

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

Melle ZOUAOUI Hafida

Présidente du Jury

M^R TATI

M^R Benslama

Promoteur

M. A. BENCHERCHALI

Promotion 2014/2015

Dédicace

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

Je dédie ce travail également

A mes chers frères et sœurs, pour leur soutien, leur aide, leurs encouragements.

A tout mes enseignants, du primaire au niveau supérieur ;

Hafida Zouaoui

Remerciements

Au terme de ce travail, je tiens à remercier en premier lieu, le bon Dieu (Allah) qui m'a donné la force, la volonté et le courage pour terminer ce projet de fin d'étude.

Je remercie sincèrement Mr. BENCHARCHALI qui a été à l'origine de ce projet, pour sa rigueur, son suivi, et ses précieux conseils.

Je tiens à remercier aussi les membres du jury, Mr Tami et Mr Benslama.

Je tiens aussi à remercier tous ceux qui ont participé de près ou de loin à ma formation.

Table des matières

Introduction générale..... 01

Chapitre I

Meta heuristiques

I.1 Introduction..... 03
 I.2 Meta heuristiques 03
 I.2.1 optimisation difficile..... 03
 I.2.2 Algorithme d'optimisation approchée..... 04
 I.3 Algorithme de colonie de fourmis..... 06
 I.3.1 Optimisation naturelle..... 06
 I.3.2 Optimisation par colonie de fourmis et problème de voyageur de commerce.... 08
 I.3.3 Autres problèmes combinatoires..... 13
 I.3.4 Formalisation et propriétés d'un algorithme de colonie de fourmis..... 13

Chapitre II

les quaternions

II.1 Introduction..... 27
 II.2 Rappel sur la rotation instantanée..... 28
 II.3 Algèbre des quaternions 28
 II.4 Dynamique d'un satellite et quaternions d'attitude..... 31
 II.4.1 Repères -rotation-angles de cardan..... 40
 II.4.2 Expression des quaternions en fonction des angles de cardan..... 40

Chapitre III

La logique floue

II.1 Introduction..... 43
 II.2 Principe fondamental de la logique floue..... 43
 II.3 Les différents domaines d'application..... 44
 II.4 Bases générales de la logique floue..... 45
 II.4.1 Les sous-ensemble Flou..... 45
 II.4.2 Variables floues linguistiques..... 45
 II.4.2.1 Formes des fonctions d'appartenance..... 45
 II.4.2.2 Considérations générales sur les fonctions d'appartenances..... 47
 II.4.3 Dédutions floues (Inférences)..... 47
 II.4.3.1 Inférence avec une seule règle..... 47
 II.4.3.2 Inférence avec plusieurs règles..... 48
 II.4.4 Opérateurs de la logique floue..... 49

II.4.4.1	L'opérateur NON par le complémentaire de A.....	49
II.4.4.2	Opérateur ET.....	49.
II.4.4.3	Opérateur OU.....	50
II.5	Structure d'une commande floue.....	51
II.5.1	Généralité.....	51
II.5.2	Avantage et désavantages du réglage par logique floue.....	51.
II.5.3	Base de règles et définition.....	52
	- Partition floue.....	52
	- Base de règles.....	52
II.5.4	Procédure de raisonnement flou.....	53
	- Fuzzification.....	53
	- conclusion.....	56

Chapitre IV

Application

IV.1	Introduction.....
IV.2	Principe d'optimisation.....
IV.3	Problématique du sujet de l'adaptation des colonies de fourmis a la commande d'attitude.....	59
IV.4	Optimisation sous contraintes.....	61
IV.5	Description du système.....	62
IV.5.1	Étude du mouvement libre du satellite.....	63
IV.5.2	Étude du système en boucle fermée.....	64
	-Caractéristiques du contrôleur flou.....	64
IV.6	Application.....	65
IV.7	Courbes et interprétations.....	65
	Mouvement libre.....
	Utilisation du contrôleur flou.....	67
	*Mini satellite.....	67
IV.7	Conclusion.....	68

Conclusion générale.....	87
---------------------------------	-----------

Bibliographie

Annexes

Construction d'une station spatiale internationale, exploration martienne, constellations de satellites terrestres..., jamais les programmes spatiaux n'ont été si ambitieux.

