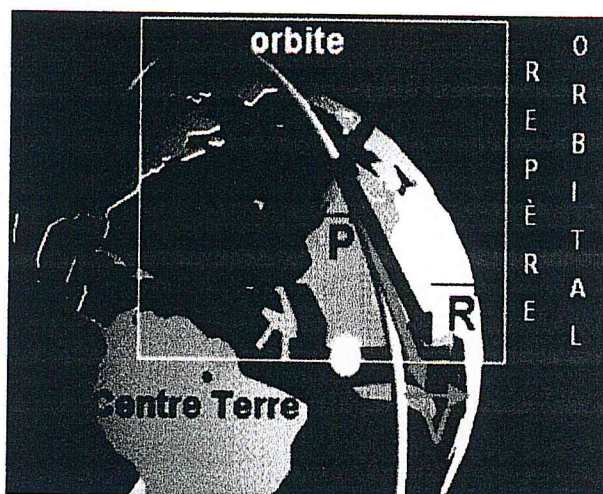


Figure I.6 : Repère orbital local.

b) **Repère solaire** : Il est lié à la direction Terre-Soliel, pour les études thermiques ou énergétiques.

c) **Repère inertiel (ECI)** : (Earth Centred Inertiel) a comme origine le centre de la Terre. L'axe x des abscisses est dans le plan équatorial et pointe l'équinoxe vernal, l'axe z se dirige vers le pôle nord de la terre et il est parallèle à son axe de rotation. L'axe y complète les deux autres axes pour avoir un repère direct. [5]

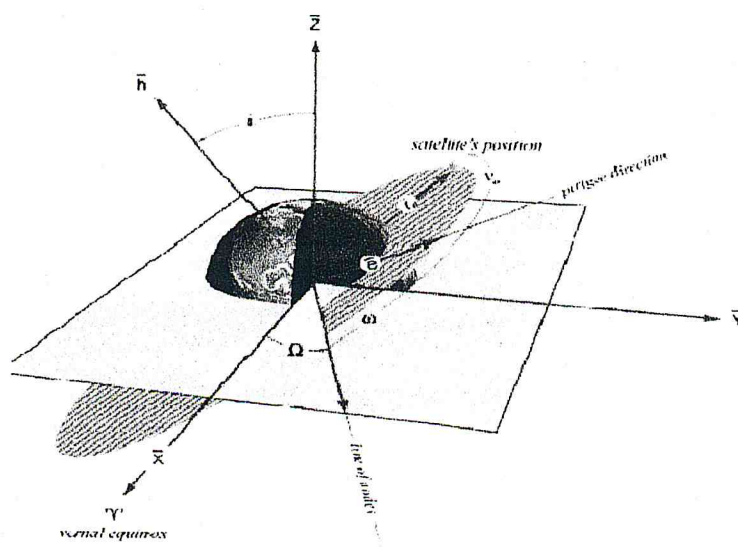


Figure I.7 : Repère inertiel.

- **Repère de consigne** : Se définit par rapport au repère de référence par la matrice de rotation permettant de passer du repère de consigne au repère satellite (angles d'Euler ou de Cardan). Il s'agit d'un repère purement vectoriel (sans point origine), dans la mesure où il ne concerne pas la position du satellite, qui relève du contrôle d'orbite, mais uniquement son attitude.

### II.3.2 Capteurs et actionneurs [5]

Afin d'assurer le contrôle permanent du satellite, une chaîne de senseurs est nécessaire pour assurer l'orientation de ce dernier. En effet, des **capteurs** asservissant des **actionneurs** qui maintiennent ou modifient selon les besoins, la direction du satellite grâce à un calculateur de bord. Les équipements suivants sont nécessaires aux commandes d'une mission :

### II.3.2.1 Les capteurs

- Une centrale inertielle composé de quatre *gyromètres* (*c'est un appareil qui permet l'acquisition de l'attitude du satellite à chaque instant*); ces derniers étant des gyromètres à deux axes, deux d'entre eux suffisent à délivrer des mesures de vitesse angulaire sur les trois axes du trièdre satellite. Les deux autres sont donc utilisés en secours.
- Deux *senseurs terrestres digitaux* (STD), un nominal et un redondant, permettent de mesurer les écarts angulaires autour des axes de tangage et de roulis.
- Deux *senseurs solaires digitaux* (SSD), un nominal et un redondant, permettent de mesurer les écarts angulaires autour de l'axe de lacet (une fois par orbite).

### II.3.2.2 Les actionneurs

- Trois roues à *réaction à palier magnétique* (RRPM) permettent d'appliquer un couple au satellite et ainsi, de provoquer une rotation autour de l'un des axes  $x_s$ ,  $y_s$ ,  $z_s$ .
- Deux *magnétocoupleurs* (MAC) qui, par interaction avec le champ magnétique terrestre, entraînent la création de couples ayant pour but de permettre le contrôle de la vitesse de rotation des roues à réaction.
- Des propulseurs (tuyères fonctionnant par combustion d'hydrazine) de deux types produisent chacun soit une force de 3.5 Newton, soit une force de 15 Newton; leur direction par rapport au centre de gravité du satellite induit une rotation autour de l'un des axes  $x_s$ ,  $y_s$  ou  $z_s$ .

Par ailleurs, au sous-ensemble SCAO, est rattaché le mécanisme d'entraînement du générateur solaire (GS). Il permet de contrôler la position du GS de façon à l'orienter au mieux vers le soleil en fonction du mouvement du satellite sur son orbite. L'éclairement recueilli par le générateur solaire sur la position éclairée de l'orbite est converti en énergie électrique pour assurer l'alimentation du satellite et la charge des batteries utilisées pour la portion en éclipse.

## II.4 Modélisation dynamique du satellite

Il est très important de développer un modèle qui représente le comportement dynamique de l'installation dans un but de dimensionnement, de simulation ou de commande. Pour notre travail nous avons présenté la modélisation d'attitude suivante.

### II.4.1 Principaux repères utilisés dans la modélisation

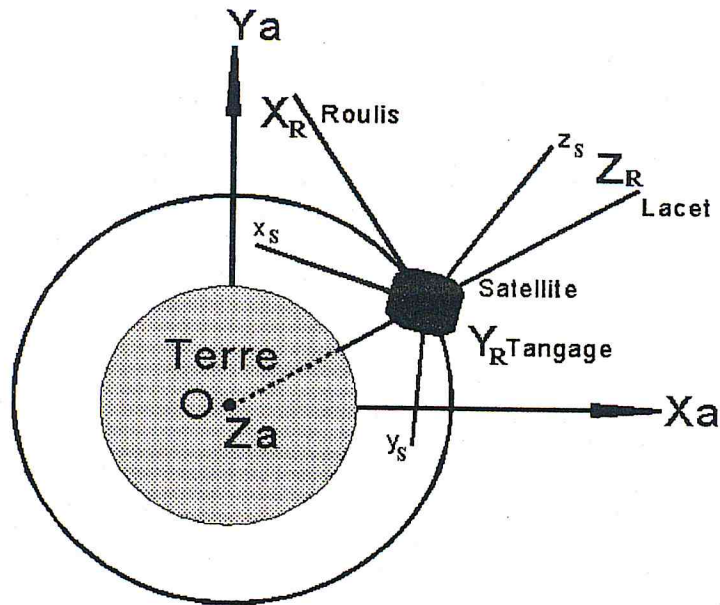


Figure II.8 : les principaux repères utilisés dans la modélisation d'attitude.

- ❖  $O X_a Y_a Z_a$  est un repère fixe galiléen, qu'il n'est pas utile de préciser outre mesure.
- ❖ Le satellite S est en orbite circulaire, de rayon  $r$ . On appelle repère orbital locale R le repère d'origine S et d'axes  $X_s Y_s Z_s$ , avec :
  - $X_s$  axe de *roulis*, de même sens et direction que la vitesse orbitale, il est tangent à l'orbite.
  - $Y_s$  axe de *tangage*, porte le moment cinétique, il est normal à la trajectoire.
  - $Z_s$  axe de *lacet*, suivant la géocentrique.
- ❖ S xyz le repère principal d'inertie pour le satellite, avec  $I_R, I_T, I_L$  les moments principaux d'inertie.

Nous avons ainsi défini les angles conventionnels, également appelés *Angles de Cardan*.

- Lacet  $\psi$  mesuré autour de Z.
- Tangage  $\theta$  mesuré autour de  $\beta$
- Roulis  $\varphi$  mesuré autour de x.

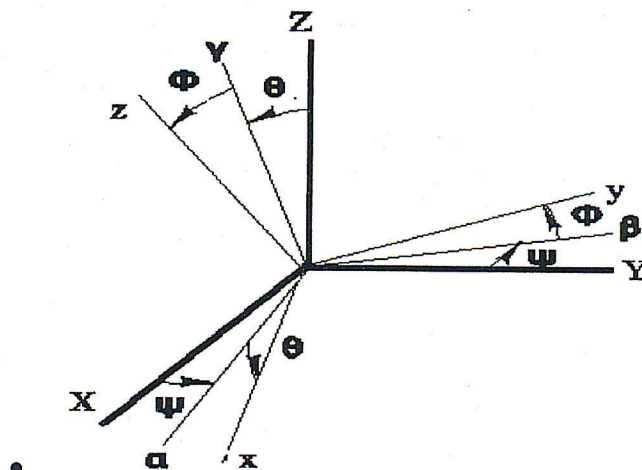


Figure I.9 : Orientation du repère lié au satellite.

Soit la matrice de rotation (passage) :

$$P_{R \rightarrow S} = \begin{pmatrix} x_S^{(R)} & y_S^{(R)} & z_S^{(R)} \end{pmatrix} = \begin{bmatrix} [X_R^{(S)}]^T \\ [Y_R^{(S)}]^T \\ [Z_R^{(S)}]^T \end{bmatrix} \quad (2.1)$$

Avec :

$$P_{S \rightarrow R} = P_{R \rightarrow S}^{-1} = P_{R \rightarrow S}^T \quad (2.2)$$

Pour trouver la matrice de passage nous avons suivi la succession des repères suivants :

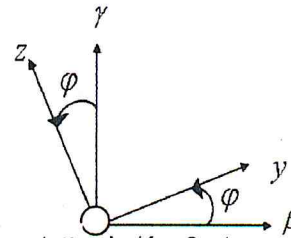
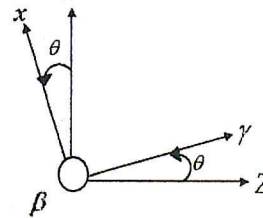
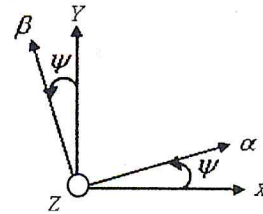
$$XYZ \xrightarrow{\psi} \alpha\beta Z \xrightarrow{\theta} x\beta\gamma \xrightarrow{\varphi} xyz$$

La projection de chaque vecteur dans le repère adéquat, assure que :

$$P_{R \rightarrow R'} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (I.3)$$

$$P_{R' \rightarrow R''} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \quad (I.4)$$

$$P_{R'' \rightarrow S} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{bmatrix} \quad (I.5)$$



Le passage du repère R vers S, se fait par rotation successive autour de Z, beta et x.

$$P_{R \rightarrow S} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{bmatrix} \quad (2.6)$$

$$P_{R \rightarrow S} = \begin{bmatrix} \cos\psi \cos\theta & \sin\theta \cos\psi \sin\varphi - \sin\psi \cos\varphi & \sin\theta \cos\psi \cos\varphi + \sin\psi \sin\varphi \\ \cos\theta \sin\psi & \sin\theta \sin\psi \sin\varphi + \cos\psi \cos\varphi & \sin\theta \sin\psi \cos\varphi - \cos\psi \sin\varphi \\ -\sin\theta & \cos\theta \sin\varphi & \cos\theta \cos\varphi \end{bmatrix} \quad (2.7)$$

Le déterminant de la matrice  $P_{R \rightarrow S}$  est  $-\cos\theta$ , l'indétermination est donc pour  $\theta = \frac{\pi}{2} + k\pi$ .

Le vecteur de rotation du repère satellite S autour du repère de référence R est donnée

$$\text{par : } \vec{\Omega}_{S/R} = \frac{d\theta}{dt} \cdot \vec{r}$$

$$\text{Aussi : } \vec{\Omega}_{S/R} = \vec{\Omega}_{R'/R} + \vec{\Omega}_{R''/R'} + \vec{\Omega}_{S/R''} = \dot{\psi} \vec{Z} + \dot{\theta} \vec{\beta} + \dot{\phi} \vec{x} \quad (2.8)$$

$$\begin{aligned} \vec{\Omega}_{S/R}^{(R)} &= \dot{\psi} \vec{Z}_R + \dot{\theta} (-\sin \psi \vec{X}_R + \cos \psi \vec{Y}_R) + \dot{\phi} (\cos \theta \vec{\alpha} - \sin \theta \vec{Z}_R) \\ &= \dot{\psi} \vec{Z}_R + \dot{\theta} (-\sin \psi \vec{X}_R + \cos \psi \vec{Y}_R) + \dot{\phi} \cos \theta (\cos \psi \vec{X}_R + \sin \psi \vec{Y}_R) - \dot{\phi} \sin \theta \vec{Z}_R \end{aligned}$$

$$\vec{\Omega}_{S/R}^{(R)} = \begin{bmatrix} -\dot{\theta} \sin \psi + \dot{\phi} \cos \theta \cos \psi \\ \dot{\theta} \cos \psi + \dot{\phi} \cos \theta \sin \psi \\ \dot{\psi} - \dot{\phi} \sin \theta \end{bmatrix} \quad (2.9)$$

Pour  $\vec{\Omega}_{S/R}^{(R)}$  nous avons :

$$\begin{aligned} \vec{\Omega}_{S/R}^{(S)} &= \dot{\psi} \vec{Z}_R + \dot{\theta} \vec{\beta} + \dot{\phi} \vec{x}_S \\ &= \dot{\psi} (-\sin \theta \vec{x}_S + \cos \theta \vec{y}) + \dot{\theta} (\cos \varphi \vec{y}_S - \sin \varphi \vec{z}_S) + \dot{\phi} \vec{x}_S \\ &= \dot{\psi} (-\sin \theta \vec{x}_S + \cos \theta (\sin \varphi \vec{y}_S + \cos \varphi \vec{z}_S)) + \dot{\theta} (\cos \varphi \vec{y}_S - \sin \varphi \vec{z}_S) + \dot{\phi} \vec{x} \end{aligned} \quad (2.10)$$

$$\vec{\Omega}_{S/R}^{(S)} = \begin{bmatrix} \dot{\phi} - \dot{\psi} \sin \theta \\ \dot{\psi} \cos \theta \sin \varphi + \dot{\theta} \cos \varphi \\ \dot{\psi} \cos \theta \cos \varphi - \dot{\theta} \sin \varphi \end{bmatrix} \quad (2.11)$$

Le vecteur rotation instantanée de R par rapport à un repère inertiel d'origine terre est :

$$\vec{\Omega}_{R/R_i}^{(R)} = -\omega_0 \vec{Y}$$

$$(2.12)$$

donc :

$$\vec{\Omega}_{S/R_i}^{(S)} = \vec{\Omega}_{S/R}^{(S)} + \vec{\Omega}_{R/R_i}^{(S)} = \vec{\Omega}_{S/R}^{(S)} + \vec{P}_{R \rightarrow S}^T \cdot \vec{\Omega}_{R/R_i}^{(R)} \quad (2.13)$$

tel que :

$$P_{R \rightarrow S}^T \cdot \vec{\Omega}_{R/R_i}^{(R)} = \begin{bmatrix} \cos \psi \cos \theta & \sin \psi \cos \theta & -\sin \theta \\ -\sin \psi \cos \varphi + \sin \theta \cos \psi \sin \varphi & \cos \psi \cos \varphi + \sin \theta \sin \psi \sin \varphi & \sin \varphi \cos \theta \\ \sin \varphi \sin \psi + \sin \theta \cos \psi \cos \varphi & -\cos \psi \sin \varphi + \sin \theta \sin \psi \cos \varphi & \cos \theta \cos \varphi \end{bmatrix} \begin{bmatrix} 0 \\ -\omega_0 \\ 0 \end{bmatrix}$$

$$\vec{\Omega}_{R/R_i}^{(S)} = P_{R \rightarrow S}^T \cdot \vec{\Omega}_{R/R_i}^{(R)} = \begin{bmatrix} -\omega_0 \sin \psi \cos \theta \\ -\omega_0 (\cos \psi \cos \varphi + \sin \theta \sin \psi \sin \varphi) \\ \omega_0 (\cos \psi \sin \varphi - \sin \theta \sin \psi \cos \varphi) \end{bmatrix} \quad (2.14)$$

Donc :

$$\vec{\Omega}_{S/R_i}^{(S)} = \begin{bmatrix} \dot{\phi} - \dot{\psi} \sin \theta - \omega_0 \sin \psi \cos \theta \\ \dot{\psi} \cos \theta \sin \varphi + \dot{\theta} \cos \varphi - \omega_0 (\cos \psi \cos \varphi + \sin \theta \sin \psi \sin \varphi) \\ \dot{\psi} \cos \theta \cos \varphi - \dot{\theta} \sin \varphi + \omega_0 (\cos \psi \sin \varphi - \sin \theta \sin \psi \cos \varphi) \end{bmatrix} \quad (2.15)$$

Exprimé dans les axes satellites, le vecteur « rotation instantanée » absolue a pour composantes (celles qui seraient mesurées par des gyroscopes) :

$$\begin{aligned} p &= \dot{\varphi} - \dot{\psi} \sin \theta - \omega_0 \sin \psi \cos \theta \\ q &= \dot{\psi} \cos \theta \sin \varphi + \dot{\theta} \cos \varphi - \varphi_0 (\cos \psi \cos \varphi + \sin \theta \sin \psi \sin \varphi) \\ r &= \dot{\psi} \cos \theta \cos \varphi - \dot{\theta} \sin \varphi + \omega_0 (\cos \psi \sin \varphi - \sin \theta \sin \psi \cos \varphi) \end{aligned} \quad (2.16)$$

En se basant sur la mécanique générale et le théorème du moment cinétique appliquée au satellite en son centre d'inertie et projeté sur S. on obtient que :

$$\left[ \frac{d\vec{H}_G}{dt} \right]_S = \left[ \frac{d\vec{H}_G}{dt} \right] + \vec{\Omega} \wedge \vec{H}_G = \vec{M}_G \quad (2.17)$$

Où : 
$$\vec{H}_G = \vec{I} \cdot \vec{\Omega} \quad (2.18)$$

En remplaçant l'expression de  $\vec{H}_G$  dans (IV.18), on trouve :

$$\begin{cases} I_1 \frac{dp}{dt} + (I_3 - I_2)qr = L \\ I_2 \frac{dq}{dt} + (I_1 - I_3)pr = M \\ I_3 \frac{dr}{dt} + (I_2 - I_1)pq = N \end{cases} \quad (2.19)$$

### II.4.2 Couples de gradient de gravité

Tout point du satellite est soumis à l'attraction gravitationnelle, dont l'intensité varie en fonction de la distance r au centre de la terre.

Le centre de gravité (barycentre des forces de gravité) du satellite ne coïncide donc pas forcément avec son centre de masse (ou d'inertie, barycentre des masses élémentaires), ce qui provoque l'apparition d'un couple dont l'expression dépend au bout du compte des moments d'inertie du satellite.

#### II.4.2.1 Modélisation en présence du couple de gradient de gravité

L'expression du couple de gradient de gravité est donnée par :

$$\vec{\Gamma} / \vec{x}\vec{y}\vec{z} = \frac{3k}{r^3} \vec{Z} \wedge (\vec{I} \cdot \vec{Z}) \quad (2.20)$$

Tel que

k : constante gravitationnelle de la terre  $3.986005 \cdot 10^{14} \text{ m}^3/\text{s}^2$

r : distance du centre de la terre au centre de masse du satellite .

I : tenseur d'inertie du satellite.