Pourtant, l'état d'esprit dans lequel ils sont réalisés n'est plus celui de la course spatiale. Il n'est plus question de parvenir à tout prix à réaliser une mission, car les acteurs du domaine spatial sont devenus des exploitants. D'où l'apparition d'une notion relativement nouvelle dans ce secteur, celle de la rentabilité. Conséquence logique, la réduction des coûts dans le domaine spatial est devenu ces dernières années un enjeu majeur.

Cet objectif est réalisé principalement grâce à la réduction des temps de développement et à la miniaturisation des satellites. Ainsi, petit à petit, l'ingénierie spatiale se rapproche du domaine technique qui regroupe les domaines des microsystèmes, de l'optique intégrée, de la production industrielle et de la robotique sachant de plus que les systèmes robotiques mobiles dédiés à l'utilisation spatiale n'en sont qu'à leur balbutement et manquent encore singulièrement d'autonomie. Ces raisons, auxquels s'est joint notre désir d'étendre nos connaissances aux applications spatiales, ont motivé la réalisation de notre travail.

Pour décrire précisément la position d'un satellite sur une orbite terrestre, il faut un certain nombre de paramètres, appelés éléments Képlériens. Les premiers d'entre eux décrivent l'orbite elle-même par rapport à la terre et au soleil, les seconds les coordonnées du satellite sur cette orbite.

Une orbite terrestre est définie par une ellipse dont un des deux foyers se trouve sur le centre de gravité de la terre. Le point de cette orbite le plus éloigné de la terre se nomme l'*Apogée* et le point le plus proche le *Périgée*. Cette orbite se trouve inclinée par rapport à un plan passant par l'équateur terrestre (Figure 1).

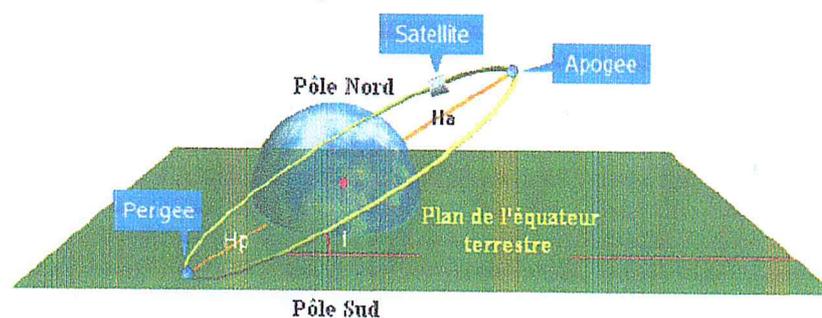


Figure 1 : Apogée, périgée et inclinaison d'une orbite terrestre.

L'*inclinaison* « i » est un paramètre important car il définit la latitude géographique maximale, à la verticale de laquelle le satellite pourra passer. De plus, comme l'*inclinaison* dépend intimement de la latitude du lieu de lancement puisqu'il n'est pas possible de lui donner une valeur inférieure sans de coûteuses corrections ultérieures, les exploitants s'appuient sur ce principal argument pour imposer le placement des sites de lancement le plus proche possible de l'équateur, à l'instar de Cap Canaveral (28.5° N) ou de Kourou (5.2° N).

Notre projet fait suite à l'étude et la modélisation d'un satellite afin de commander son attitude sur sa propre orbite. L'implémentation de loi d'asservissements appropriés permettra l'orientation par pointage des axes du satellite vers des directions souhaitées. Cette étude faite suite à un travail effectué précédemment englobe plusieurs aspects, en ce qui nous concerne. Ce genre de problème ayant déjà fait l'objet d'une étude de modélisation et de commande. [6]

Le choix d'une stratégie de commande par contrôleur flou a été imposée par notre cahier de charge afin d'optimiser son rendement par des algorithmes génétiques. Pour ce faire notre mémoire activera cinq volets délimitant les principaux chapitres que nous avons manipulés durant notre parcours de thèse.

Dans un premier temps, nous nous intéressons à la définition du méta heuristique dans le premier chapitre. Dans le second, on définira les quaternions et leur expression en fonction des angles de Cardan. La présentation des bases et des propriétés essentielles de la logique floue entrant dans l'organisation générale et la structure du régulateur par logique floue fait l'objet du troisième chapitre. Le quatrième conclura l'objet de ce mémoire par la présentation des applications.

Chapitre 1

Le monde des colonies de fourmis en optimisation

L'optimisation est un sujet central en recherche opérationnelle. Un grand nombre de problèmes de décision pouvant en effet être décrits sous la forme de problèmes d'optimisation, les problèmes d'identification, l'apprentissage supervisé de réseaux de neurones et la recherche du plus court chemin sont par exemple, des problèmes d'optimisation.

Le présent chapitre présente tout d'abord le cadre de l'optimisation difficile et des méta heuristiques. Nous présentons ensuite les algorithmes de colonies de fourmis, puis présentons un état de l'art sur les méta heuristiques « colonies de fourmis ».

Les algorithmes de colonies de fourmis forment une classe des méta heuristiques proposée pour l'optimisation difficile. Ces algorithmes s'inspirent des comportements de recherche et de suivi de piste observés dans les colonies de fourmis. Une colonie de fourmis (les fourmis) communiquent indirectement via des modifications de leur environnement (les pistes de phéromones) et construisent ainsi une solution optimale, s'appuyant sur leur expérience collective.

Problèmes pour 'l'optimisation difficile'

1.1 Optimisation difficile

1.1.1 Définition de l'optimisation difficile

L'optimisation au sens général est définie par un ensemble de solutions possibles. La qualité peut être décrite par une fonction objectif f . On cherche alors à trouver s^* possédant la meilleure qualité $f(s^*)$ (par la suite, on peut chercher à minimiser $f(s)$). Un problème d'optimisation peut présenter des contraintes (égalité) sur s , être dynamique si $f(s)$ change avec le temps ou encore plusieurs fonctions objectifs doivent être optimisées.

Les problèmes d'optimisation sont souvent déterministes (dites « exactes ») permettant un certain nombre de propriétés de la fonction objectif, comme la stricte convexité, la continuité ou encore la différentiabilité. On peut citer comme exemple de méthode de Newton, etc.

1.1.2 Problèmes d'optimisation difficile

Les problèmes d'optimisation difficile demeurent cependant hors de portée des méthodes classiques. Un grand nombre de caractéristiques peuvent en effet être problématiques, comme la non-convexité stricte (multimodalité), l'existence de discontinuité, une fonction non différentiable, le bruit, etc.

Chapitre 1

Dans de tels cas, le problème d'optimisation est dit « difficile », car aucune méthode exacte n'est capable de le résoudre exactement en un temps « raisonnable », on devra alors faire appel à des heuristiques permettant une optimisation approchée.

L'optimisation difficile peut se découper en deux types de problèmes : les problèmes discrets et les problèmes continus . le premier cas rassemble les problèmes de type NP-complets, tels que le problème du voyageur de commerce. Un problème « NP » est dit complet s'il est possible de le décrire à l'aide d'un algorithme polynomial sous la forme d'un sous-ensemble d'instances . concrètement , il est facile de décrire une solution à un tel problème, mais le nombre de solutions nécessaires à la résolution croit de manière exponentielle avec la taille de l'instance. Jusqu'à présent, la conjecture postulant que les problèmes NP-complets ne sont pas solubles en un temps polynomial n'a été ni démontrée , ni révoquée . Aucun algorithme polynomial de résolution n'a cependant été trouvé pour de tels problèmes. L'utilisation d'algorithmes d'optimisation permettant de trouver une solution approchée en un temps raisonnable est donc courante.