Dans le cas d'une orbite circulaire  $\omega_0^2 = \frac{k}{r^3}$

Nous avons :

$$P_{R \rightarrow S} = \begin{bmatrix} \cos \psi \cos \theta & \sin \theta \cos \psi \sin \varphi - \sin \psi \cos \varphi & \sin \theta \cos \psi \cos \varphi + \sin \psi \sin \varphi \\ \cos \theta \sin \psi & \sin \theta \sin \psi \sin \varphi + \cos \psi \cos \varphi & \sin \theta \sin \psi \cos \varphi - \cos \psi \sin \varphi \\ -\sin \theta & \cos \theta \sin \varphi & \cos \theta \cos \varphi \end{bmatrix}$$

$$P_{R \rightarrow S}^T = \begin{bmatrix} \cos \psi \cos \theta & \cos \theta \sin \psi & -\sin \theta \\ \sin \theta \cos \psi \sin \varphi - \sin \psi \cos \varphi & \sin \theta \sin \psi \sin \varphi + \cos \psi \cos \varphi & \cos \theta \sin \varphi \\ \sin \theta \cos \psi \cos \varphi + \sin \psi \sin \varphi & \sin \theta \sin \psi \cos \varphi - \cos \psi \sin \varphi & \cos \theta \cos \varphi \end{bmatrix}$$

En appliquant donc :

$$\vec{\Gamma} / \vec{xyz} = 3\omega_0^2 \vec{Z} \wedge \vec{I} \cdot \vec{Z} = 3\omega_0^2 (P^T \vec{z} \wedge \vec{I} \cdot P^T \vec{z}) \quad (2.21)$$

Nous trouvons :

$$P^T \vec{z} = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \theta \cos \psi \sin \varphi - \sin \psi \cos \varphi & \sin \theta \sin \psi \sin \varphi + \cos \psi \cos \varphi & \cos \theta \sin \varphi \\ \sin \theta \cos \psi \cos \varphi + \sin \psi \sin \varphi & \sin \theta \sin \psi \cos \varphi - \cos \psi \sin \varphi & \cos \theta \cos \varphi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$P^T \vec{z} = \begin{pmatrix} -\sin \theta \\ \cos \theta \sin \varphi \\ \cos \theta \cos \varphi \end{pmatrix} \quad (2.22)$$

Avec :

$$\vec{I} \cdot P^T \vec{z} = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix} \begin{bmatrix} -\sin \theta \\ \cos \theta \sin \varphi \\ \cos \theta \cos \varphi \end{bmatrix} = \begin{bmatrix} -I_1 \sin \theta \\ I_2 \cos \theta \sin \varphi \\ I_3 \cos \theta \cos \varphi \end{bmatrix} \quad (2.23)$$

Donc :

$$P^T \vec{z} \wedge \vec{I} \cdot P^T \vec{z} = \begin{bmatrix} (I_3 - I_2) \cos^2 \theta \cos \varphi \sin \varphi \\ (I_3 - I_1) \cos \theta \cos \varphi \sin \theta \\ (I_1 - I_2) \cos \theta \sin \varphi \sin \theta \end{bmatrix} \quad (2.24)$$

En conclusion nous avons :

$$\vec{\Gamma} / \vec{xyz} = 3\omega_0^2 \vec{Z} \wedge \vec{I} \cdot \vec{Z} = 3\omega_0^2 \begin{bmatrix} (I_3 - I_2) \cos^2 \theta \cos \varphi \sin \varphi \\ (I_3 - I_1) \cos \theta \cos \varphi \sin \theta \\ (I_1 - I_2) \cos \theta \sin \varphi \sin \theta \end{bmatrix} = \begin{bmatrix} 3\omega_0^2 (I_3 - I_2) \cos^2 \theta \cos \varphi \sin \varphi \\ 3\omega_0^2 (I_3 - I_1) \cos \theta \cos \varphi \sin \theta \\ 3\omega_0^2 (I_1 - I_2) \cos \theta \sin \varphi \sin \theta \end{bmatrix} \quad (2.25)$$

On peut écrire I sous la forme suivante :

$$I = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix} = \begin{bmatrix} I_R & 0 & 0 \\ 0 & I_T & 0 \\ 0 & 0 & I_L \end{bmatrix}$$

On trouve :

$$\vec{\Gamma} / S = \begin{bmatrix} 3\omega_0^2 (I_L - I_T) \cos^2 \theta \cos \varphi \sin \varphi \\ 3\omega_0^2 (I_L - I_R) \cos \theta \cos \varphi \sin \theta \\ 3\omega_0^2 (I_R - I_T) \cos \theta \sin \varphi \sin \theta \end{bmatrix} \quad (2.26)$$

**II.4.2.2 Modélisation du mouvement libre du satellite en présence du couple de gradient de gravité ( $\Gamma$ ) et soumis à des moments de perturbations externes ( $M_P$ )**

En se basant sur la mécanique générale et le théorème du moment cinétique appliqué au satellite en son centre d'inertie et projeté sur R, en tenant compte uniquement du couple de gradient de gravité tel que  $\vec{\Gamma}$  représente ce couple.

$$\begin{cases} I_R \frac{dp}{dt} + (I_L - I_T)qr = \Gamma_x + M_{Px} \\ I_T \frac{dp}{dt} + (I_R - I_L)pr = \Gamma_y + M_{Py} \\ I_L \frac{dr}{dt} + (I_T - I_R)pq = \Gamma_z + M_{Pz} \end{cases} \quad (2.27)$$

Des expressions (II.26) et (II.27) on obtient :

$$\begin{cases} I_R \frac{dp}{dt} + (I_L - I_T)qr = \Gamma_x + M_x = 3\omega_0^2 (I_L - I_T) \cos^2 \theta \cos \varphi \sin \varphi \\ I_T \frac{dq}{dt} + (I_R - I_L)pr = \Gamma_y + M_y = -3\omega_0^2 (I_R - I_L) \cos \theta \cos \varphi \sin \theta \\ I_L \frac{dr}{dt} + (I_T - I_R)pq = \Gamma_z + M_z = -\omega_0^2 (I_T - I_R) \cos \theta \sin \varphi \sin \theta \end{cases} \quad (2.28)$$

Ces équations représentent les équations du mouvement libre du satellite. C'est-à-dire sous la seule action du gradient de gravité, sans commande.

Nous avons :  $\vec{\Omega}_{S/R_i} = \vec{\Omega}_{S/R} + \vec{\Omega}_{R/R_i}$

Nous avons vu que :

$$\begin{aligned} p &= \dot{\varphi} - \dot{\psi} \sin \theta - \omega_0 \sin \psi \cos \theta \\ q &= \dot{\psi} \cos \theta \sin \varphi + \dot{\theta} \cos \varphi - \varphi_0 (\cos \psi \cos \varphi + \sin \theta \sin \psi \sin \varphi) \\ r &= \dot{\psi} \cos \theta \cos \varphi - \dot{\theta} \sin \varphi + \omega_0 (\cos \psi \sin \varphi - \sin \theta \sin \psi \cos \varphi) \end{aligned} \quad (2.29)$$

On remplace dans (IV.28) on trouve :

Ou le moment total des perturbations extérieures autres que gravitationnelles est sous la forme :

$$M_P = \begin{bmatrix} M_{Px} \\ M_{Py} \\ M_{Pz} \end{bmatrix} \quad (2.30)$$

$$\begin{cases} I_R \ddot{\varphi} - I_R \omega_0 \dot{\psi} + (I_L - I_T) (-\omega_0 + \dot{\theta}) (\dot{\psi} + \omega_0 \varphi) - 3\omega_0^2 (I_L - I_T) \cos^2 \theta \cos \varphi \sin \varphi = M_{Cx} + M_{Px} \\ I_T \ddot{\theta} + (I_R - I_L) (\dot{\varphi} - \omega_0 \psi) (\dot{\psi} + \omega_0 \varphi) + 3\omega_0^2 (I_R - I_L) \cos \theta \cos \varphi \sin \theta = M_{Cy} + M_{Py} \\ I_L \ddot{\psi} + I_L \omega_0 \dot{\varphi} + (I_T - I_R) (-\omega_0 \psi + \dot{\varphi}) (-\omega_0 + \dot{\theta}) + 3\omega_0^2 (I_T - I_R) \cos \theta \sin \varphi \sin \theta = M_{Cz} + M_{Pz} \end{cases} \quad (2.31)$$



### III.1 Introduction

La *logique floue* vise l'univers intuitif du réglage et de la commande de processus. Formulée en 1965 par Lotfi A. ZADEH, cette théorie permet, en dehors des méthodes de réglages conventionnelles, basées sur une modélisation appropriée du système à régler, d'inclure des résultats d'expérience, permettant d'obtenir des lois de réglage assez efficaces de systèmes, sans forcément passer par des études analytiques approfondies décrivant le procédé par des fonctions de transfert ou d'équations d'état associées, ceci grâce à une symbolique mathématique.

### III.2 Principe fondamental de la logique floue [12]

Selon la logique classique (logique de boole), qui n'admet pour les variables que les deux valeurs 0 et 1, la logique de classification floue est analogique.

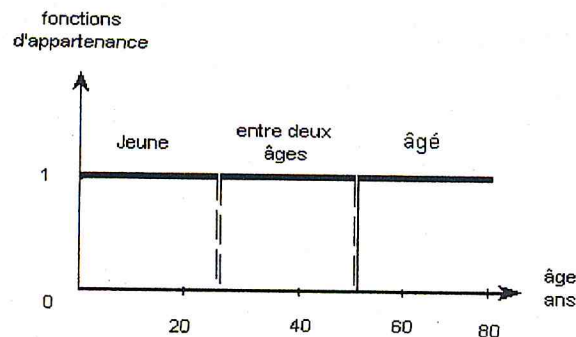


Figure III.1 : Classification des personnes en trois sous-ensembles selon la logique classique.

La logique floue, dont les variables peuvent prendre n'importe quelle valeur entre 0 et 1 (on parle alors de *fonction d'appartenance*  $\mu$ ), permet de tenir compte de cette réalité. Les limites ne varient pas soudainement, mais progressivement.

La logique floue possède des règles de calcul bien précises permettant de combiner plusieurs *fonctions d'appartenance* et de tirer des conclusions pondérées.

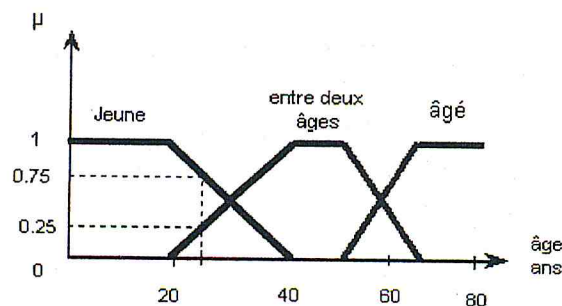
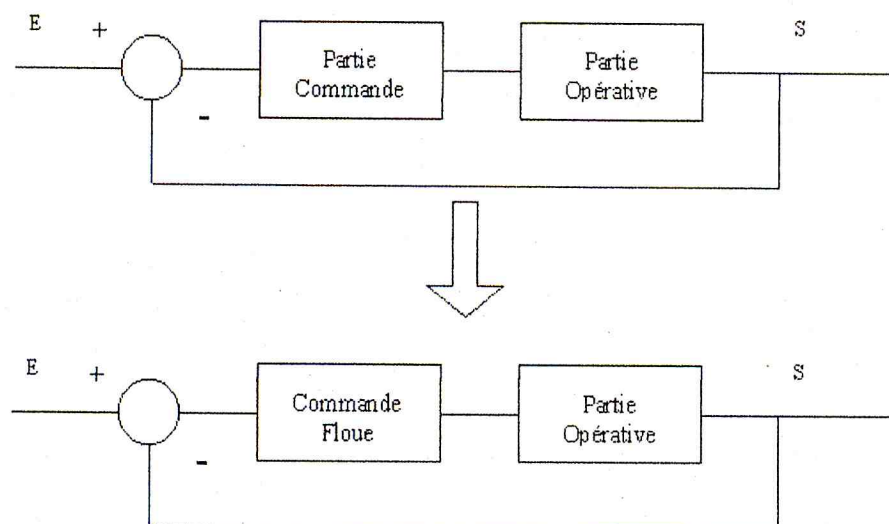


Figure III.2 : Classification des personnes en trois sous-ensembles selon la logique floue.

### III.3 Les différents domaines d'application [12] [13] [21] [22]

La commande floue est l'application la plus utilisée de la logique floue. Elle consiste à remplacer les algorithmes de réglage conventionnels par des réglages linguistiques du types : SI..., ALORS..... Ainsi on obtient un algorithme linguistique qui se prête mieux que les méthodes traditionnelles à la commande d'un processus.



Ainsi, il n'y a en fait pas de domaine où l'on ne puisse appliquer les raisonnements propres à la logique floue car elle a été conçue pour s'adapter aux techniques de pensée humaine.

Domaine d'application	Exemples
Automatisme	ateliers de production, machines-outils, automates programmables, pilotes d'avion, trains automatiques, machine à laver, caméras, etc.
Robotique	Commande de robots très divers : aide aux mouvements et à la détection des formes.
Informatique	Systèmes experts, base de données relationnelles, langage de programmation floue.
Engineering	Analyse et évaluation des systèmes, contrôle de circulation (aérienne ou automobile), télécommunications, etc.
Gestion et prise de décision	Théorie des jeux, gestion de personnel, lancement de nouveaux produits et d'investissements, etc.

Environnement	Météorologie (prévisions, évaluation des variations climatiques), évaluations sismiques, contrôle de processus dans l'industrie chimique ou agro-alimentaire.
Médecine	Système expert d'aide aux diagnostics comme par exemple la MYCIN en 1974.
Mathématiques appliquées	Statistiques, recherche opérationnelle (programmation linéaire PERT, files d'attente, etc.)
Sciences humaines	Sociologie, psychologie.

Tableau III.1 : les domaines d'application de la logique floue.

### III.4 Bases générales de la logique floue

Les éléments de base de la logique floue sont :

#### III.4.1 Les sous-ensemble Flou [7] [15] [18]

Un *sous-ensemble flou* est défini, sur le *domaine de variation*  $U$  de  $x$ , par le triplet  $\{A, \mathcal{A}, \mu_A\}$ , où :

- $A$  est un *sous-ensemble* de  $U$ .
- $\mathcal{A}$ , un label linguistique, caractérisant qualitativement une partie des valeurs de  $x$ .
- $\mu_A$  une *fonction d'appartenance* de  $x \in U \rightarrow \mu_A(x) \in [0, 1]$ , qui donne le *degré d'appartenance* d'une observation de  $x$  au *sous-ensemble flou*  $\{A\}$ .

#### III.4.2 Variables floues linguistiques [13] [18]

La description d'une certaine situation, d'un phénomène ou d'un procédé passe par des expressions comme : Quelque, beaucoup, souvent, chaud, froid, rapide, lent, grand, petit,...

Ce genre de formulation, forme les *valeurs des variables linguistiques* de la logique floue. Leur formulation sous forme de définition en terme de *fonctions d'appartenance*, permettra leur traitement numérique.

##### III.4.2.1 Formes des fonctions d'appartenance [12] [14]

Le plus souvent, les *fonctions d'appartenance* utilisées sont de formes trapézoïdales ou triangulaires, car d'expression aisées (morceaux de droites).

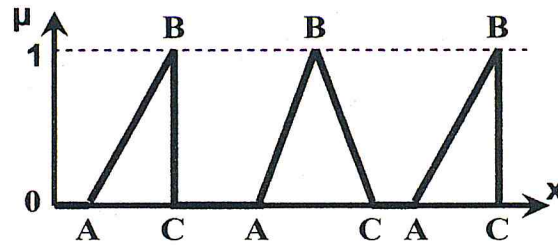


Figure III.3 : Fonctions d'appartenance de forme triangulaire.

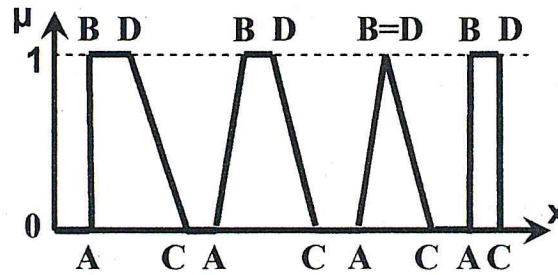
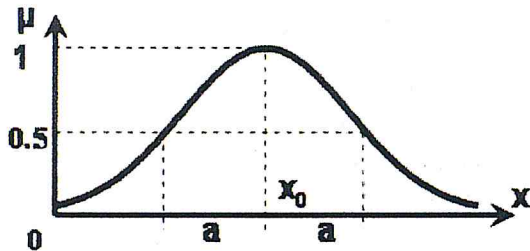


Figure III.4 : Fonction d'appartenance de forme trapézoïdale.

Cependant, d'autres formes existent, telles que la forme en cloche (figure ci-dessous) :



$$\mu(x) = \frac{1}{1 + \left(\frac{x - x_0}{a}\right)^2} ; -\infty < x < +\infty \quad (III.1)$$

Figure III.5 : Fonction d'appartenance en forme Gaussienne selon la relation (III.1)

Enfin, il est possible de composer la fonction d'appartenance par des segments de droites. Ainsi, on peut réaliser une forme concave et convexe, comme la montre la figure III.6.

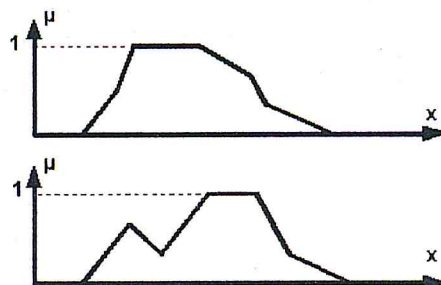
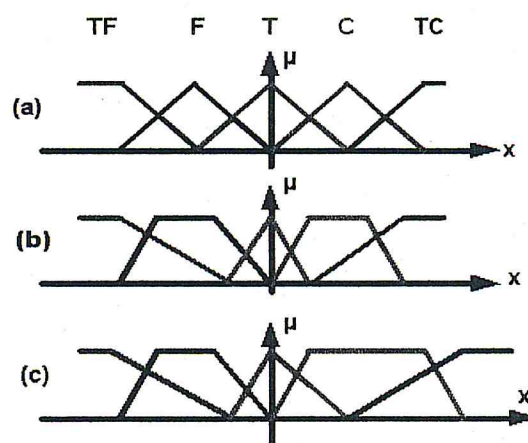


Figure III.6 : Fonctions d'appartenance composées de segments de droites (fonction convexe et concave).

### III.4.2.2 Considérations générales sur les fonctions d'appartenances [22]

Les fonctions d'appartenance peuvent être symétriques et distribuées de manière équilibrée. Une forme est définie symétrique lorsque les fonctions d'appartenances sont symétriques par rapport  $x=0$ . Par contre, la forme est définie équilibrée lorsque les maxima des fonctions d'appartenance des différents ensembles sont écartés de manière équilibrée. On peut aussi choisir les fonctions d'appartenance de manière non symétriques et non équilibrées, ceci pour un comportement optimale du réglage.



**Figure III.7 :** Structure des fonctions d'appartenance :  
 (a) symétrique et équilibrée ; (b) symétrique et non équilibrée ;  
 (c) non symétrique et non équilibrée.

### III.4.3 Dédutions floues (Inférences) [12]

En général, plusieurs valeurs de variables linguistiques, convenablement définies par des fonctions d'appartenance, sont liées entre elles par des règles, afin de tirer des conclusions. On parle alors de *dédutions floues* ou d'*inférences*. Dans ce contexte, on peut distinguer deux genres de règles d'inférence.

#### III.4.3.1 Inférence avec une seule règle [12]

Le cas d'une inférence avec une seule règle se présente lorsqu'il faut comparer plusieurs concurrents (objet ou personnes) dans une certaine situation et en choisir l'optimum. Les variables qui déterminent la situation sont des variables linguistiques (variable floues). On trouve cette problématique essentiellement dans les domaines non techniques, où il faut

prendre une décision, comme par exemple l'achat d'un appareil, le recrutement d'un employé etc.

Si  $x_1, x_2, \dots, x_n$  sont les variables linguistiques pour évaluer une certaine situation, le critère  $Y$  pour la prise de décision peut être formulé par une règle, comme par exemple :

$$Y = [x_1 \text{ OU } (x_2 \text{ ET } x_3) \text{ OU } \dots] \text{ ET } x_n \quad (\text{III.2})$$

La forme exacte de cette règle dépend évidemment du problème en question. Elle est subjective et possède souvent un certain caractère arbitraire.

A chaque variable formant un ensemble est attribuée une fonction d'appartenance  $\mu_{x_1}, \mu_{x_2}, \dots, \mu_{x_m}$ , dont les valeurs numériques (facteurs d'appartenance) sont en général différentes pour chaque concurrent.

### III.4.3.2 Inférence avec plusieurs règles [12]

Le cas d'une inférence avec plusieurs règles se présente lorsqu'une ou plusieurs variables nécessitent une prise de décision différente suivant les valeurs qu'atteignent ces variables. Ces dernières sont des variables linguistiques. Cette problématique se présente essentiellement pour des problèmes de réglage et de commande et la prise de décision conduit à l'exécution d'une certaine opération.

Les règles peuvent alors être exprimées sous la forme générale suivante :

$$\begin{aligned} \text{Opération} := & \text{ Si condition 1, Alors opération 1, OU} \\ & \text{ Si condition 2, Alors opération 2, OU} \\ & \vdots \\ & \text{ Si condition m, Alors opération m.} \end{aligned} \quad (\text{III.3})$$

Les conditions peuvent dépendre d'une ou de plusieurs variables. A chaque variable sont attribuées des fonctions d'appartenance, tenant compte des ensembles flous formés par ces variables. En général, il faut aussi introduire des fonctions d'appartenances pour les opérateurs, surtout si ces dernières sont exprimées par des variables floues.