Dans la seconde catégorie , les variables du problème d'optimisation sont continues . c'est le cas par exemple des problèmes d'identification , où l'on cherche à minimiser l'erreur entre le modèle d'un système et des observations expérimentales . ce type de problème est moins formalisé que le précédent , mais un certain nombre de difficultés sont bien connues , comme l'existence de nombreuses variables présentant des corrélations non identifiées , la présence de bruit ou plus généralement une fonction objectif accessible par simulation uniquement . En pratique, certains problèmes sont mixtes et présentent à la fois des variables discrètes et des variables continues.

1.2.2 Algorithmes d'optimisation approchée

1.2.2.1 Heuristique

Une heuristique d'optimisation est une méthode approchée se voulant simple , rapide et adaptée à un problème donné. Sa capacité à optimiser un problème avec un minimum d'informations est contrebalancée par le fait qu'elle n'offre aucune garantie quant à l'optimalité de la meilleure solution trouvée.

Du point de vue de la recherche opérationnelle , ce défaut n'est pas toujours un problème , tout spécialement quand seule une approximation de la solution optimale est recherchée.

1.2.2.2 Méta heuristiques

Parmi les heuristiques , certaines sont adaptables à un grand nombre de problème différents sans changement majeurs dans l'algorithme, on parle alors de méta-heuristiques .

La plupart des heuristiques et des méta heuristiques utilisent des processus aléatoires comme moyen de récolter de l'information de de faire face à des problèmes comme l'explosion combinatoire. En plus de cette base stochastique, les méta heuristiques sont

Chapitre 1

généralement itératives , c'est-à-dire qu'un même schéma de recherche est appliqué plusieurs fois au cours de l'optimisation , et directes , c'est-à-dire qu'elles n'utilisent pas l'information du gradient de la fonction objectif. Elles tirent en particulier leur intérêt de leur capacité à éviter les optima locaux , soit en acceptant une dégradation de la fonction objectif au cours de leur progression , soit en utilisant une population de points comme méthode de recherche (se démarquant ainsi des heuristiques de descente locale).

Souvent inspirées d'analogies avec la réalité (physique , biologie , éthologie ...) , elles sont généralement conçues au départ pour des problèmes discrets, mais peuvent faire l'objet d'adaptations pour des problèmes continus .

Les méta heuristiques, du fait de leur capacité à être utilisées sur un grand nombre de problèmes différents , se prêtent facilement à des extensions . pour illustrer cette caractéristique , citons notamment :

L'optimisation multi objectif (dites aussi multicritère) , où il faut optimiser plusieurs objectifs contradictoires . La recherche vise alors non pas à trouver un optimum global , mais un ensemble d'optima « au sens de Pareto » formant le « surface de compromis » du problème.

L'optimisation multimodale, où l'on cherche un ensemble des meilleures optima globaux et ou locaux.

L'optimisation des problèmes bruités , où il existe une incertitude sur le calcul de la fonction objectif. Incertitude dont il faut alors tenir compte dans la recherche de l'optimum .

L'optimisation dynamique où la fonction objectif varie dans le temps .Il faut alors approcher au mieux l'optimum à chaque pas de temps.

La parallélisation , où l'on cherche à accélérer la vitesse de l'optimisation en répartissant la charge de calcul sur des unités fonctionnant de concert . le problème revient alors à adapter les méta heuristiques pour qu'elles soient distribuées .

L'hybridation , qui vise à tirer parti des avantages respectifs méta heuristiques différentes en les combinant .

Enfin , la grande vitalité de ce domaine de recherche ne doit pas faire oublier qu'un des intérêts majeurs des méta heuristiques est leur facilité d'utilisation dans des problèmes concrets. L'utilisateur est généralement demandeur de méthodes efficaces permettant d'atteindre un optimum avec une précision acceptable dans un temps raisonnable . Un des enjeux de la conception des métaheuristiques est donc de faciliter le choix d'une méthode et de simplifier son réglage pour l'adapter à un problème donné.

Du fait du foisonnement de la recherche, un grand nombre de classes de méta heuristiques existent , on pourra citer comme exemple, le recuit simulé, les algorithmes évolutionnaires , la recherche avec tabous ou la méta heuristique dite 'de colonies de fourmis' objet de notre étude dans ce chapitre.

Chapitre 1

1.3 Algorithmes de colonies de fourmis en optimisation

Le premier algorithme de ce type (Ant System) a été conçu pour le problème du voyageur de commerce , mais n' a pas permis de produire des résultats compétitifs . cependant , l'intérêt pour le métaphore était lancé et de nombreux algorithmes s'en inspirant ont depuis été proposés dans divers domaines , certains atteignant des résultats très convaincants.

1.3.1 Optimisation naturelle : piste de phéromone

Les algorithmes de colonies de fourmis sont nés à la suite d'une constatation : les insectes sociaux en général , et les fourmis en particulier, résolvent naturellement des problèmes relativement complexes. Les biologistes ont étudié comment les fourmis arrivent à résoudre collectivement des problèmes trop complexes pour un seul individu , notamment les problèmes de choix lors de l'exploitation de sources de nourriture.

Les fourmis ont la particularité d'employer pour communiquer des substances volatiles appelées phéromones . Elles sont attirées par ces substances , qu'elles perçoivent grâce à des récepteurs situés dans leurs antennes . Ces substances sont nombreuses et varient selon leur espèces . Les fourmis peuvent déposer des phéromones au sol , grâce à une glande située dans leur abdomen , et former ainsi des pistes odorantes , qui pourront être suivies par leurs congénères , comme indiqué dans la figure ci-dessous.



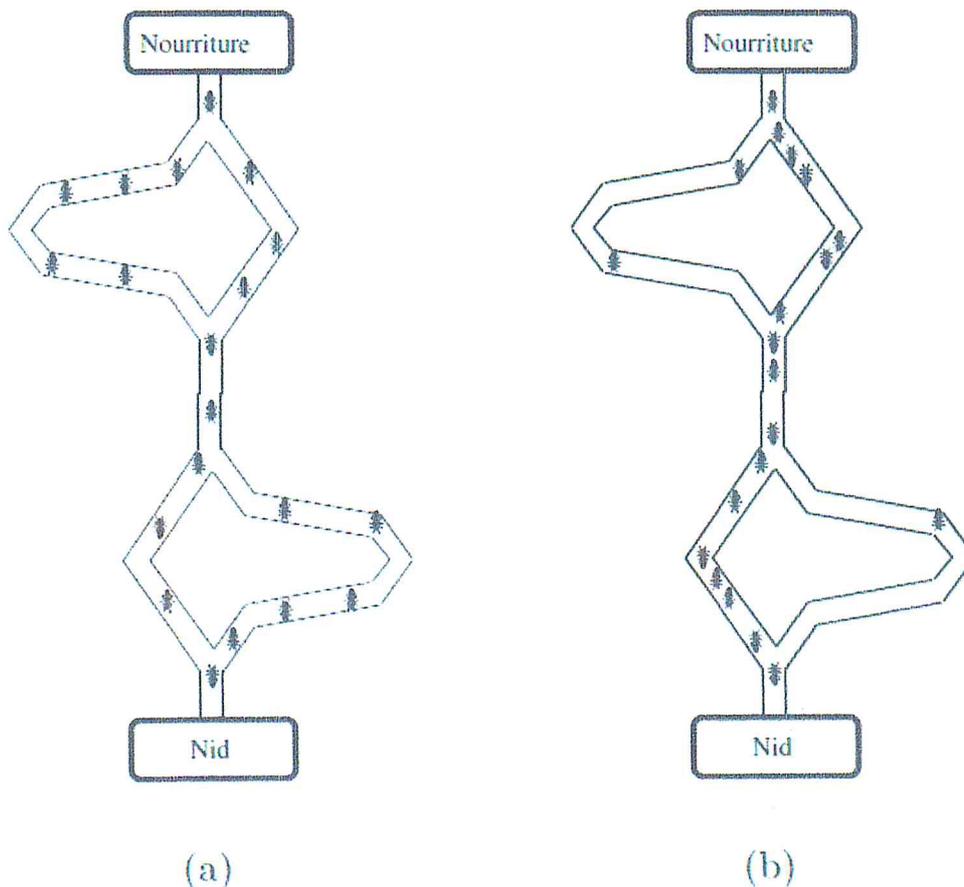
Figure 1.1 Des fourmis suivant une piste de phéromones