### III.4.4.3 Opérateur OU

Le sous-ensemble  $A \text{ OU } B$ ,  $A \cup B$  est représenté par la fonction d'appartenance :

$$\mu_{A \cup B}(z) = \max\{\mu_A(x), \mu_B(y)\}; \forall x, y, z \in U$$

(III.6)

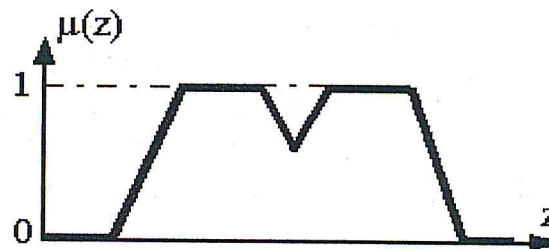


Figure III.10 : Opérateur OU.

D'une manière générale l'opérateur ET peut être représenté par une norme triangulaire (T-norme) définie de la façon suivante :

$$\begin{aligned} T : [0,1] \times [0,1] &\rightarrow [0,1] \\ x, y &\rightarrow z = x T y \end{aligned}$$

Tel que T soit :

Commutative :  $x T y = y T x$ .

Associative :  $(x T y) T z = x T (y T z)$ .

Non décroissante par rapport aux arguments :  $\forall (x \leq y, w \leq z), x T w \leq y T z$

Admette 0 comme élément absorbant et 1 comme élément neutre :

$$0 T x = x T 0 = 0 \text{ et } 1 T x = x T 1 = x$$

De même, l'opérateur OU peut être défini par la donnée d'une S-norme ou T-conorme :

$$\begin{aligned} S : [0,1] \times [0,1] &\rightarrow [0,1] \\ x, y &\rightarrow z = x S y \end{aligned}$$

Et satisfaisant les trois premiers critères de la T-norme et en admettant 1 comme élément absorbant et 0 comme élément neutre :

$$0 S x = x S 0 = x \text{ et } 1 S x = x S 1 = 1$$

Cependant dans certaines circonstances, il peut être judicieux d'utiliser d'autres opérateurs, soit pour simplifier le traitement numérique, soit pour mieux tenir compte des opérations floues.

### III.5 Structure d'une commande floue

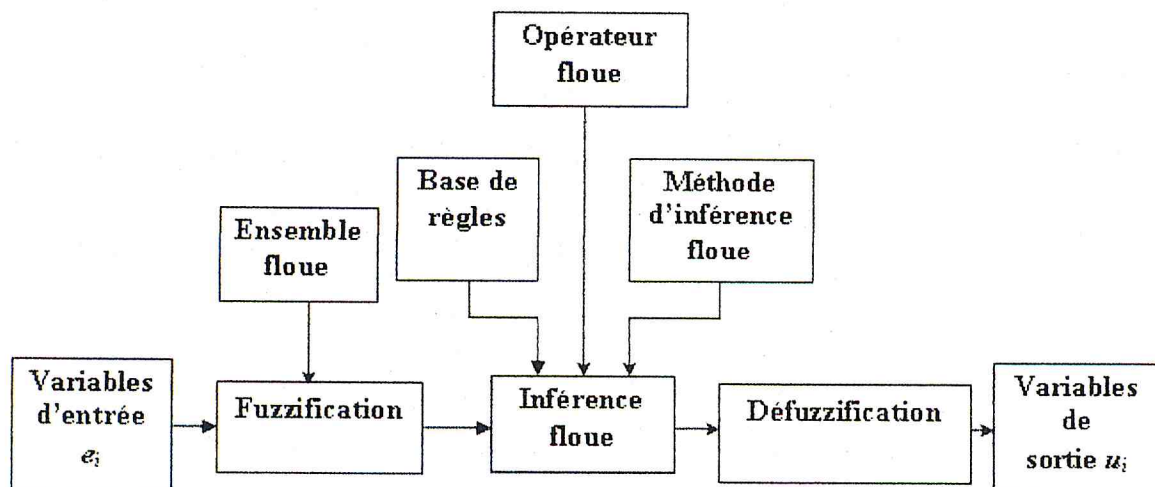
#### III.5.1 Généralité [12] [14]

On procède tout d'abord à la partition en sous-ensembles flous des différents univers de discours (ou référentiels) que le système impose. Ensuite on détermine la base de règles qui va caractériser le fonctionnement désiré du système.

Puis il faut transformer les variables réelles, c'est-à-dire celles qui ont une réalité physique, en variables floues. On appelle cette étape *la fuzzification* (Fuzzy = flou en anglais).

On utilise alors ces variables floues dans *un mécanisme d'inférence* qui crée et détermine les variables floues de sortie en utilisant les opérations sur les fonctions d'appartenance.

Enfin, on opère à *la défuzzification* qui consiste à extraire une valeur réelle de sortie à partir de la fonction d'appartenance du sous-ensemble flou de sortie établi par le mécanisme d'inférence.



#### III.5.2 Avantages et désavantages du réglage par logique floue [12] [14]

Les avantages principaux des régulateurs flous sont les suivants :

- Il n'est pas nécessaire de faire un modèle mathématique du système.
- On peut maîtriser les systèmes non linéaires et difficiles à modéliser.
- Grâce à des processeurs dédiés, on peut régler des processus rapides.

Les désavantages sont les suivants :

- Manque de directives précises pour la conception d'un régulateur.



- Approche artisanale et non systématique (il est souvent très difficile d'implanter les connaissances de l'opérateur).
- Impossibilité de démontrer la stabilité du circuit de réglage en toute généralité (en absence d'un modèle valable).
- précision du réglage en général peu élevée.
- Cohérence des inférences non garantie a priori (possible apparition de règles d'inférence contradictoires).

### III.5.3 Base de règles et définition [22]

Ensemble des définitions utilisées dans la commande floue (univers de discours, partitions floue, choix des opérateurs.....), ainsi que la base de règles « Si....Alors... » de la stratégie de commande de l'expert.

#### ➤ Partition floue

La partition floue d'un *univers de discours*  $U$  consiste à définir  $n$  sous-ensembles flous de façon à recouvrir  $U$ . l'ensemble de sous-ensembles flous à définir dans une partition d'un univers de variation est fixé par l'expert. Plus ce nombre est important et plus on définit de classes sur cet univers, ce qui permet d'augmenter la sensibilité de la commande floue.

#### ➤ Base de règles

Une fois la partition des univers de discours réalisée, il est possible de définir la base de règles. Par conséquent, si l'on considère  $n$  univers de discours  $U_i$  pour les prémisses des règles floues et si pour chaque univers  $U_i$  on définit une partition en  $m_i$  sous-ensemble flou, le nombre maximum de règles  $r_{\max}$  est :

$$r_{\max} = \prod_{i=1}^n m_i \quad (\text{III.7})$$

Le nombre de règles définis par l'expert peut être inférieur à  $r_{\max}$  c'est le cas particulier, s'il existe des sous-ensembles flous impossibles à obtenir pour le système. De plus, il est possible de considérer des configurations différentes aboutissant à la même conclusion.

Enfin, on peut remarquer qu'une augmentation de la sensibilité de la commande floue obtenue par une partition plus fine des univers de discours des prémisses aboutit à un accroissement important du nombre de règles et par conséquent complique leur formulation, elle augmente aussi le temps de traitement des *inférence* et de *la défuzzification*.

### III.5.4 Procédure de raisonnement flou

#### ➤ Fuzzification

La *fuzzification* est le traitement qui consiste à transformer les grandeurs physique (variables d'entrée) en variables linguistiques (variables floues), c'est-à-dire pondérer les fonctions d'appartenance des entrée aux divers sous-ensembles (attribution de degrés d'appartenance à chaque valeur d'entrée).

Il n'y a pas de règle précise pour la définition de fonctions d'appartenance. On peut introduire pour une variable  $x$ , trois, cinq ou sept valeur linguistiques. Ce nombre dépend de la résolution du réglage désirée.

#### ➤ Inférence floue

Le but de l'*inférence floue* (ou évaluation des règles) est de déterminer les sorties du contrôleur à partir des entrées floues issues de la fuzzification des entrées réelles et des règles de commande.

Les règles d'inférence, sont établies de manière générale comme suit :

$$\begin{array}{l}
 \text{Si } x \text{ est } A_1 \text{ et } y \text{ est } B_1 \text{ alors } z \text{ est } C_1. \\
 \text{Si } x \text{ est } A_2 \text{ et } y \text{ est } B_2 \text{ alors } z \text{ est } C_2. \\
 \vdots \\
 \text{Si } x \text{ est } A_n \text{ et } y \text{ est } B_n \text{ alors } z \text{ est } C_n.
 \end{array}
 \tag{III.8}$$

Où  $x$ ,  $y$  et  $z$  sont des variables linguistiques qui représentent les variables d'état du processus ( $x$ ,  $y$  : variable d'entrées), et  $z$  : variable de sortie.  $A_i$ ,  $B_i$  et  $C_i$  ( $i=1,n$ ) sont les sous-ensembles flous définis dans les ensembles de référence pour  $x$ ,  $y$  et  $z$  respectivement. En toute généralité, n'importe quelle combinaison des opérateurs **OU**, **ET** et **NON** peut apparaître dans la condition d'une règle, suivant les conditions imposées par le système à régler.

Supposons que l'on a défini neuf règles dans un système de réglage par logique floue. Ce système a deux entrées  $x$ ,  $y$  et une sortie  $z$ ; avec cinq ensembles flous pour chacune (négatif grand (NG), négatif petit (NP), environ zéro (EZ), positif petit (PP), positif grand (PG)).

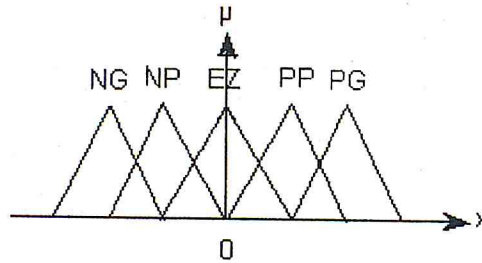


Figure III.11 : Représentations des cinq sous-ensembles flous.

Selon une certaine stratégie de réglage, les règles d'inférence floue, peuvent s'écrire sous *la forme linguistique* :

Si (x est négatif grand) ET (y est environ zéro) Alors (z égale positif grand) OU

Si (x est négatif petit) ET (y est environ zéro) Alors (z égale positif petit) OU

⋮

On peut obtenir une certaine simplification de l'écriture par *la description symbolique* :

Si (x est NG) ET (y est EZ) Alors (z est PG) OU

Si (x est NP) ET (y est EZ) Alors (z est PP) OU

⋮

Une simplification ultérieure s'obtient à l'aide d'une représentation graphique, appelée *matrice d'inférence* :

z	x				
	NG	NP	EZ	PP	PG
NG			PG		
NP			PP		
y EZ	PG	PP	EZ	NP	NG
PP			NP		
PG			NG		

Tableau III.2 : Matrice d'inférence.

Ainsi, il est plus simple d'utiliser *les tableaux d'inférence* :

Règle N	x <sub>1</sub>	x <sub>2</sub>	y
1	NG	EZ	PG
2	NP	EZ	PP
3	EZ	EZ	EZ
4	PP	EZ	NP
5	PG	EZ	NG
6	EZ	NG	PG
7	EZ	NP	PP
8	EZ	PP	NP
9	EZ	PG	NG

**Tableau III.3** : Tableau d'inférence.

Le traitement numérique des inférences, c'est-à-dire, des données fuzzifiées, se fait suivant l'une des méthodes d'inférence suivantes :

- Méthode d'inférence Max-Min.
- Méthode d'inférence Max-Prod.
- Méthode d'inférence Somme-Prod.

Parmi ces méthodes, la plus utilisée est la méthode d'inférence **Max-Min** (*implication de Mamdani* [20]), qui réalise au niveau de la condition, l'opérateur **OU** par la formation du maximum et l'opérateur **ET** par la formation du minimum. La conclusion dans chaque règle, introduite par **ALORS**. Elle se rapporte à la réalisation de **OU** liant les règles par (max) et la réalisation de **ALORS** par (min).

La figure III.12 représente graphiquement le principe de la méthode.

Pour chaque règle, on obtient la fonction d'appartenance partielle par la relation :

$$\mu_{r_i}(x_r) = \min[\mu_{c_i}, \mu_{o_i}(x_r)]; \quad i = 1, 2, \dots, m. \quad (\text{III.9})$$

Où  $\mu_{o_i}(x_r)$  est la fonction d'appartenance liée à l'opération imposée par la règle  $r_i$ . La fonction d'appartenance résultante est alors donnée par :

$$\mu_{res}(x_r) = \max[\mu_{r1}(x_r), \mu_{r2}(x_r), \dots, \mu_{rm}(x_r)]$$

(III.10)

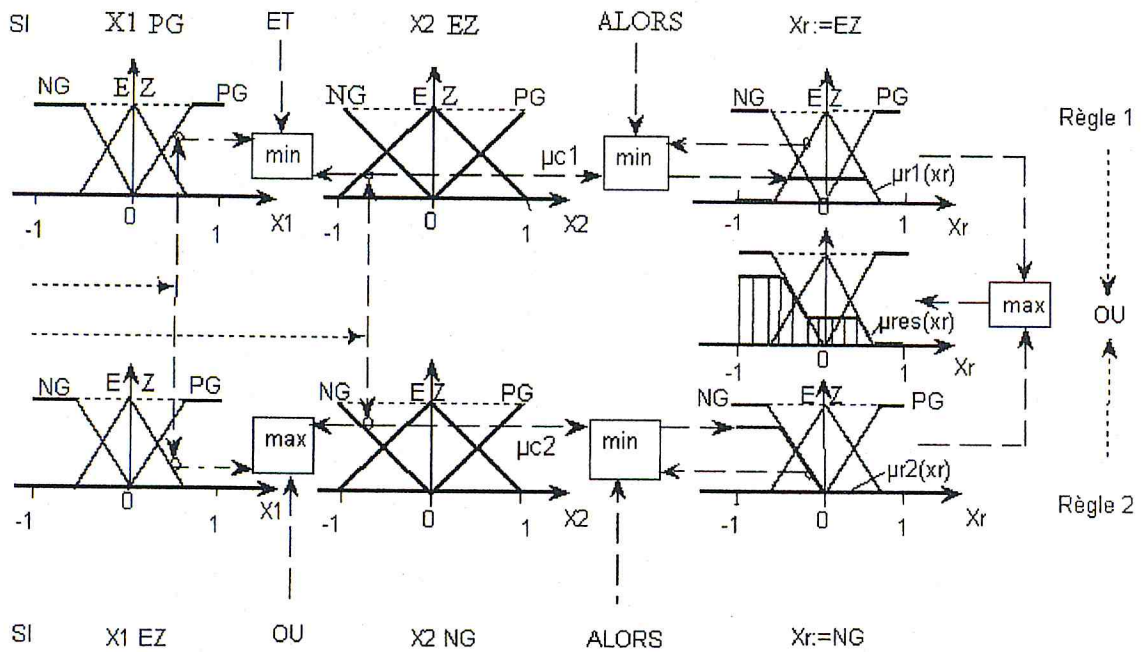


Figure III.12 : Méthode d'inférence Max-Min pour deux variables d'entrée et deux règles.

➤ **Défuzzification**

Le résultat d'une inférence floue est une fonction d'appartenance  $\mu_{res}(z)$ . C'est un sous-ensemble flou. La transformation d'une information floue en une information déterminée est la *défuzzification* (concrétisation), alors le but de ce dernier est d'extraire une valeur  $Z_0$  à partir de la fonction d'appartenance  $\mu_{res}(z)$ .

Il y a plusieurs méthodes de défuzzification :

La méthode la plus simple et la plus approximative est alors de prendre en compte *le premier maximum* de la fonction  $\mu_{res}(z)$ .

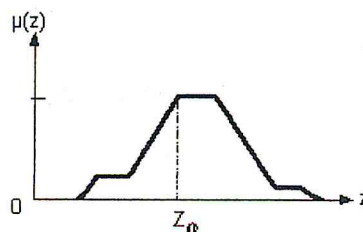


Figure III.13 : Défuzzification par le premier maximum.

La deuxième est de considérer la moyenne des maximums de la fonction  $\mu_{res}(z)$ .

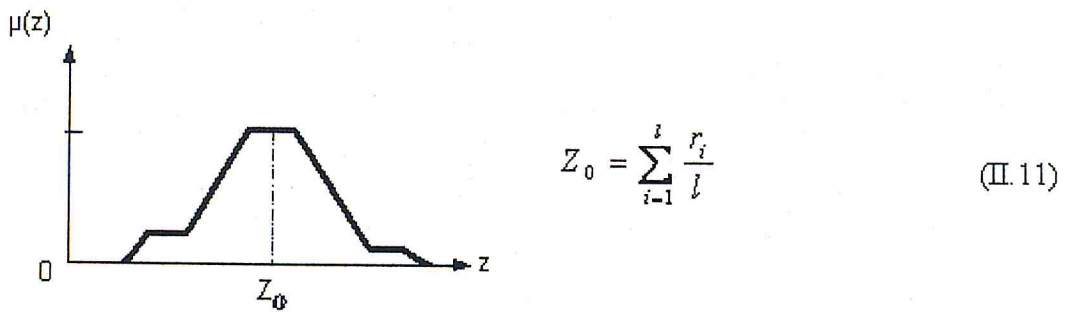


Figure III.14 : Défuzzification par la méthode de la moyenne des maximums.

Où  $l$  est le nombre de valeurs quantifiées  $r_i$  pour lesquelles l'appartenance est maximale.

La troisième méthode est l'extraction barycentrique de la valeur  $Z_0$ , qui consiste à faire une moyenne des valeurs du référentiel pondérées par leur degré d'appartenance. Cette méthode est encore peut coûteuse en calcul et donne de bons résultats.

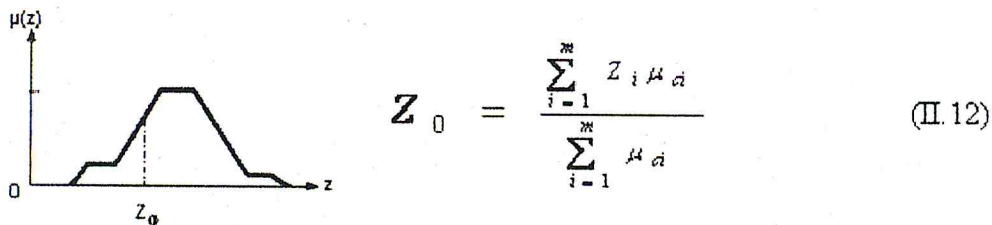


Figure III.15 : Défuzzification par la méthode de la moyenne pondérée.

La dernière méthode qui est de loin utile est la méthode du centre de gravité, elle consiste à déterminer l'abscisse  $Z_0$  du centre de gravité de la fonction d'appartenance résultant  $\mu_{res}(z)$ .

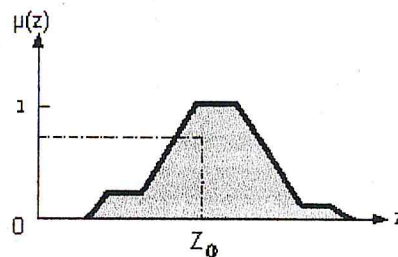


Figure III.16 : Défuzzification par le centre de gravité.

Pour le cas d'inférence max-min on obtient :

$$Z_0 = \frac{\sum_{i=1}^n Z_i \mu_{ci} \mu_{oi}(Z_i)}{\sum_{i=1}^n \mu_{ci} \mu_{oi}(Z_i)}$$

(III.13)

Où  $m$  est le nombre de règles contenues dans la fonction d'appartenance résultante  $\mu_{res}(z)$ .

### III.6 Conclusion

La logique floue reste, même si elle est déjà beaucoup utilisée, une technique nouvelle dont certains aspects sont encore inexploités. Elle est le sujet de nombreuses recherches et pourrait dans un futur proche devenir une technique incontournable pour l'amélioration de la communication Homme-machine.

L'utilisation d'un contrôleur flou pour une application donnée, nécessite une base de connaissances spécifiques (base de données et base des règles), ces informations sont généralement fournies par un expert. Or l'avis d'un être humain est souvent entaché d'incertitude.

Pour assurer une bonne base de connaissances, *les algorithmes génétiques* présentent une alternative puissante et intéressante de recherche combinatoire.

## IV.1 Introduction

La procédure d'un système flou nécessite deux types de réglages, appelées *réglages structurel* et *paramétrique*. Le réglage de la structure concerne le nombre de règles à utiliser dans le système ou encore le nombre de sous-ensembles flous à utiliser pour le codage d'une variable linguistique. Une fois le réglage structurel terminé, on peut ajuster de façon automatique les paramètres des variables d'entrée et de sortie. Plusieurs méthodes de réglage paramétrique ont été développées, parmi ces méthodes *les algorithmes génétiques*.

Les algorithmes des colonies de fourmis permettent, pour le design des contrôleurs flous, de déterminer le nombre et la position optimale des fonctions d'appartenance, le choix optimal des règles floues, la structure des sous-ensembles flous est définie sans a priori.

## IV.2 Principe d'optimisation

Un contrôleur flou repose sur la structure représentée à la figure IV.1 qui inclut :

- ✓ La fuzzification des variables d'entrée, avec éventuellement une normalisation ;
- ✓ L'inférence à partir d'une base de règles ;
- ✓ La défuzzification, avec éventuellement une dénormalisation.