Chapitre 1

Les fourmis utilisent les pistes de phéromones pour marquer leur trajet , par exemple entre le nid et une source de nourriture . Une colonie est ainsi capable de choisir (sous certaines conditions) le plus court chemin vers une source à exploiter sans que les individus aient une vision globale du trajet.

En effet , comme l'illustre la figure 1.2 , les fourmis le plus rapidement arrivées au nid, après avoir visité la source de nourriture , sont celles qui empruntent les deux branches les plus courtes . Ainsi la quantité de phéromone présente sur le plus court trajet est légèrement plus importante que celle présente sur le chemin le plus long. Or , une piste présentant une plus grande concentration en phéromones est plus attirante pour les fourmis , elle a une probabilité plus grande d'être empruntée . La piste courte va alors être renforcée que la longue , et, à terme, sera choisie par la grande majorité des fourmis .

On constate qu'ici le choix s'opère par un mécanisme d'amplification d'une fluctuation initiale. Cependant , il est possible qu'en cas d'une plus grande quantité de phéromones déposée sur les grandes branches, au début de l'expérience, la colonie choisisse le plus long parcours.



Chapitre 1

Figure 1.2 Expérience de sélection des branches les plus courtes par une colonie de fourmis (a) au début de l'expérience ; (b) à la fin de l'expérience

D'autres expériences [Beckers et al, 1992], avec une autre espèce de fourmis, ont montré que si les fourmis sont capables d'effectuer des demi-tours sur la base d'un trop grand écart par rapport à la direction de la source de nourriture, alors la colonie est plus flexible et le risque d'être piégé sur le chemin long est plus faible.

Il est difficile de connaître avec précision les propriétés physico-chimiques des pistes de phéromones, qui varient en fonction des espèces et d'un grand nombre de paramètres. Cependant, les méta heuristiques d'optimisation de colonies de fourmis s'appuient en grande partie sur le phénomène d'évaporation des pistes de phéromones. Or, on constate dans la nature que les pistes s'évaporent plus lentement que ne prévoient les modèles. Les fourmis réelles disposent en effet d'heuristiques leur apportant un peu plus d'informations sur le problème (par exemple une information sur la direction). Il faut garder à l'esprit que l'intérêt immédiat de la colonie (trouver le plus court chemin vers une source de nourriture) peut être en concurrence avec l'intérêt adaptatif de tels comportement. Si l'on prend en compte l'ensemble des contraintes que subit une colonie de fourmis (prédation, compétition avec d'autres colonies, etc), un choix rapide et stable peut être meilleur, et un changement de site exploité peut entraîner des coûts trop forts pour permettre la sélection naturelle d'une telle option.

1.3.2 Optimisation par colonie de fourmis et problème du voyageur de commerce

Le problème du voyageur de commerce (Travelling Salesman Problem, TSP) a fait l'objet de la première implémentation d'un algorithme de colonies de fourmis : le Ant System (AS) [Coloni et al ..1992]. Le passage de la métaphore à l'algorithme est ici relativement facile et le problème du voyageur de commerce est bien connu et étudié. Il est intéressant d'approfondir le principe de ce premier algorithme pour bien comprendre le mode de fonctionnement des algorithmes de colonies de fourmis. Il y a deux façons d'aborder ces algorithmes. La première, la plus évidente eu premier abord, est celle qui a historiquement mené au 'Ant System' original ; nous avons choisi de la décrire dans cette section. La seconde est une description plus formelle des mécanismes communs aux algorithmes de colonies de fourmis, elle sera décrite dans la section 1.3.4.

Le problème du voyageur de commerce consiste à trouver le trajet le plus court (désigné par 'tournee' ou plus loin par 'tour') reliant n villes données, chaque ville ne devant être qu'une seule fois. Le problème est plus généralement défini comme un graphe complètement connecté (N,A) , où les villes sont des nœuds N et les trajets entre ces villes, les arêtes A .

1.3.2.1 Algorithme de base

Dans l'algorithme AS, à chaque itération t ($1 \leq t \leq t_{max}$), chaque fourmi k ($k = 1, \dots, m$) parcourt le graphe et construit un trajet complet de $n = |N|$ étapes (on note $|N|$ le cardinal de l'ensemble N). Pour chaque fourmi, le trajet entre une ville i et une ville j dépend de :

Chapitre 1

1. La liste des villes déjà visitées , qui définit les mouvements possibles à chaque pas, quand la fourmi k est sur le ville i : J_i^k ;
2. L'inverse de la distance entre les villes : $n_{ij} = \frac{1}{d_{ij}}$, appelée visibilité . cette information 'statique' est utilisée pour diriger le choix des fourmis vers des villes proches , et éviter les villes trop lointaines ;
3. La quantité de phéromones déposée sur l'arête reliant les deux villes , appelée l'intensité de la piste . Ce paramètre définit l'attractivité d'une partir du trajet global et change à chaque passage d'une fourmi . C'est , en quelque sorte, une mémoire globale du système , qui évolue par apprentissage .

La règle de déplacement (appelée ' règle aléatoire de transition proportionnelle ' par les auteurs [Bonabeau et al , 1999]) est la suivante :

$$p_{ij}^k(t) = \begin{cases} \frac{(\tau_{ij}(t))^\alpha \cdot (n_{ij})^\beta}{\sum_{l \in J_i^k} (\tau_{il}(t))^\alpha \cdot (n_{il})^\beta} & \text{si } j \in J_i^k \\ 0 & \text{si } j \notin J_i^k \end{cases} \quad (1.1)$$

Où α et β sont deux paramètres contrôlant l'importance relative de l'intensité de la piste, $\tau_{ij}(t)$, et de la visibilité , n_{ij} . Avec $\alpha = 0$, seule la visibilité de la ville est prise en compte ; la ville la plus proche est donc choisie à chaque pas . Au contraire avec $\beta = 0$ seules les pistes de phéromones jouent . Pour éviter une sélection trop rapide d'un trajet , un compromis entre ces deux paramètres , jouant sur le comportement de diversification et d'intensification , est nécessaire.

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

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)} & \text{si } (i,j) \in T^k(t) \\ 0 & \text{si } (i,j) \notin T^k(t) \end{cases} \quad (1.2)$$

Où $T^k(t)$ est le trajet effectué par la fourmi k à l'itération t , $L^k(t)$ la longueur de la tournée et Q un paramètre fixé.