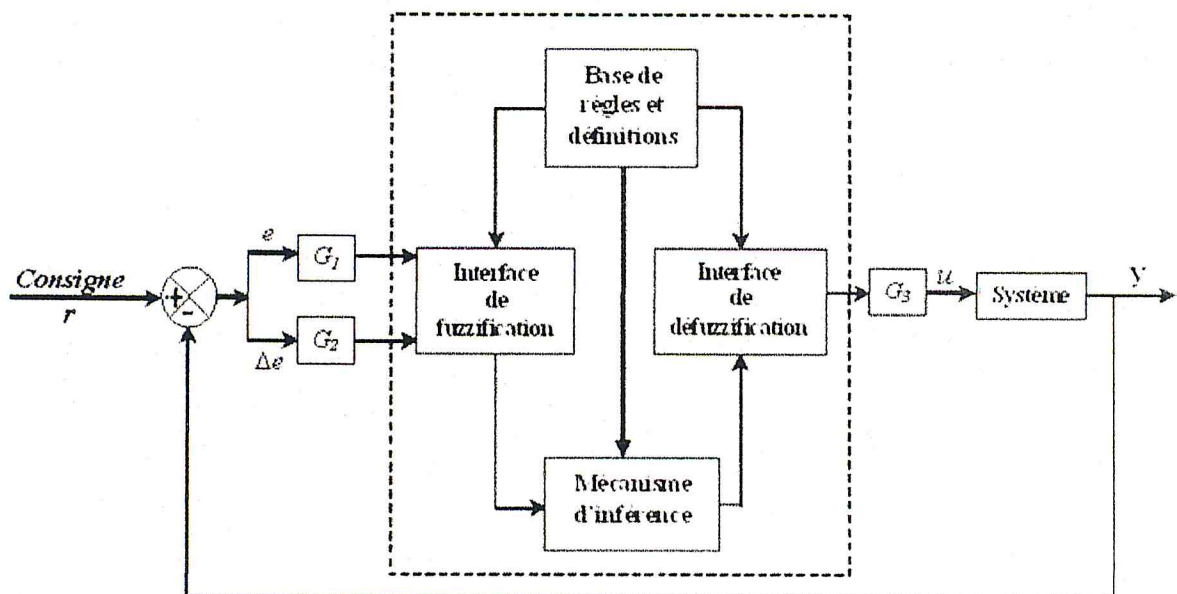


Figure IV.1 : Structure d'une commande floue.

Le contrôleur reçoit en entrée les variables suivantes :

- L'erreur  $e$  entre une valeur mesurée  $y$  et une valeur de consigne  $r$  ( $e=y-r$ )
- La variation de l'erreur  $\Delta e$ .

Le système flou permet de générer en sortie le signal de commande  $u$ .

Notre objectif est l'utilisation des colonies de fourmis pour résoudre les problèmes de la logique floue. Cette technique d'apprentissages permet :

- L'ajustement des parties conclusions des règles d'inférence tout en laissant fixe les parties conditions.
  - ✓ Fixer les centres des fonctions d'appartenance et ajuster les largeurs.
  - ✓ Fixer les largeurs des fonctions d'appartenance et ajuster les centres
  - ✓ Ajuster à la fois les centres et les largeurs des fonctions d'appartenance.



### IV.3 PROBLEMATIQUE DU SUJET DE L'ADAPTATION DES COLONIES DE FOURMIS A LA COMMANDE D'ATTITUDE

Pour la suite de l'étude, quelques définitions relatives à la théorie des graphes sont nécessaires.

- Un graphe non orienté  $G$  est un couple  $(S, A)$  où  $S$  est un ensemble fini des sommets et  $A$  l'ensemble des paires constituées de sommets (les arêtes). Pour orienter le graphe, on remplace  $A$  par une relation binaire sur  $S$ .
- Une chaîne d'un graphe non orienté  $G$  est une séquence de sommets de  $G$
- Un graphe non orienté est connexe si chaque paire de sommet est reliée par une chaîne.
- Lorsque le graphe est orienté, il est dit fortement connexe si chaque sommet est accessible à partir d'un autre.
- Une chaîne  $\langle v_i \rangle_{1 \leq i \leq n}$  est un cycle si et seulement si ( $n \geq 3$ ,  $v_0 = v_n$  et  $\forall i \neq j$ ,  $v_i \neq v_j$ ).
- Un graphe non orienté acyclique est appelée forêt, un graphe non orienté connexe acyclique est appelé un arbre.
- Un cycle hamiltonien d'un graphe non orienté  $G$  est un cycle où chaque sommet de  $G$  n'est visité qu'une et une seule fois.

Il en ressort que le problème du voyageur de commerce est formellement la recherche du cycle hamiltonien minimisant la pondération du graphe, lorsque on associe à chaque sommet du graphe étudié une fonction de pondération. La méthode de la colonie de fourmis simule le comportement de ces insectes qui, lorsque l'on pose un obstacle sur leur trajet, trouvent toujours le chemin le plus court pour le contourner. Leur technique repose sur la pose de marqueurs chimiques, les phéromones, déposés sur les trajets parcourus.

Cela peut paraître surprenant au premier abord mais un chemin plus court reçoit plus de phéromones qu'un chemin plus long. Les fourmis suivantes ayant tendance à emprunter les chemins les plus marqués, une solution va progressivement se dessiner. S'inspirant du problème du TPS, un ensemble de villes (données) choisi l'ensemble des segments est initialisé par une quantité de phéromones représentant la valeur d'un équivalent énergétique caractérisant notre système satellitaire.

Pour la suite, à chaque essai, des fourmis sont lancées sur le parcours de notre banque de données en inscrivant des "odeurs" leur permettant de modifier en fonction des trajets parcourus, des décisions de commandes. Les réglages opérés au niveau des contrôleurs, permettront d'approcher les consignes souhaitées.

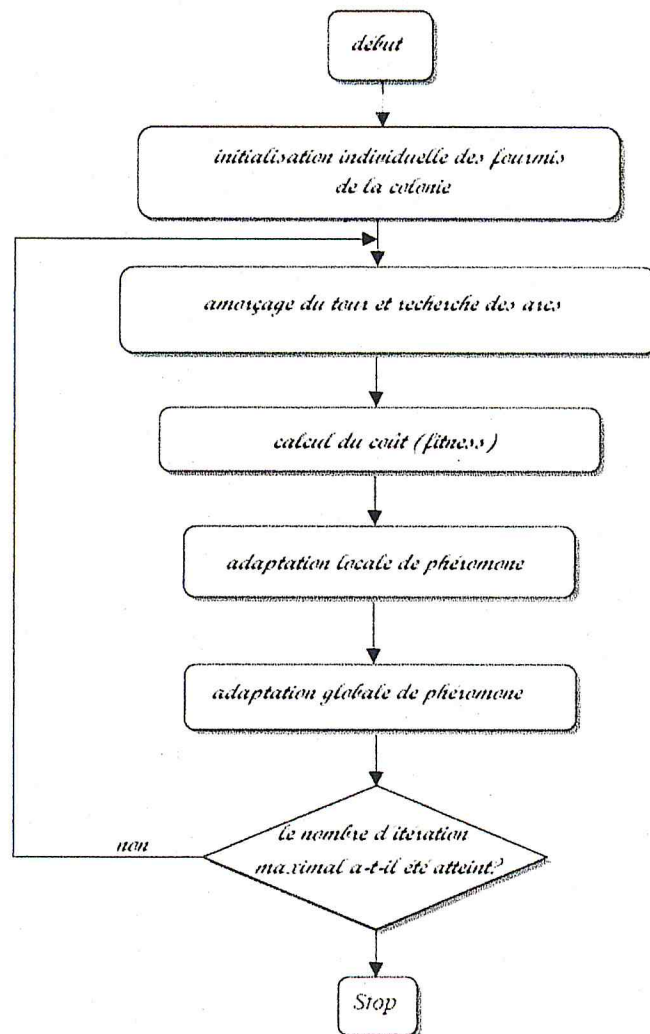


figure 4.2: Organigramme des colonies de fourmie représentant la méthodologie d'optimisation des paramètres d'un contrôleur flou

## Chromosome associé à la structure des données

Le chromosome d'un ensemble flou peut être subdivisé en deux types de gènes :

- Les gènes de contrôle : qui déterminent les fonctions d'appartenances actives.
- Les gènes paramétriques (nombres réels) : qui représentent les paramètres des fonctions d'appartenance.

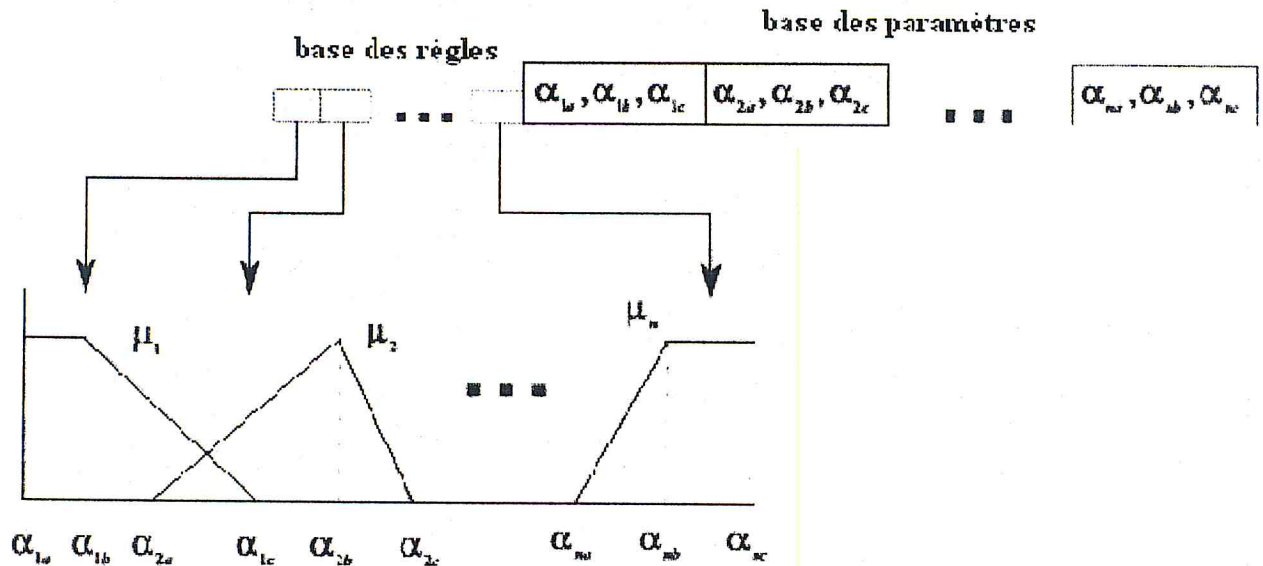


Figure IV.3 : Structure associée au chromosome d'un ensemble flou.

La structure du chromosome qui représente les fonctions d'appartenance du contrôleur flou est la suivante :

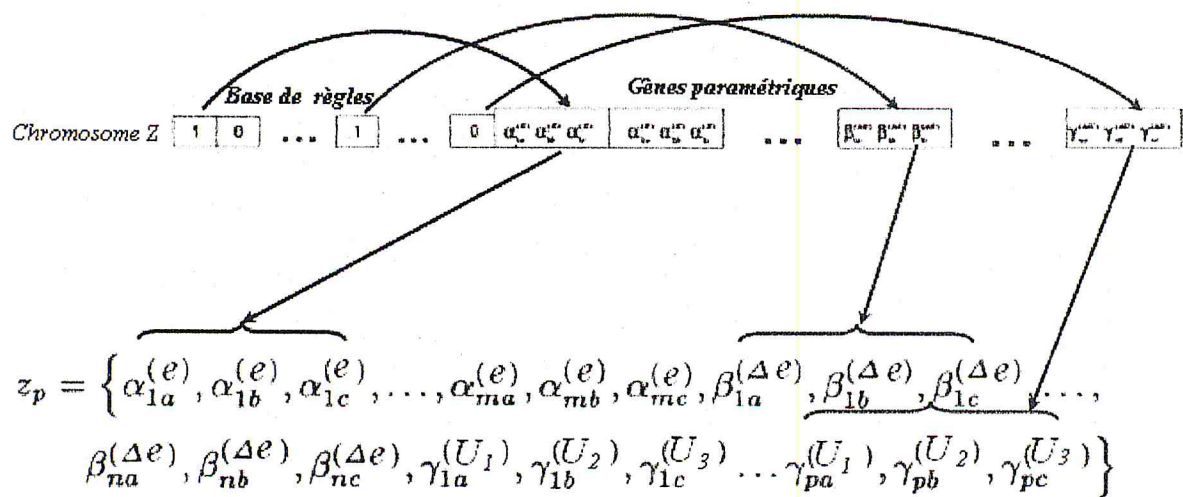


Figure IV.4 : Structure des fonctions d'appartenance du contrôleur flou.

Tel que :

- m, n et p : le nombre maximum de sous-ensemble flou de e, Δe et u respectivement.
- Les paramètres du i<sup>ième</sup> sous-ensemble flou d'entrée de e.
- Les paramètres du j<sup>ième</sup> sous-ensemble flou de variation d'entrée Δe.
- Les paramètres du k<sup>ième</sup> sous-ensemble flou de sortie u.

### IV.3 Les facteurs d'échelle (gain de normalisation) [2]

Lors de la fuzzification il est plus facile de définir les fonctions d'appartenance des qualificatifs une fois pour toutes, sur un seul intervalle fermé symétrique par rapport à zéro  $\Delta \subset R$ .

Dans ce contexte, les variables d'entrées ( $e, \Delta e$ ) du régulateur sont multipliées par deux coefficients  $G_1, G_2$  appelés facteurs d'échelle, tel que :

$$e_1 = G_1(y-r) \quad , \quad e_2 = G_2 \Delta e$$

#### IV.4 Optimisation sous contraintes [18]

L'optimisation avec contraintes a pour but essentiel d'assurer la lisibilité et la sémantique des règles. Les méfaits d'une absence de contraintes sur les fonctions d'appartenance sont illustrés en figure IV.4, qui recense à peu près tous les désordres que l'on peut rencontrer.

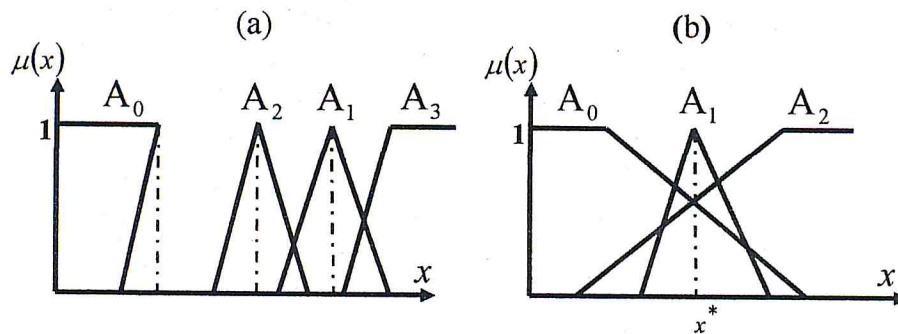


Figure IV.5 : Les méfaits d'une absence de contraintes sur les fonctions d'appartenance.

Les problèmes rencontrés sont les suivants :

**Recouvrement nul (lacune)** : comme dans figure IV.1.a, ce qui peut arriver si les points d'apprentissage ne sont pas denses dans l'espace d'entrée. La contrainte  $C_1$  assure le recouvrement.

$$C_1 : \forall x \in U, \exists j \text{ tel que } \mu_{A_j}(x) \geq \delta > 0$$

Où les  $\mu_{A_j}(x)$  sont des sous-ensembles flous définis sur le domaine  $U$  de  $x$ , et  $\delta$  un degré de recouvrement donné.

**Incohérence** : Le trapèze  $A_0$  de la figure IV.4.a ne correspond plus à un label clairement identifié ni à une fonction d'appartenance (ici l'image d'un point  $x$  n'est pas toujours unique).

**Perte de sémantique** : Les triangles  $A_1$  et  $A_2$  de là figure IV.4.a ont été inversés et la relation d'ordre "linguistique" entre les labels correspondants n'est plus vérifiée. La contrainte  $C_2$  assure qu'il n'y aura pas d'inversion de labels (c'est une relation d'ordre sémantique).

$$C_2 : \forall i < j, \max_{a_i \in N(A_i)} a_i < \min_{a_j \in N(A_j)} a_j$$

**Chevauchement "Brouhaha"** : Illustré en figure IV.4.b au point  $x^*$  tous les degrés d'appartenance ont une valeur élevée et le fait que "x est  $A_1$ " ne peut pas être examiné séparément.

Comme notre problème ne fonctionne pas comme une boîte noire et les connaissances acquises doivent être interprétées, alors il est nécessaire d'introduire les contraintes.

#### IV.5 Description du système

Afin de réaliser la commande d'un tel système, une étape de modélisation est nécessaire pour permettre l'étude en simulation.

Le modèle non linéaire du système est donné par les équations différentielles suivantes :

$$\left\{ \begin{array}{l} \ddot{\phi} = \omega_0 \dot{\psi} - \frac{(I_L - I_T)(-\omega_0 + \dot{\theta})(\dot{\psi} + \omega_0 \phi)}{I_R} + \frac{3\omega_0^2(I_L - I_T)\cos^2 \theta \cos \phi \sin \phi}{I_R} + \frac{M_{px}}{I_R} \\ \ddot{\theta} = -\frac{(I_R - I_L)(\dot{\phi} - \omega_0 \psi)(\dot{\psi} + \omega_0 \phi)}{I_T} - \frac{3\omega_0^2(I_R - I_L)\cos \theta \cos \phi \sin \theta}{I_T} + \frac{M_{py}}{I_T} \\ \ddot{\psi} = -\omega_0 \dot{\phi} - \frac{(I_T - I_R)(-\omega_0 \psi + \dot{\phi})(-\omega_0 + \dot{\theta})}{I_L} - \frac{3\omega_0^2(I_T - I_R)\cos \theta \sin \phi \sin \theta}{I_L} + \frac{M_{pz}}{I_L} \end{array} \right.$$

L'expression de ce modèle doit être sous forme d'une équation différentielle ordinaire :

$\dot{x} = f(x(t), u(t), p, t)$  comme le montrent les expressions suivantes :

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = \omega_0 x_6 - \frac{(I_L - I_T)(-\omega_0 + x_4)(x_6 + \omega_0 x_1)}{I_R} + \frac{3\omega_0^2(I_L - I_T)\cos^2 x_3 \cos x_1 \sin x_1}{I_R} + \frac{M_{px}}{I_R} \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = -\frac{(I_R - I_L)(x_2 - \omega_0 x_5)(x_6 + \omega_0 x_1)}{I_T} - \frac{3\omega_0^2(I_R - I_L)\cos x_3 \cos x_1 \sin x_3}{I_T} + \frac{M_{py}}{I_T} \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = -\omega_0 x_2 - \frac{(I_T - I_R)(-\omega_0 x_5 + x_2)(-\omega_0 + x_4)}{I_L} - \frac{3\omega_0^2(I_T - I_R)\cos x_3 \sin x_1 \sin x_3}{I_L} + \frac{M_{pz}}{I_L} \end{array} \right.$$

Les vecteur d'état, de commande et de perturbation sont donnés par :

$$x = \begin{pmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{pmatrix}; \quad u = \begin{pmatrix} M_{cx} \\ M_{cy} \\ M_{cz} \end{pmatrix}; \quad p = \begin{pmatrix} M_{px} \\ M_{py} \\ M_{pz} \end{pmatrix}$$

Le vecteur de sortie est le suivant :

$$y = \begin{pmatrix} \varphi \\ \theta \\ \psi \end{pmatrix} = \begin{pmatrix} x_1 \\ x_3 \\ x_5 \end{pmatrix}$$

Le trajectoire de référence est donnée par :  $y_r = (y_{r1} \quad y_{r2} \quad y_{r3})^T$

### IV.5.1 Étude du mouvement libre du satellite

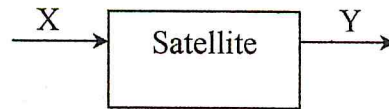


Figure IV.6 : Représentation du mouvement libre du satellite.

Les équations qui caractérisent le mouvement libre du satellite soumis uniquement au couple de gradient de gravité sans commande sont les suivantes :

$$\begin{cases} I_R \ddot{\varphi} - I_R \omega_0 \dot{\psi} + (I_L - I_T)(-\omega_0 + \dot{\theta})(\dot{\psi} + \omega_0 \varphi) - 3\omega_0^2 (I_L - I_T) \cos^2 \theta \cos \varphi \sin \varphi = 0 \\ I_T \ddot{\theta} + (I_R - I_L)(\dot{\varphi} - \omega_0 \psi)(\dot{\psi} + \omega_0 \varphi) + 3\omega_0^2 (I_R - I_L) \cos \theta \cos \varphi \sin \theta = 0 \\ I_L \ddot{\psi} + I_L \omega_0 \dot{\varphi} + (I_T - I_R)(-\omega_0 \psi + \dot{\varphi})(-\omega_0 + \dot{\theta}) + 3\omega_0^2 (I_T - I_R) \cos \theta \sin \varphi \sin \theta = 0 \end{cases}$$

Les résultats de simulation obtenus par la méthode de Runge-Kutta d'ordre 4 avec un pas de simulation 0.1.

### IV.5.2 Étude du système en boucle fermée

L'objectif de notre application, est d'établir le contrôle flou des paramètres décrivant l'orientation du système d'axes liée au satellite par rapport au repère orbital local par les angles de : roulis, tangage et lacet.

Pour assurer le pointage du satellite vers les directions requises, nous avons utilisé un système d'asservissement, dont le schéma fonctionnel est donné par la figure suivante:

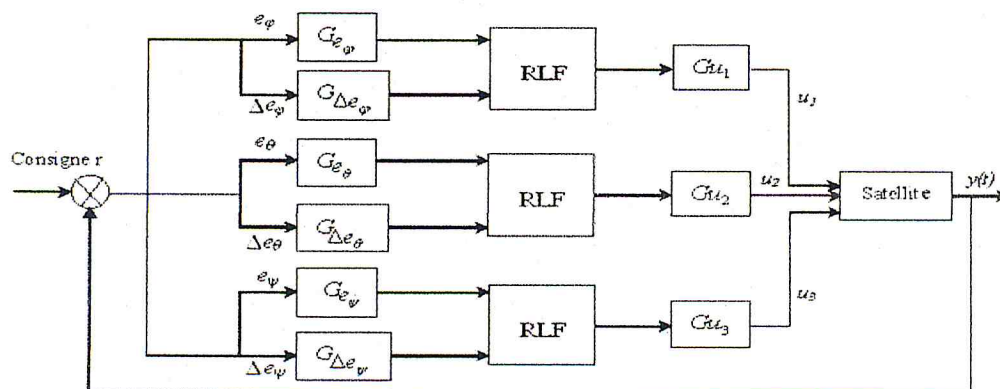


Figure IV.7 : Schéma fonctionnel d'un système asservi.

### Caractéristiques du contrôleur flou

- Nous avons 3 angles à régler pour cela on détermine 3 régulateurs flou de type MISO (plusieurs entrées, une seule sortie) : pour chaque régulateur nous avons utilisé l'implication de Mamdani (méthode max-min), et qui a pour méthode de défuzzification la méthode le centre de gravité.
- Les variables d'entrées de chaque régulateur sont :  
 Régulateur (1) : l'erreur  $e_\varphi$  et sa variation  $\Delta e_\varphi$  ;  
 Régulateur (2) : l'erreur  $e_\theta$  et sa variation  $\Delta e_\theta$  ;  
 Régulateur (3) : l'erreur  $e_\psi$  et sa variation  $\Delta e_\psi$   
 Les variables de la sortie sont :  $u_1, u_2$  et  $u_3$ .
- Chacun de ces régulateurs est représenté de la même répartition des espaces d'entrées et de sortie.
- Les règles de contrôle se composent de paires situation/action de la forme :

$$\text{Si } \underbrace{e \text{ est } \{...\} \text{ et } \Delta e \text{ est } \{...\}}_{\text{Situation}} \text{ Alors } \underbrace{u \text{ est } \{...\}}_{\text{Action}}$$

La définition de la base de données (univers de discours + ouvertures de la fonction d'appartenance + les gains de la normalisation) et l'écriture des règles de contrôle, peut se faire par le biais d'un expert.

Néanmoins, comme souligné auparavant n'existerait-il pas un second expert qui puisse nous donner une base de connaissance qui influencerait positivement sur le rendement du contrôleur flou.

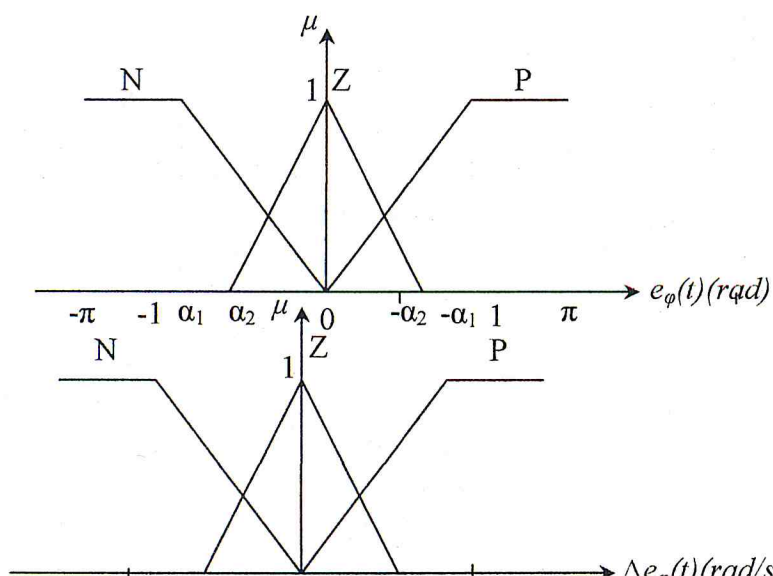
D'où l'idée d'utiliser les algorithmes des colonies de fourmis comme puissante méthode de recherche combinatoire.

### IV.7 Application

#### 1) Premier cas

L'utilisation d'un RLF avec trois fonctions d'appartenance pour toutes les variables E/S de chaque angle.

Dans notre problème pour fuzzifier les variables d'entrées et sorties, on utilise des fonction d'appartenance de type trapézoïdal et triangulaire, comme montre la figure ci-dessous :



$$\text{Trapèze (N)} = [-\pi \ -\pi \ \alpha_1 \ 0]$$

$$\text{Triangle (Z)} = [\alpha_2 \ 0 \ -\alpha_2]$$

$$\text{Trapèze (P)} = [0 \ \alpha_1 \ \pi \ \pi]$$

$$\text{Trapèze (N)} = [-\pi \ -\pi \ \beta_1 \ 0]$$

$$\text{Triangle (Z)} = [\beta_2 \ 0 \ -\beta_2]$$

$$\text{Trapèze (N)} = [0 \ \beta_1 \ \pi \ \pi]$$

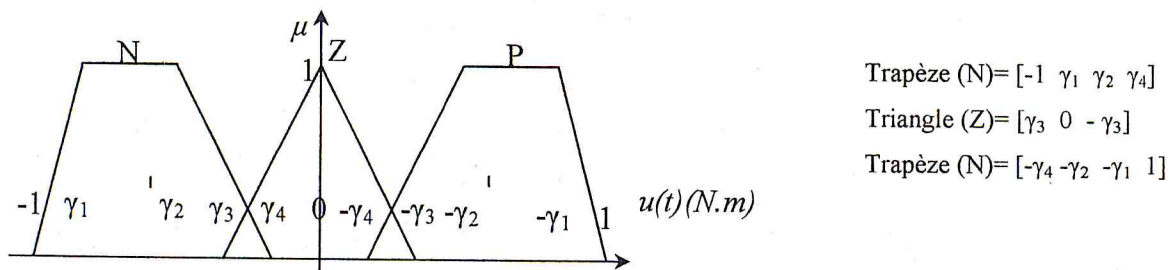


Figure IV.8 : Répartition des espaces d'entrées et de sortie.

**N.B** : les fonctions d'appartenance sont symétriques.

N : négative ; P : positive ; Z : zéro.

La table de décision est représentée par :

		$\Delta e(t)$		
		<b>N</b>	<b>Z</b>	<b>P</b>
$e(t)$	<b>N</b>	P	P	Z
	<b>Z</b>	P	Z	N
	<b>P</b>	Z	N	N

Tableau IV.1 : La table de décision pour trois fonctions d'appartenance.

Le mécanisme d'inférence utilisé par la méthode Max-min, pour la phase de défuzzification nous avons utilisé la méthode centre de gravité qui est donnée par l'expression générale suivante :

$$x_R^* = \frac{\int x_R \mu_{res}(x_R) dx_R}{\int \mu_{res}(x_R) dx_R}$$

La réalisation d'un contrôleur flou par les algorithmes des colonies de fourmis, est basée sur le choix réfléchi de la base des données.

Le codage du problème pour trois fonctions d'appartenance est réalisé à partir d'une chaîne chromosomique de 17 gènes (son équivalent géométrique aurait été associé à un vecteur de 17 composantes).

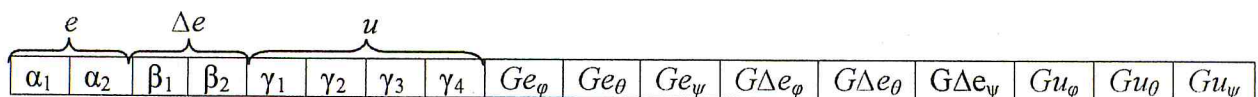


Figure IV.9 : Structure chromosomique pour trois fonctions d'appartenance.



Les huit premiers gènes représentent les projections des fonctions d'appartenance ( $e, \Delta e, u$ ) sur l'axe de l'univers du discours normalisé. Pour éviter les méfaits d'une absence de contraintes sur les fonctions d'appartenances, nous imposons les conditions suivantes :

$$\begin{aligned}
 & -1 < \alpha_1 < \alpha_2 < 0 \dots\dots\dots (1) \\
 & -1 < \beta_1 < \beta_2 < 0 \dots\dots\dots (2) \\
 & -1 < \gamma_1 < \gamma_2 < \gamma_3 < \gamma_4 \dots\dots\dots (3)
 \end{aligned}$$

Tandis que les neufs derniers gènes, elles représentent les gains; tel que :

	Cas du mini satellite			Cas du micro satellite		
	$\varphi$	$\theta$	$\psi$	$\varphi$	$\theta$	$\psi$
$Ge$	[0 1/ $\pi$ ]	[0 1/ $\pi$ ]	[0 1/ $\pi$ ]	[0.1 1/ $\pi$ ]	[0.1 1/ $\pi$ ]	[0.1 1/ $\pi$ ]
$G\Delta e$	[3 12]	[3 6]	[8 12]	[20 30]	[20 30]	[20 30]
$Gu$	[15 30]	[15 60]	[2 3]	1	1	1

**Tableau IV.2 :** Domaines d'appartenance des gains de normalisations de  $e$  et  $\Delta e$  et dénormalisation du  $u$  pour trois fonctions d'appartenance.

L'algorithme qui permet de déterminer cette base de donnée optimale à partir de ce type de codage s'appuie sur le problème posé au voyageur de commerce (TSP) qui consiste, étant donné un ensemble de villes séparées par des distances données, à trouver le plus court chemin qui relie toutes les villes. Il s'agit d'un problème d'optimisation pour lequel doit être conçu un algorithme permettant de trouver une solution exacte en un temps minimal .

De plus, la version décisionnelle de l'énoncé (*pour une distance  $D$ , existe-t-il un chemin plus court que  $D$  passant par toutes les villes ?*) est connu comme étant un problème NP-complet. Pour optimiser un parcours permettant d'utiliser un maximum de données est décrit par les étapes suivantes sont nécessaires :

❖ Modélisation nodale du problème

Cette modélisation passe par une construction analogue au problème du voyageur de commerce. La figure 4.10 représente ladite modélisation.

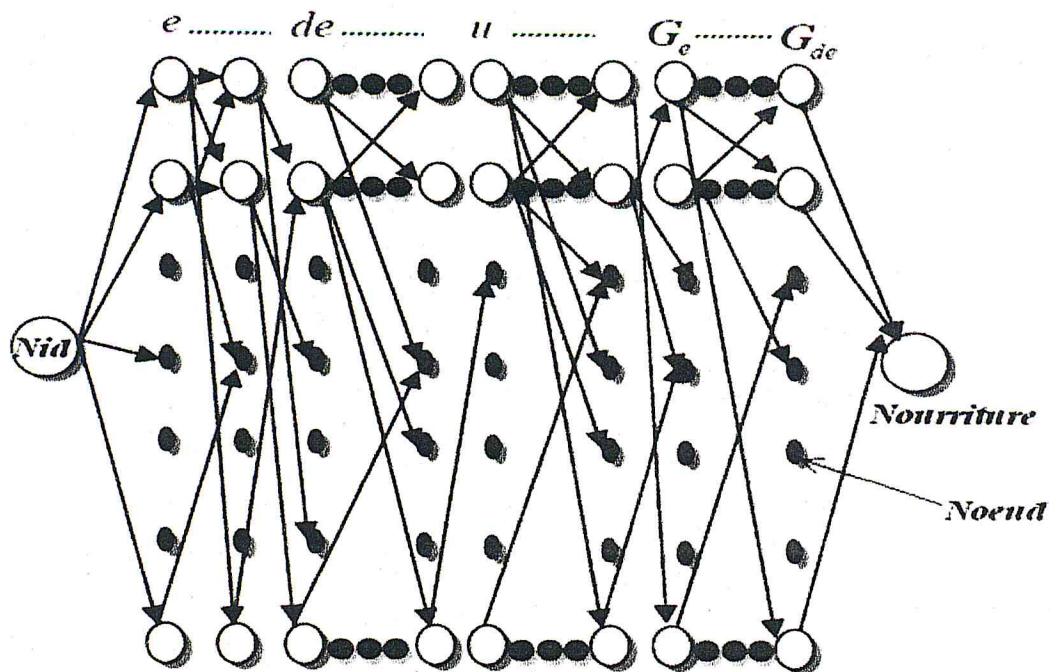


Figure 4.10 : Représentation nodale des possibilités de parcours

❖ séquence initiale de données

On doit générer une population initiale, pour cela on fixe la taille de la population (nombre d'individus) ; la longueur du chromosome.

❖ Évaluation

Chaque individu est évaluée sur la base du temps nécessaire pour que l'erreur entre les angles et ses consignes converge vers zéro, d'où :

$$e = y - y_r \quad ; \quad e = \begin{pmatrix} e_\varphi \\ e_\theta \\ e_\psi \end{pmatrix} \quad ; \quad y = \begin{pmatrix} \varphi \\ \theta \\ \psi \end{pmatrix} \quad ; \quad y_r = \begin{pmatrix} \varphi_r \\ \theta_r \\ \psi_r \end{pmatrix}$$

❖ Evaluation de l'intensité des phéromones

Soient  $C_{ij}$  un gène sélectionné ( $i$  : individus,  $j$  : le bit),  $\theta_m \ll 1$  un nombre réel :

- 1/ choisir aléatoirement un nombre  $s \in \{-1 \ 1\}$
- 2/ choisir aléatoirement  $\lambda \in [0 \ \theta_m]$
- 3/  $C_{ij} \leftarrow (1 - \lambda)C_{ij} + \lambda C_{i,j+s}$

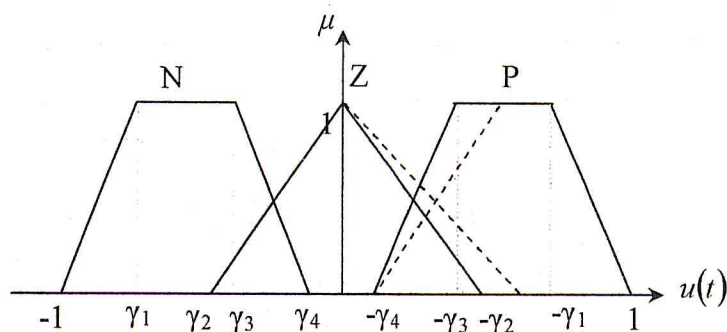


Figure IV.11 : Illustration schématique de la mutation.

❖ **Fonction d'adaptation**

$$fitness = \frac{1}{2} \sum_{i=1}^{\infty} (e_{\varphi}^2 + e_{\theta}^2 + e_{\psi}^2 + u_1^2(t) + u_2^2(t) + u_3^2(t))$$

**2) Deuxième cas**

L'utilisation d'un RLF avec cinq fonctions d'appartenance pour toutes les variables E/S de chaque angle.

On utilise des fonctions d'appartenance de type trapézoïdal et triangulaire, comme montre la figure ci-dessous :

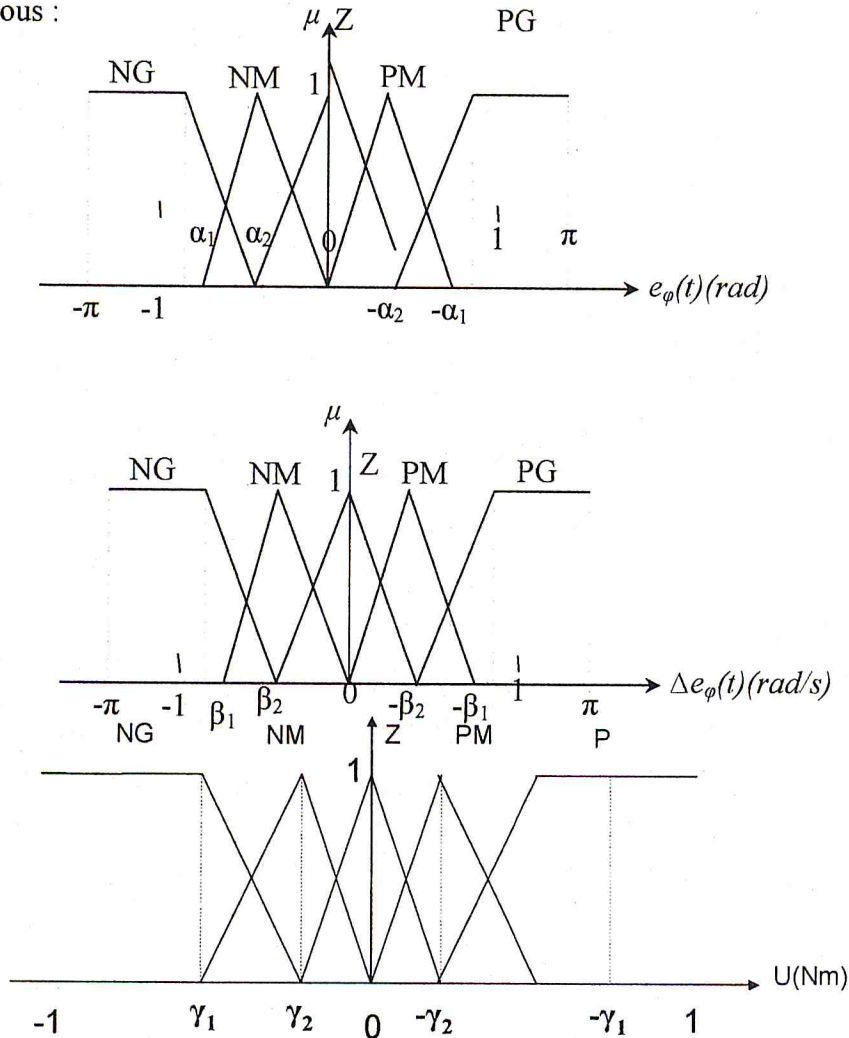


Figure IV.11 : Répartition des espaces d'entrées et de sortie pour cinq fonctions d'appartenance.

**N.B :** les fonctions d'appartenance sont symétriques.

NG : négative grand ; PG : positive grand ; NM : négative moyen ; PM : positive moyen.

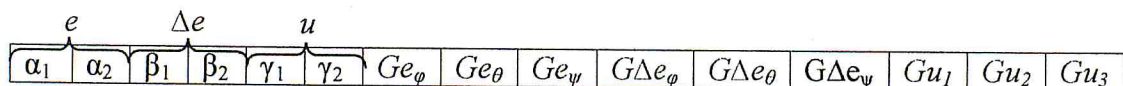
On représente la base de règles relative à cette application par la table de décision suivante :

	$\Delta e(t)$				
$u(t)$	NG	NM	Z	PM	PG

$e(t)$	<b>NG</b>	PG	PG	PM	PM	Z
	<b>NM</b>	PG	PM	PM	Z	NM
	<b>Z</b>	PM	PM	Z	NM	NM
	<b>PM</b>	PM	Z	NM	NM	NG
	<b>PG</b>	Z	NM	NM	NG	NG

**Tableau IV.3 :** La table de décision pour cinq fonctions d'appartenance.

Le codage du problème pour cinq fonctions d'appartenance est réalisé par un chromosome de 15 bits.



**Figure IV.12 :** Structure de chromosome pour cinq fonctions d'appartenance.

Les six premiers gènes représentent les projections des fonctions d'appartenance ( $e$ ,  $\Delta e$ ,  $u$ ) sur l'axe de l'univers du discours normalisé. Pour éviter les méfaits d'un absence de contraintes sur les fonctions d'appartenance, nous imposons les contraintes suivantes :

$$\begin{aligned}
 -1 < \alpha_1 < \alpha_2 < 0 & \dots \dots \dots (1) \\
 -1 < \beta_1 < \beta_2 < 0 & \dots \dots \dots (2) \\
 -1 < \gamma_1 < \gamma_2 < 0 & \dots \dots \dots (3)
 \end{aligned}$$

Tandis que les neufs derniers gènes, elles représentent les gains tel que :

	Cas du mini satellite			Cas du micro satellite		
	$\varphi$	$\theta$	$\psi$	$\varphi$	$\theta$	$\psi$
$Ge$	[0.02 0.4]	[0.02 2]	[0.1 2]	[0.1 1/π]	[0.1 1/π]	[0.1 1/π]
$G\Delta e$	[2 10]	[2 9]	[3 4]	[20 30]	[20 30]	[20 30]
$Gu$	[200 500]	[60 100]	[5 10]	2	2	2

**Tableau IV.4:** Les intervalles des gains de normalisations de  $e$  et  $\Delta e$  et dénormalisation du  $u$  pour cinq fonctions d'appartenance.

On utilise le même algorithme qui a été appliquée pour trois fonctions d'appartenance.

### IV.8 Courbes et interprétations

- Les paramètres de simulations pour **Mini satellite** sont :  $I_R=132 \text{ Kg.m}^2$ ;  $I_T=137 \text{ Kg.m}^2$ ;  $I_L=7 \text{ Kg.m}^2$ ; orbite circulaire; masse de satellite d'environ  $m_S=150 \text{ Kg}$ .

- Les paramètres de simulations pour **Micro satellite UoSAT-1 2** sont :  $I_R=40.45 \text{ Kg.m}^2$ ;  $I_T=42.09 \text{ Kg.m}^2$ ;  $I_L=40.36 \text{ Kg.m}^2$ ; orbite circulaire: hauteur=50Km; inclinaison=64.6°; masse de satellite d'environ  $m_S=\text{Kg}$ .

➤ **Mouvement libre**

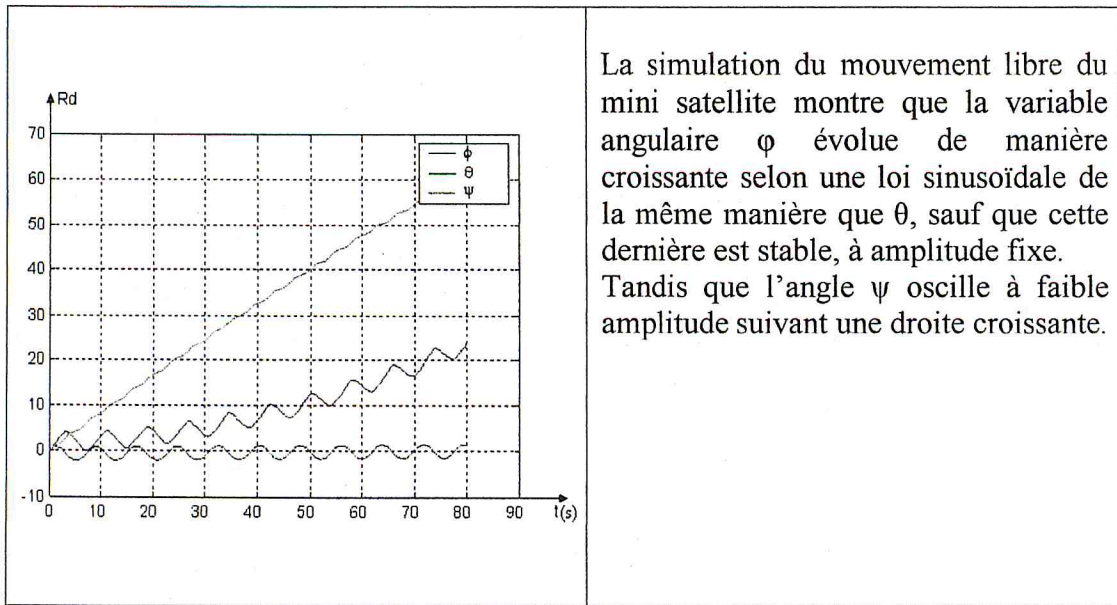


Figure IV.14 : Cas du mini satellite

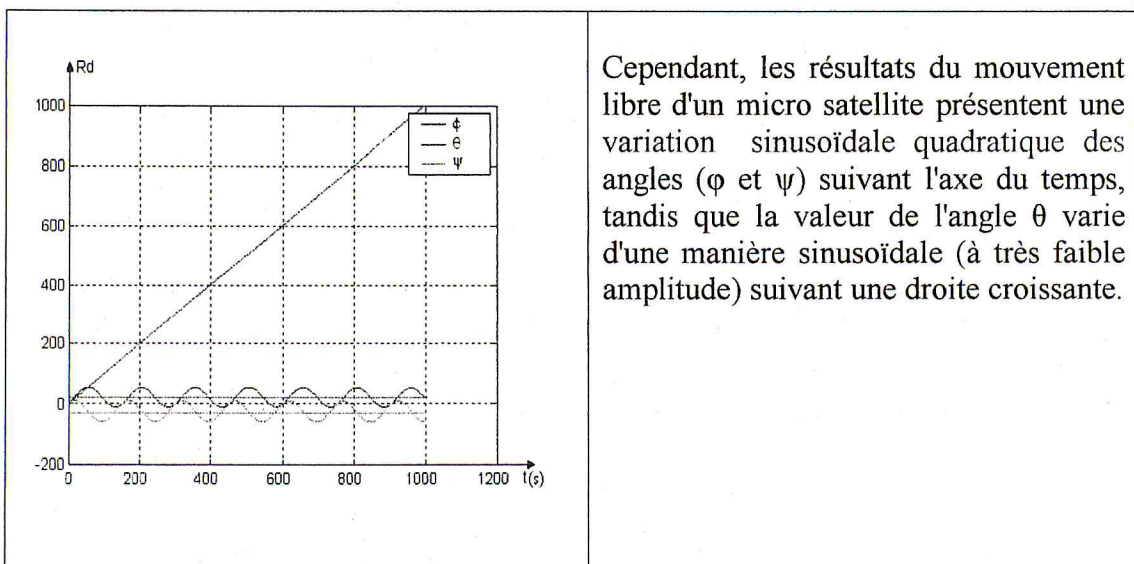


Figure IV.15 : Cas du micro satellite

### ➤ Utilisation du contrôleur flou Mini satellite

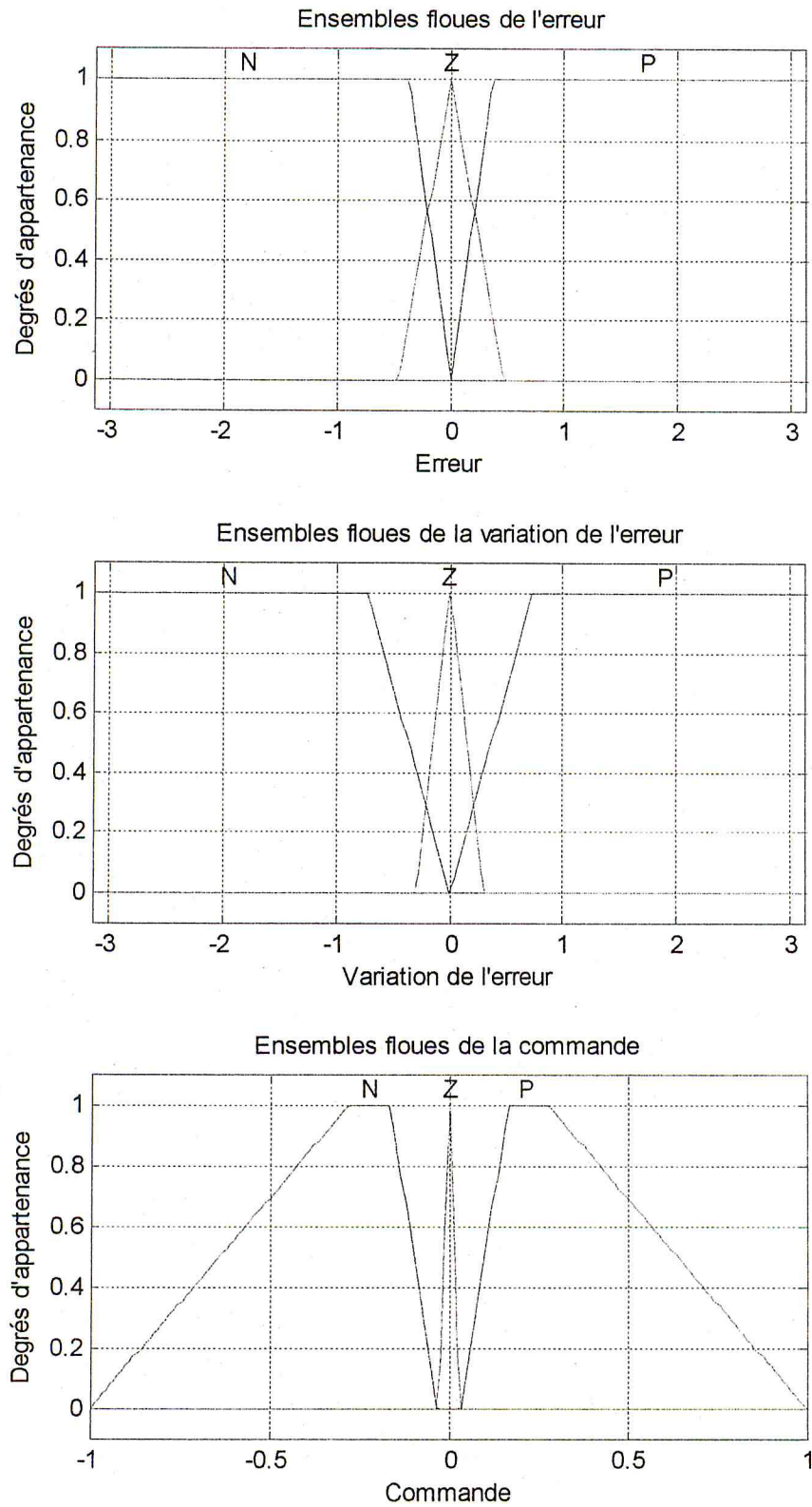


Figure IV.16 : Représentation des trois fonctions d'appartenance.

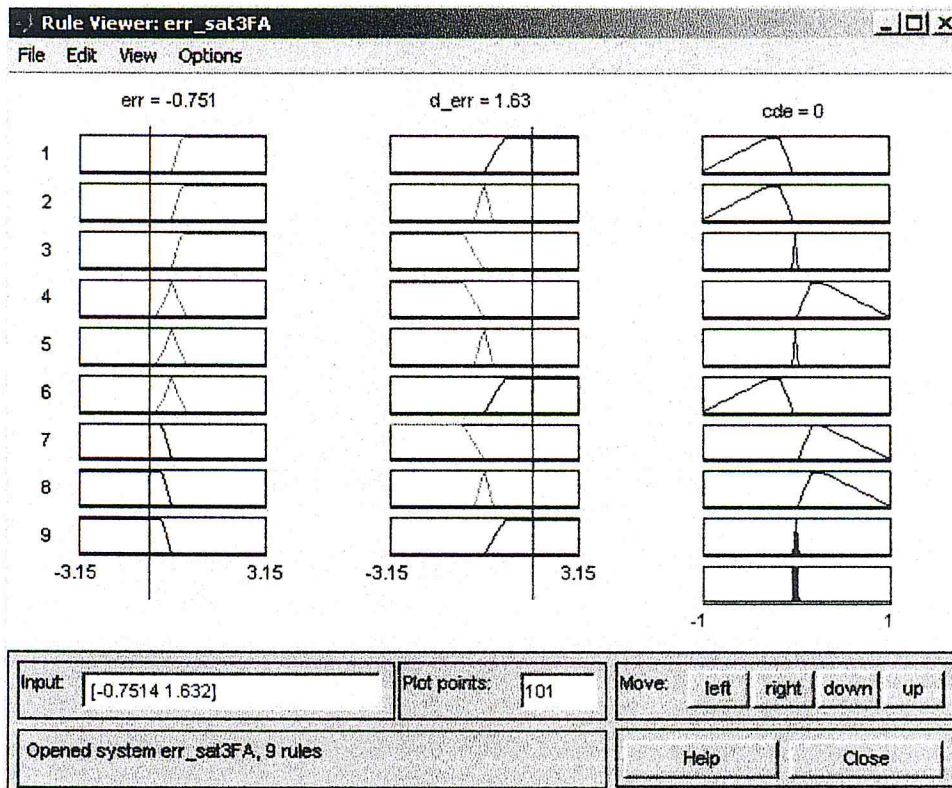


Figure IV.17 : Les règles pour trois fonctions d'appartenance.

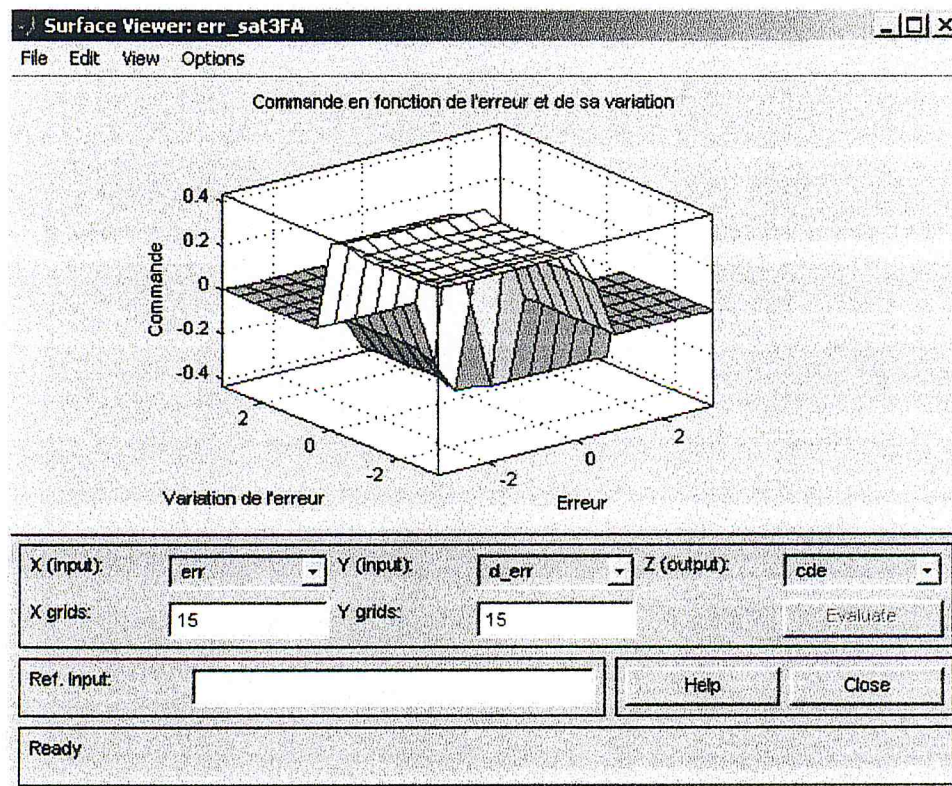


Figure IV.18 : La surface pour trois fonctions d'appartenance.

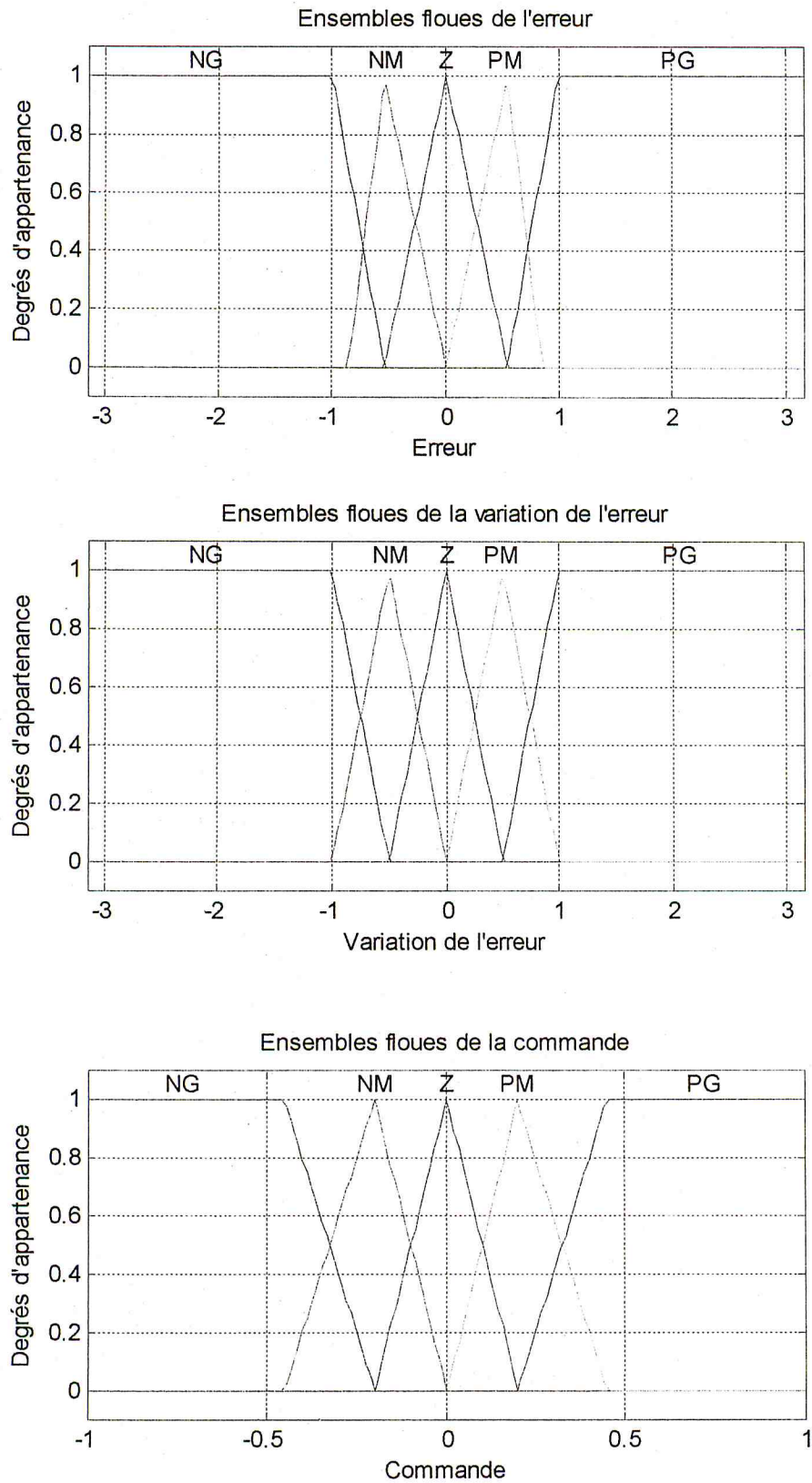


Figure IV.19 : Représentation des cinq fonctions d'appartenance.



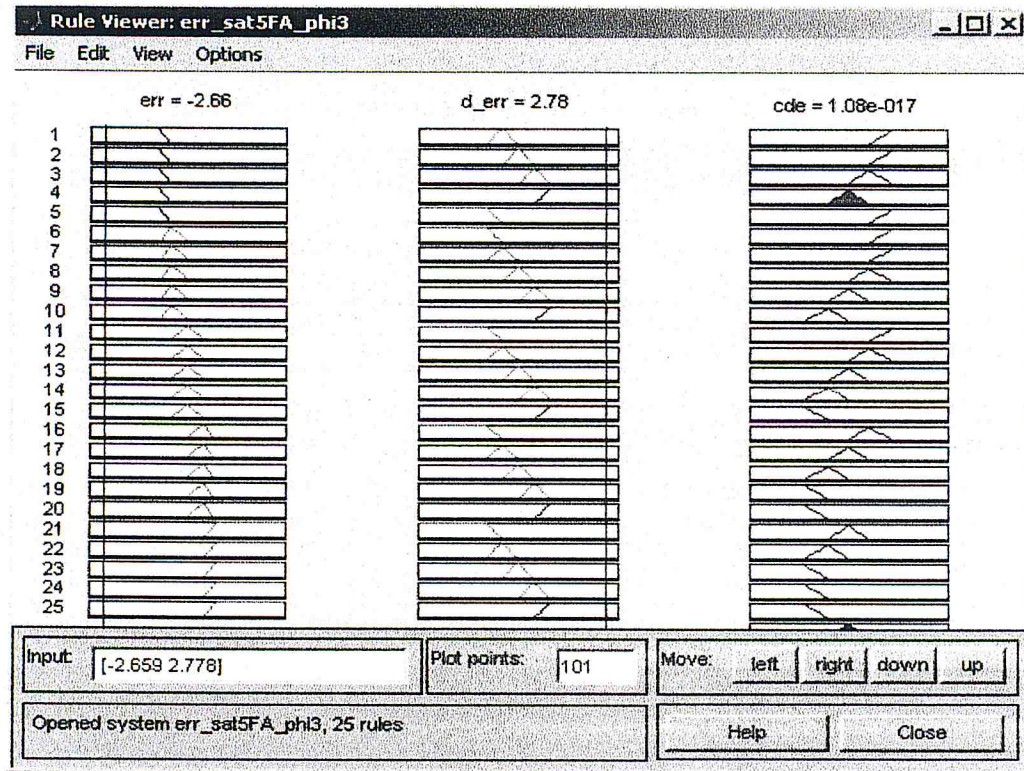


Figure IV.20 : Les règles pour cinq fonctions d'appartenance.

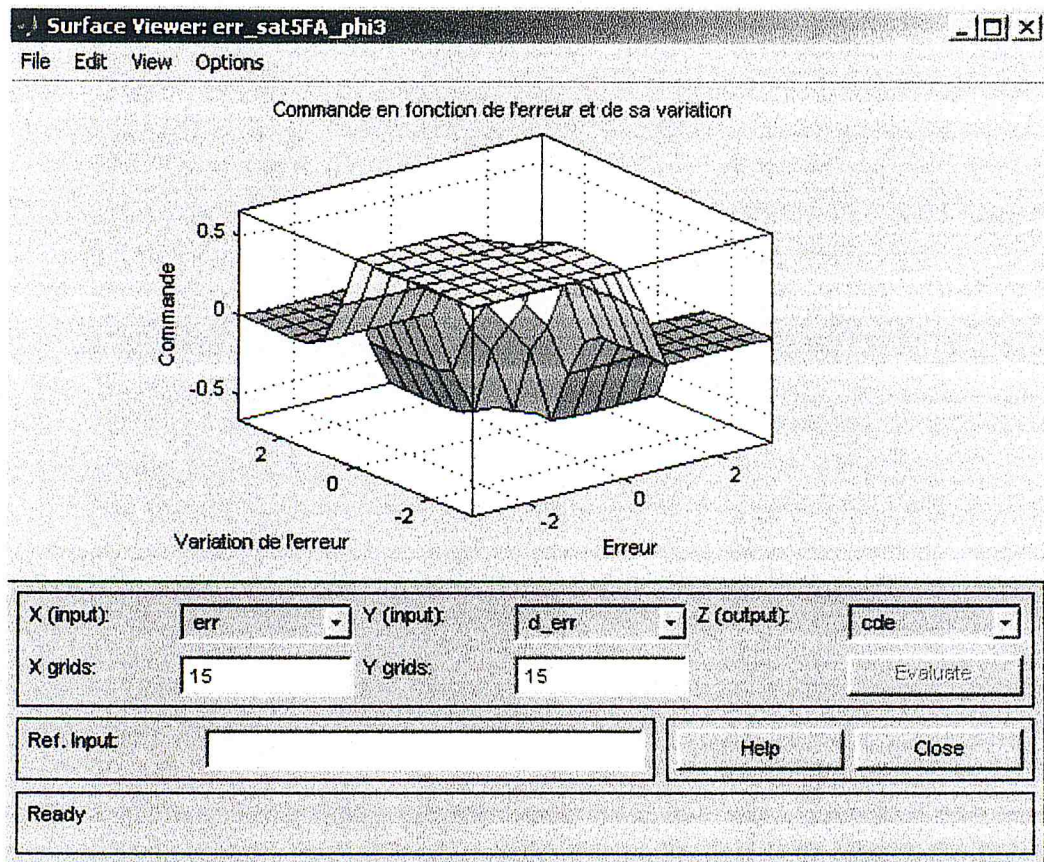


Figure IV.21 : La surface pour cinq fonctions d'appartenance.

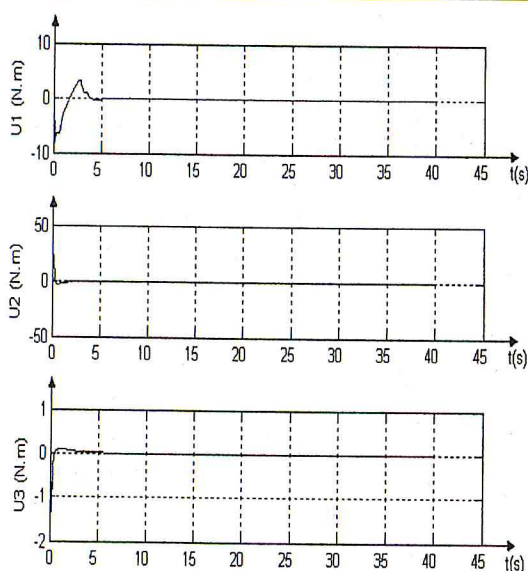


Figure IV.22.a : Commande en roulis, tangage et lacet dans le cas de trois fonctions d'appartenance.

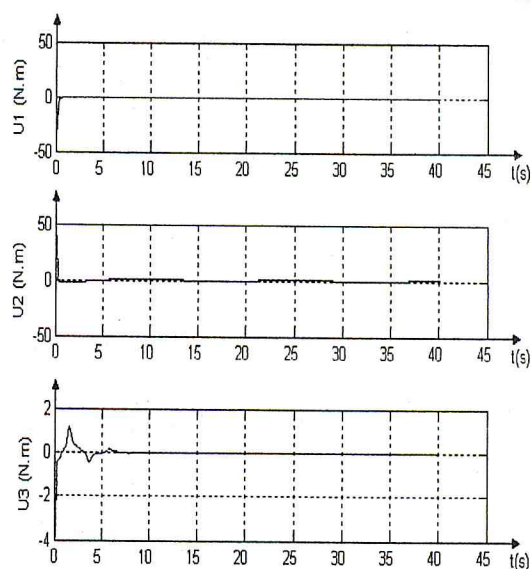


Figure IV.22.b : Commande en roulis, tangage et lacet dans le cas de cinq fonctions d'appartenance.

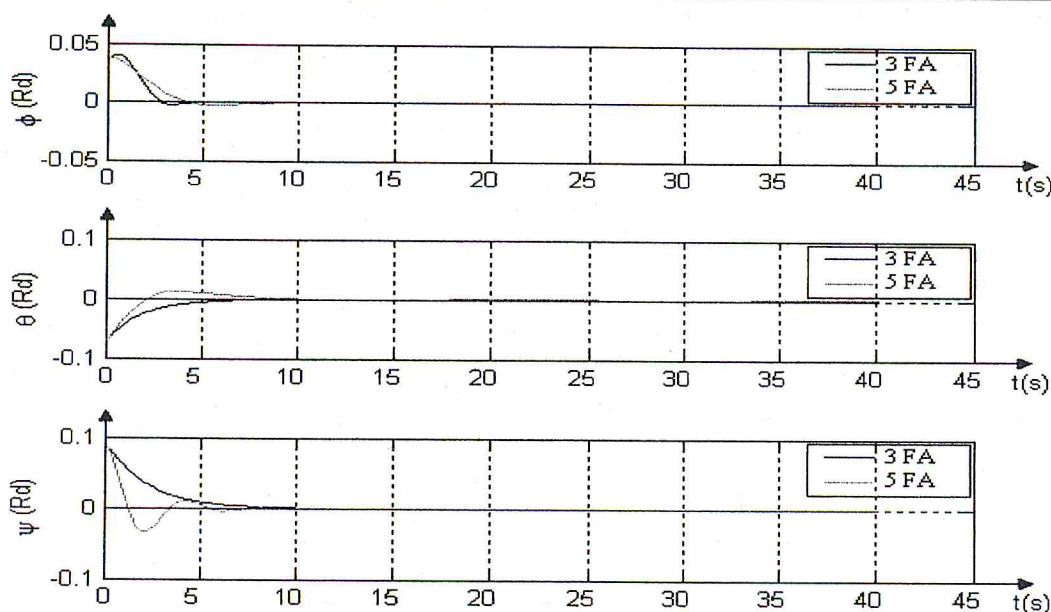


Figure IV.22.c : Attitude dans le cas de trois et cinq fonctions d'appartenance sans perturbation pour les conditions initiales  $[2^\circ, -4^\circ, 5^\circ]$  et une référence nulle.

Figure IV.22: Nous remarquons que chaque angle converge vers sa référence pour les deux cas telle que le dépassement des angles  $\theta$  et  $\psi$  dans le cas de trois fonctions d'appartenance ( $D = [116.0458, 0, 0]$  %) est inférieur à l'autre contrairement pour  $\phi$ , avec un dépassement  $D = [108.3095, 16.9054, 38.0298]$  %.

Mais la valeur maximal du couple de la commande est  $U_{\max} = [8.0236, 22.9623, 1.3586]$  N.m dans le premier cas (3FA) qui est inférieur aux deuxième cas  $U_{\max} = [28.7552, 38.8599, 3.8699]$  Nm.

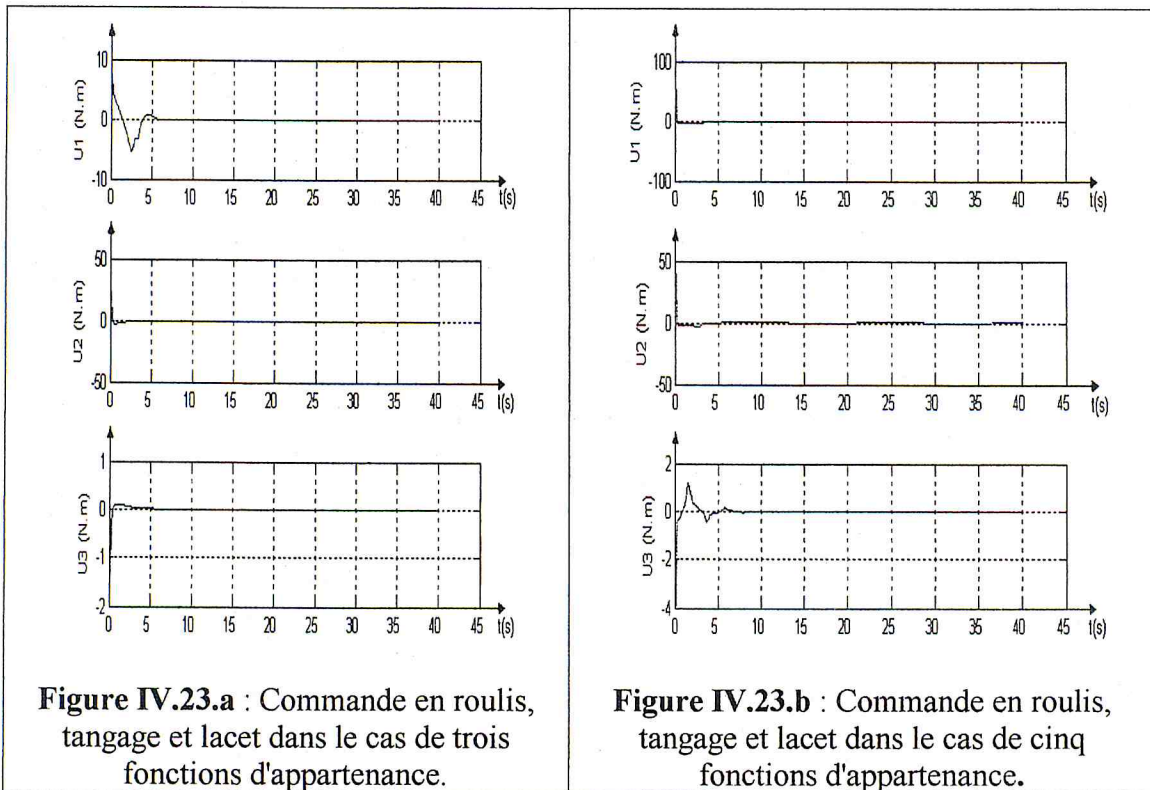


Figure IV.23.a : Commande en roulis, tangage et lacet dans le cas de trois fonctions d'appartenance.

Figure IV.23.b : Commande en roulis, tangage et lacet dans le cas de cinq fonctions d'appartenance.

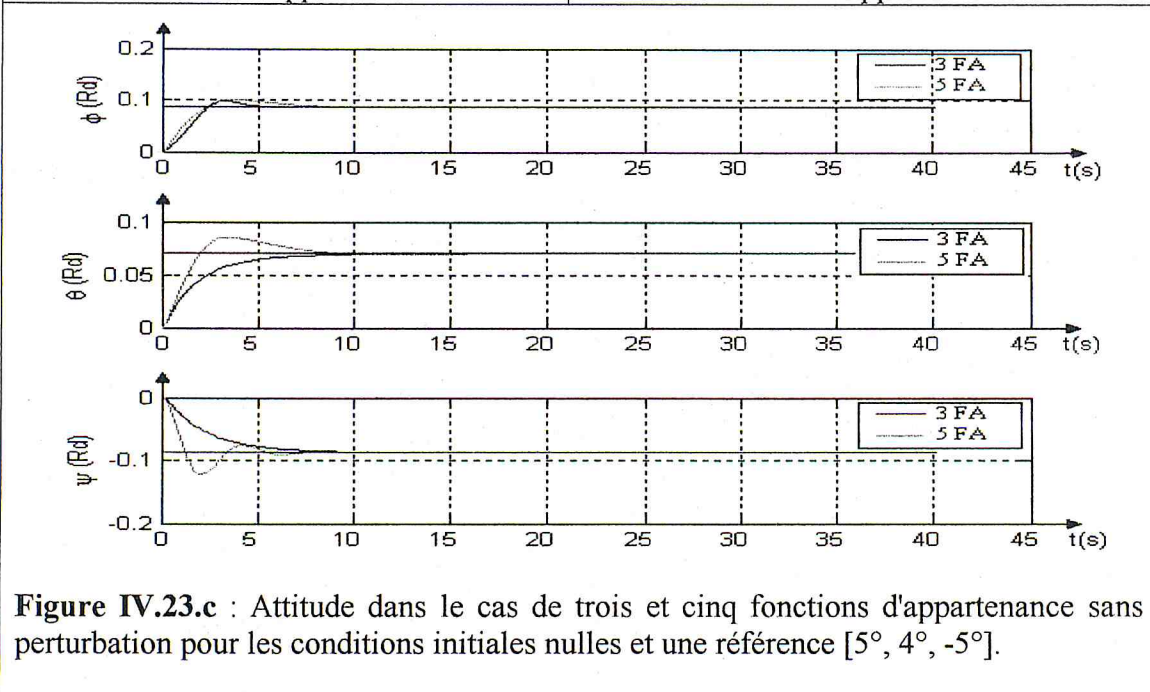
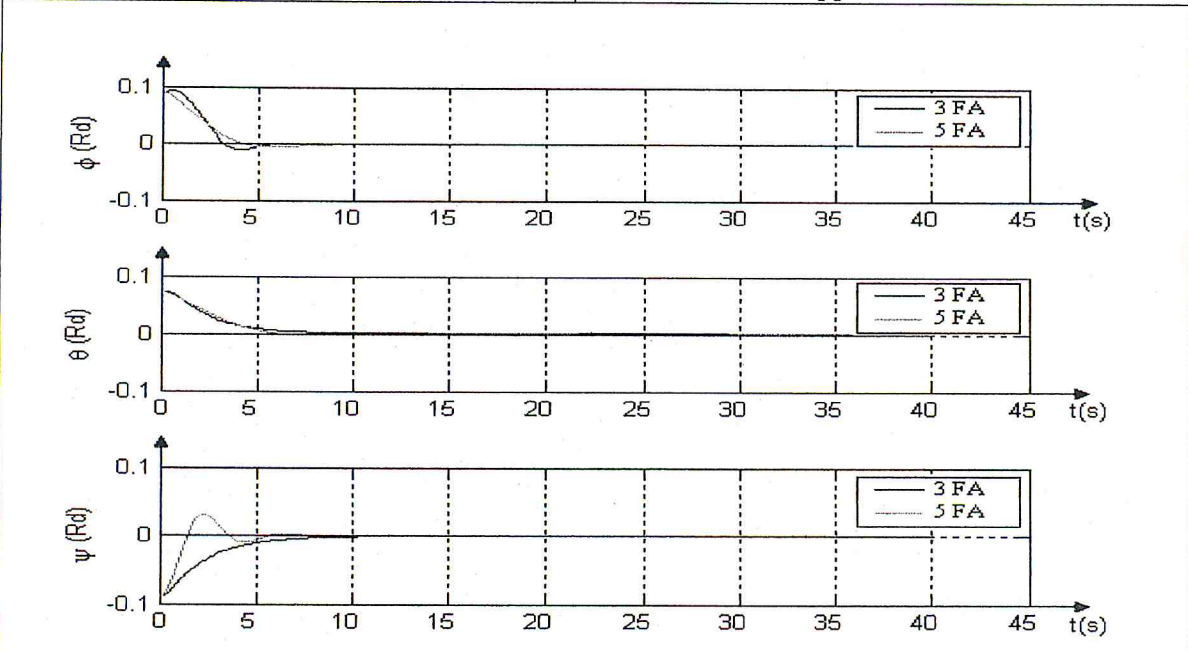
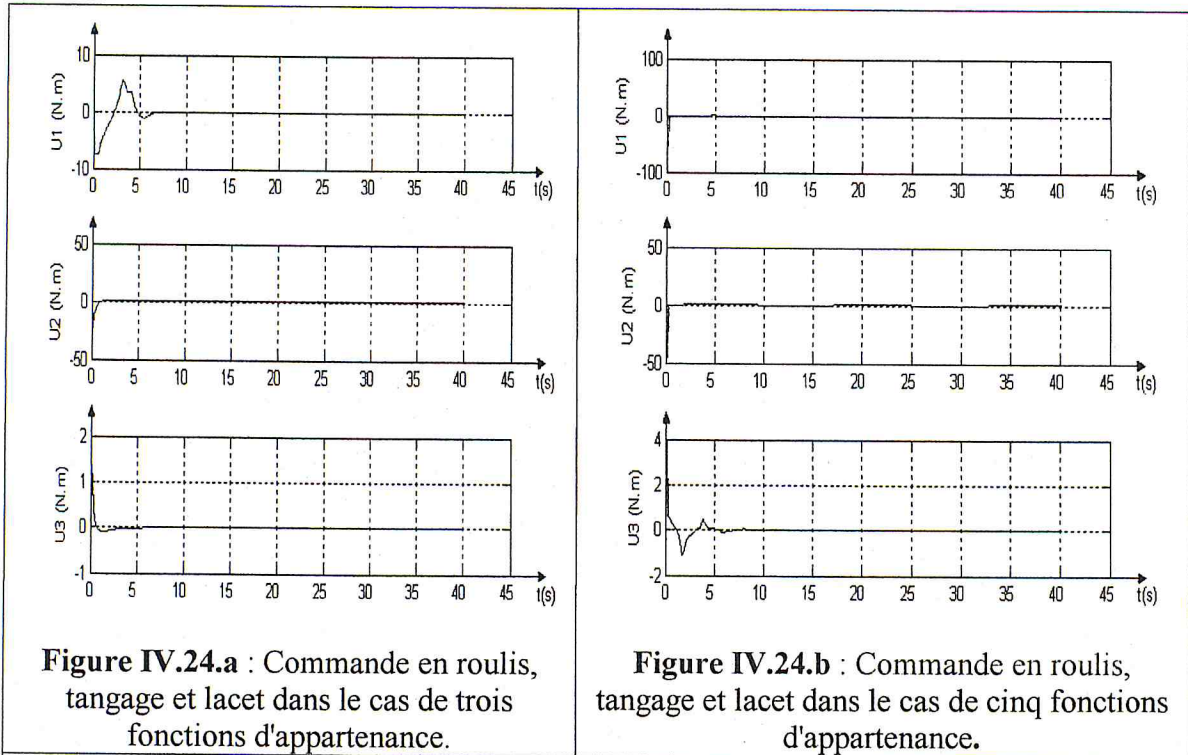
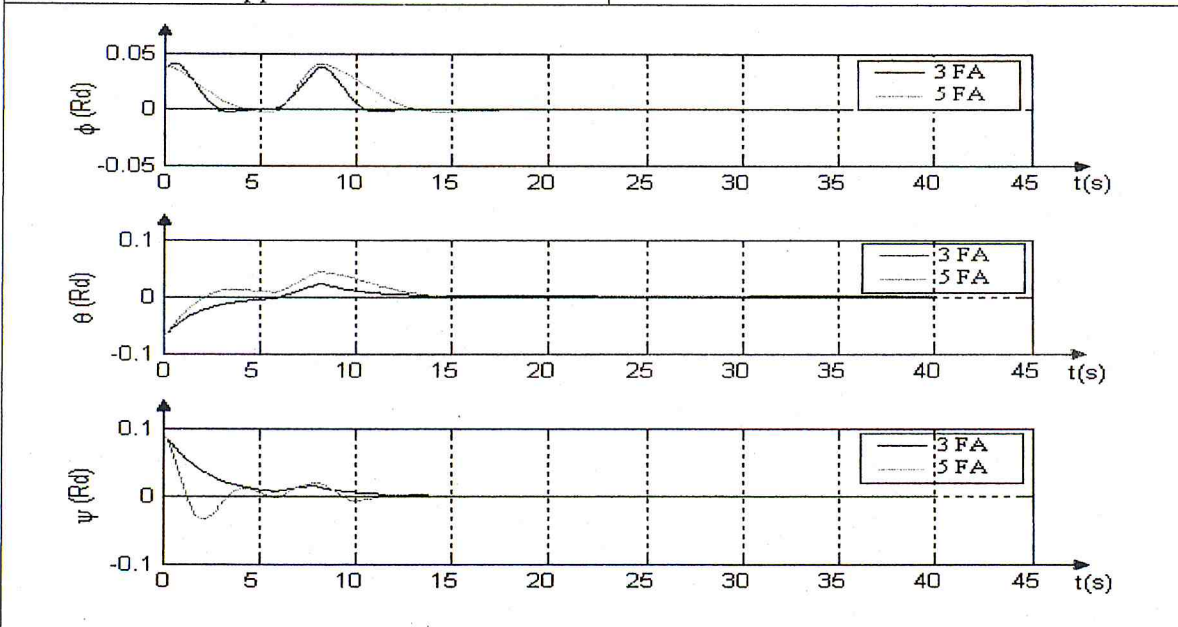
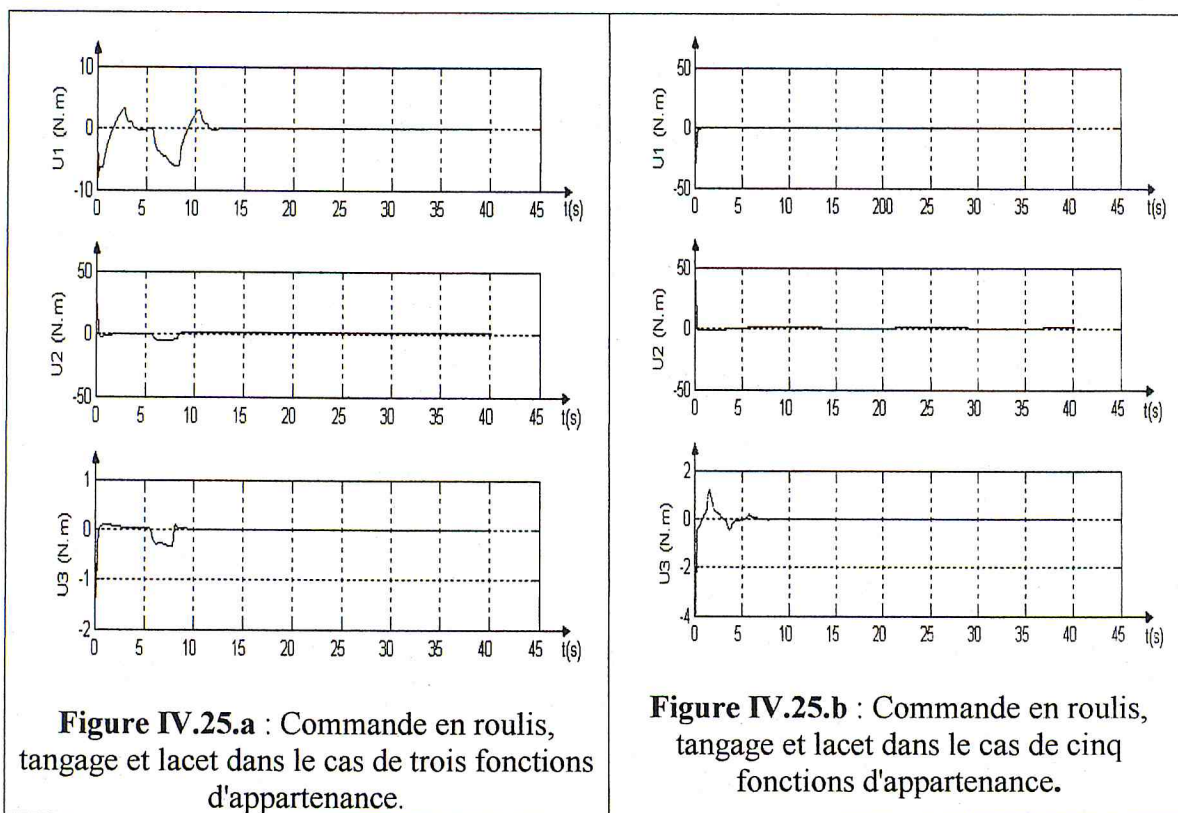


Figure IV.23.c : Attitude dans le cas de trois et cinq fonctions d'appartenance sans perturbation pour les conditions initiales nulles et une référence  $[5^\circ, 4^\circ, -5^\circ]$ .

Figure IV.23: Nous avons changés la référence et les conditions initiales, nous remarquons que le régime stationnaire n'est atteint quant aux valeurs des dépassements  $D = [11.5693, 0, 0] \%$  dans le cas de trois fonctions d'appartenance. Et dans le cas de cinq fonctions d'appartenance le dépassement est  $D = [15.0057, 22.0630, 39.0607] \%$  donc le dépassement dans le premier cas est inférieur au deuxième avec des valeurs maximales du commandes  $U_{\max} = [7.5289 \ 22.9623 \ 1.3585] \text{ Nm}$  avec un et la commande maximale pour le deuxième cas  $U_{\max} = [52.5645, 38.8599, 3.8699] \text{ Nm}$  est plus grande.



**Figure IV.24:** La commande maximale dans le deuxième cas  $U_{\max} = [54.3355, 45.0836, 3.9301]$  Nm est plus grande que celle du cas de trois fonctions d'appartenance ( $U_{\max} = [7.5353, 22.7896, 1.3587]$  Nm). Le dépassement des angles  $\theta$  et  $\psi$  du premier cas est nul ( $D = [106.1856, 0, 0]$  %) et inférieur aux dépassement des même angles pour le cas de cinq fonctions d'appartenance ( $D = [6.8729, 3.4384, 35.0515]$  %) et nous remarquons aussi que le dépassement d'angle  $\phi$  est très grand dans le premier cas.



**Figure IV.25.c :** Attitude dans le cas de trois et cinq fonctions d'appartenance sans perturbation pour les conditions initiales  $[2^\circ, -4^\circ, 5^\circ]$  et une référence nulle.

**Figure IV.25:** Correspond à la commande en présence d'un couple perturbateur constant  $p = [0.04, 0.04, 0.04]$  Nm, appliqué à partir de l'instant 55 s pendant 20 s, cette figure montre que le système rejette parfaitement ce couple perturbateur et nous remarquons qu'il n'est y a pas d'influence du couple perturbateur sur le couple maximal de la commande.

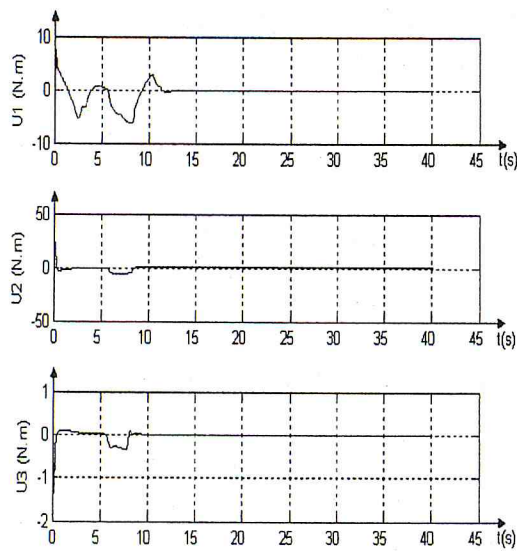


Figure IV.26.a : Commande en roulis, tangage et lacet dans le cas de trois fonctions d'appartenance.

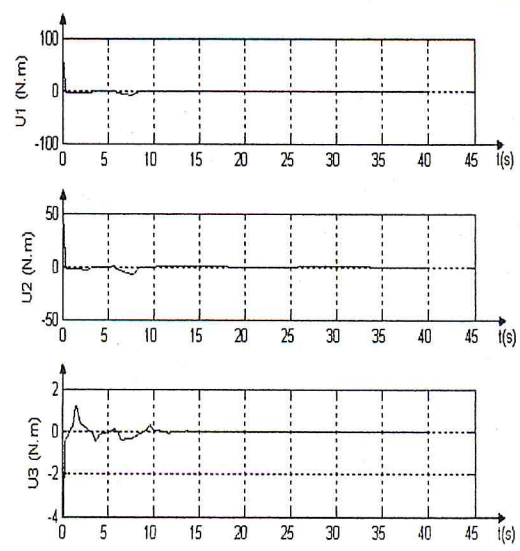


Figure IV.26.b : Commande en roulis, tangage et lacet dans le cas de cinq fonctions d'appartenance.

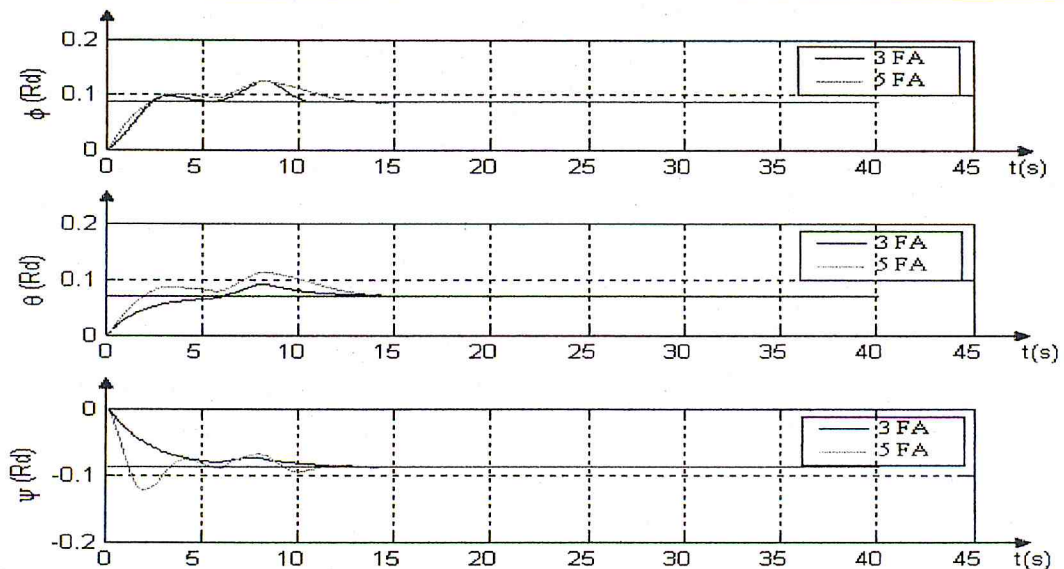
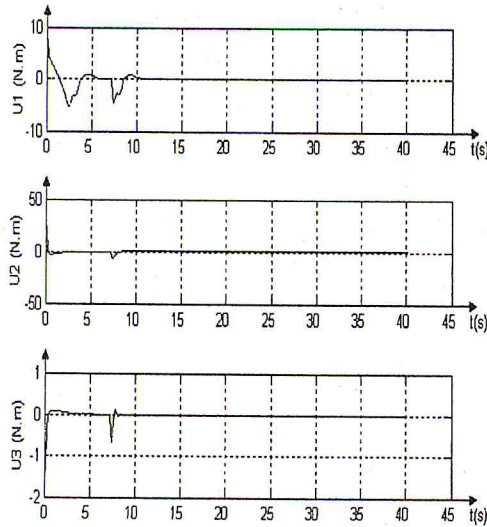


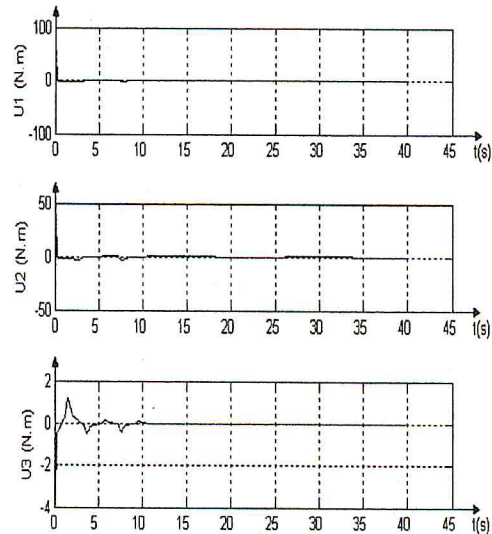
Figure IV.26.c : Attitude dans le cas de trois et cinq fonctions d'appartenance en présence d'un couple de perturbation constant  $p = [0.04, 0.04, 0.04]$  pour les conditions initiales nulle et une référence  $[5^\circ, 4^\circ, -5^\circ]$ .

Figure IV.26: Les valeurs maximales du couple de la commande appliqué afin de stabiliser le système reste les même dans les deux cas.

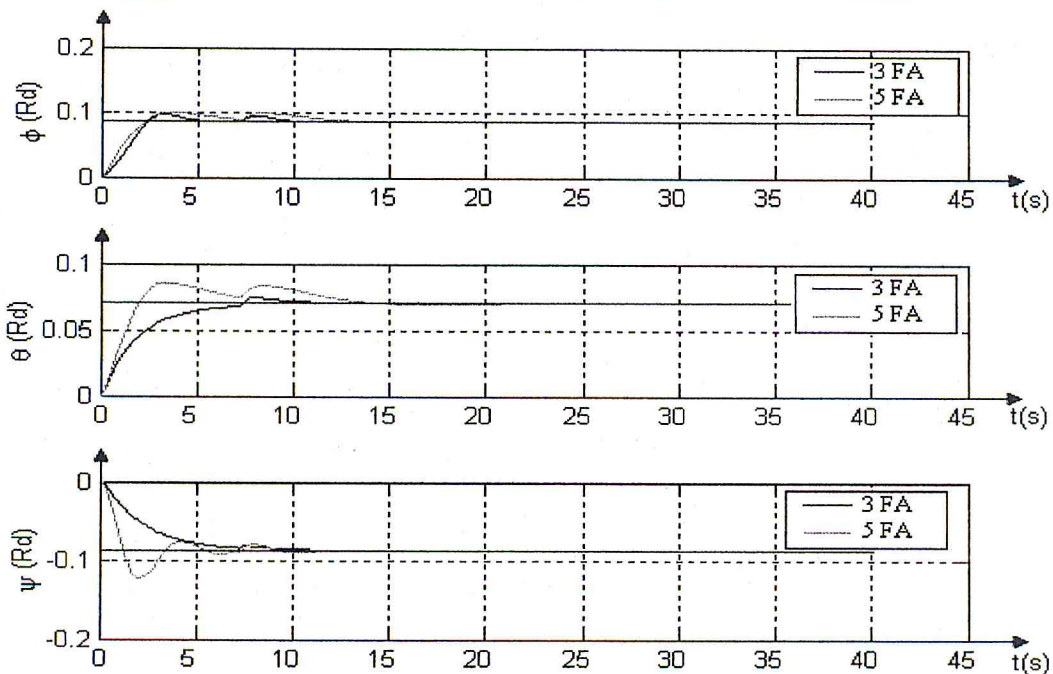
Pour le premier cas, le dépassement  $D = [43.4135, 31.0888, 16.7239] \%$ , nous remarquons qu'il a connue une augmentation pour l'angle  $\phi$  et une apparition pour les angles  $\theta$  et  $\psi$ . Tandis que les dépassements de tous les angles, dans le cas de cinq fonctions d'appartenance, sont augmenté  $D = [41.9244, 60.4585, 39.0607] \%$ .



**Figure IV.27.a :** Commande en roulis, tangage et lacet dans le cas de trois fonctions d'appartenance.



**Figure IV.27.b :** Commande en roulis, tangage et lacet dans le cas de cinq fonctions d'appartenance.



**Figure IV.27.c :** Attitude dans le cas de trois et cinq fonctions d'appartenance en présence d'un pic de perturbations égal à 0.2 Nm appliqué à  $t=70$  s, pour les conditions initiales nulle et une référence  $[5^\circ, 4^\circ, -5^\circ]$ .

**Figure IV.27:** Correspond à la commande en présence d'un pique de perturbation égal à 0.2 N.m à l'instant  $t=70$  s et montre que le système rejette parfaitement ce couple perturbateur dans les deux cas.

Et remarquons une apparition des dépassements des angles  $\theta$  et  $\psi$  de valeurs 6.3037 % et 7.7892 % respectivement dans le cas de trois fonctions d'appartenance.

## **Conclusion**

Le programme que nous avons mis au point peut servir à ajuster les centres et les largeurs des fonctions d'appartenance, et de déterminer les facteurs d'échelle. Cette étape a pour but de trouver une commande pour orienter les axes du satellite vers les directions requis.

Les résultats de simulation montrent que la commande floue a permis de stabiliser le satellite et que le meilleur rendement de ce contrôleur flou est obtenu dans le cas où les variables d'entrée et de sortie du système de commande sont fuzzifiées par trois variables linguistiques.

Nous remarquons, dans cette partie d'application, que la détermination des paramètres du contrôleur flou et les caractéristiques de l'algorithme de colonie de fourmis a été relativement facile.

Mais l'absence d'un détecteur et d'une expertise humaine dans le domaine des satellites artificiels est la seule difficulté, du fait qu'on a perdu beaucoup de temps pour la détermination de l'intervalle de variation des gains de normalisation des erreurs, ses dérivés et des commandes.



- [1] ABERKANE, AIT OUZZOU MOHAND, « Optimisation du rendement d'un contrôleur flou par les algorithmes génétiques », Mémoire d'ingénieur, Octobre 1999, Institut d'électronique (Université de BLIDA).
- [2] AHMED RACHID, « Système de régulation », Masson, Paris, 1996, p.p.236-269.
- [3] ALLICHE ZOUBIDA, BELLAG KHADIDJA, « Contrôle de la dynamique d'un Avion à l'aide d'un contrôleur hybride AG-Flou », Mémoire d'ingénieur, Novembre 2002, Institut d'électronique (Université de BLIDA).
- [4] AMAROUCHE HOCINE, OUKIL SAID, « Les algorithmes génétiques en contrôle », Mémoire d'ingénieur, Novembre 1998, Institut de mathématiques (Université de BLIDA).
- [5] AMR CHENTIR RABEH, SEBA BACHIR, « Desaturation » des roues à réaction et commande d'attitude d'un microsatellite », Mémoire d'ingénieur, Juin 2006, Institut d'automatique (Ecole Nationale Polytechnique).
- [6] ARIOUA LEYLA, BEZZAOUCHA SOUAD, « Modélisation et commande d'attitude de satellite artificiel », Mémoire d'ingénieur, Juin 2005, Institut d'automatique (Ecole Nationale Polytechnique).
- [7] BOUCHON-MEUNIER BERNARDETTE, MARSALA CHRISTOPHE, « Logique floue, Principe, Aide à la décision », Hermès science, Paris, 2003, p.p.17-38.
- [8] CHARON IRENE, GERMA ANNE, HUDRY OLIVIER « Méthodes D'optimisation Combinatoire »,
- [9] Cours de technologie spatiale: « Techniques et technologies des véhicules spatiaux » CNES (Centre National d'Etudes Spatiales) volume1 : généralités et contraintes de développement 1998
- [10] Cours de technologie spatiale: « Techniques et technologies des véhicules spatiaux » CNES (Centre National d'Etudes Spatiales) volume3 : Plates formes 1998.

- [11] DJAMILA ARIBI, FELLA DOUA, « Contrôle hybride Floue-AG d'une réaction parfaitement agitée », Mémoire d'ingénieur, Octobre 2001, Institut d'électronique (Université de BLIDA).
- [12] HANSRUEDI BUHLER, « Réglage Par Logique Floue », Presses polytechniques et universitaires romandes, 1994.
- [13] JEAN-RAPHAEL TONG-TONG, « La Logique Floue », Hermès science, Paris, 1995.
- [14] JELENA GADJEVAC, « Idées Nettes Sur La Logique Floue », Presses polytechniques et universitaires romandes, 1999.
- [15] LAURENT FOULLOY, SYLVIE GALICHET, ANDRE TITLI, « Commande floue 1 de la stabilisation à la supervision », Hermès science, Paris, 2003.
- [16] MECHOURED SAMIA, HASSAINA KHADIDJA, « La résolution du problème des voyageurs de commerce international ITCP par les algorithmes génétiques », Mémoire d'ingénieur, Octobre 1999, Institut de mathématiques (Université de BLIDA).
- [17] MICHEL CAPDEROU « Satellites orbites et missions », springer-verlag, France, 2003.
- [18] PIERRE YVES GLORENNEC, « Algorithmes d'apprentissage pour systèmes d'inférence floue », Hermès science, Paris, 1999.
- [19] SERGE POTTECK, « La conception de systèmes spatiaux », tome 1, système, paysage, mouvements, Toulouse, 1999.
- [20] WALTER GARNIELLI, « Multi-Valued logics », Hermès science, Paris, 1999.
- [21] <http://fleche.org/pierre/logfloue/index.html>
- [22] [http://www.tn.refer.org/hebergement/cours/logique\\_floue/intro.html](http://www.tn.refer.org/hebergement/cours/logique_floue/intro.html)
- [23] <http://fr.wikipedia.org/wiki/Optimisation>

[24] [http://artemis.univ-mrs.fr/cybermecca/Formcont/mecaspa/SCAO/IND\\_SCAO.htm](http://artemis.univ-mrs.fr/cybermecca/Formcont/mecaspa/SCAO/IND_SCAO.htm)

[25] [http://spot4.cnes.fr/spot4\\_fr/attitude.htm](http://spot4.cnes.fr/spot4_fr/attitude.htm)