L'algorithme ne serait pas complet sans le processus d'évaporation des pistes de phéromones . En effet, pour éviter d'être piégé dans des solutions sous-optimales , il est nécessaire de permettre au système 'd'oublier' les mauvaises solutions. On contre balance

Chapitre 1

donc l'additivité des pistes par une décroissance constante des valeurs des arêtes à chaque itération. La règle de mise à jour des pistes est donc :

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

Où m est le nombre de fourmis et ρ le taux d'évaporation. La quantité initiale de phéromones sur les arêtes est une distribution uniforme d'une petite quantité $\tau_0 \geq 0$.

La figure 1.3 présente un exemple simplifié de problème du voyageur de commerce optimisé par un algorithme AS dont le pseudo-code est présenté sur l'algorithme 1.1.

1.3.2.2 Variantes

Ant System et élitisme : Une première variante du 'Système de fourmi' a été proposée par [Dorigo et al., 1996] : elle est caractérisée par l'introduction de fourmis 'élitistes'. Dans cette version, la meilleure fourmi (celle qui a effectué le trajet le plus court) dépose une quantité de phéromones plus grande, dans l'optique d'accroître la probabilité des autres fourmis d'explorer la solution la plus prometteuse.

Algorithme 1.1 : Algorithme de colonie de fourmis de base :

Le Ant System

Pour tout $t=1, \dots, t_{max}$

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

 Choisir une ville au hasard

 Pour chaque ville non visitée i

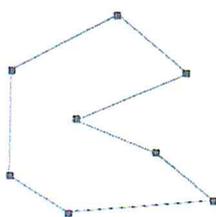
 Choisir une ville j , dans la liste J_i^k des villes restantes, selon la formule

1.1.

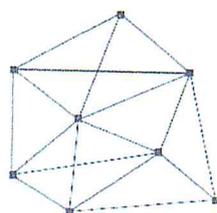
 Fin pour

 Evaporer les pistes selon la formule 1.3

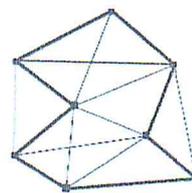
Fin pour



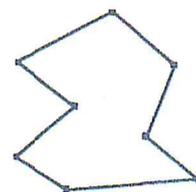
(a)



(b)



(c)



(d)

Chapitre 1

Où τ_0 est la valeur initiale de la piste . A chaque passage , les arêtes visitées voient leur quantité de phéromone diminuer , ce qui favorise la diversification par la prise en compte des trajets non explorés . A chaque itération , la mise à jour globale s'effectue comme ceci :

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

Où les arêtes (i, j) appartiennent au meilleur tour T^+ de longueur L^+ et où $\Delta\tau_{ij}(t) = \frac{1}{L^+}$. Ici , seule la meilleure piste est donc mise à jour, ce qui participe à une intensification par sélection de la meilleure solution.

3 Le système utilise une liste de candidats . Cette liste stocke pour chaque ville les v plus proches voisines , classées par distances croissantes . Une fourmi ne prendra en compte une arête vers une ville en dehors de la liste que si celle-ci a déjà été explorée.

Concrètement, si toutes les arêtes ont déjà été visitées dans la liste de candidats, le choix se fera en fonction de la règle 1.4 , sinon, c'est la plus proche des villes non visitées qui sera choisie.

ACS 3-opt : Cette variante est une hybridation entre le ACS et une recherche locale de type 3-opt [Dorigo and Gambardella, 1997 b] . Ici, la recherche locale est lancée pour améliorer les solutions trouvées par les fourmis (et donc les ramener à l'optimum local le plus proche).

Max -Min Ant System : Cette variante (notée MMAS) est fondée sur l'algorithme AS et présente quelques différences notables [Stutzle and Hoos , 1997 , Stutzle and Hoos , 2000] :

- 1 .Seule la meilleure fourmi met à jour une piste de phéromones
- 2 Les valeurs des pistes sont bornées par τ_{min} et τ_{max} .
- 3 Les pistes sont initialisées à la valeur maximum τ_{max} .
- 4 La mise à jour des pistes se fait de façon proportionnelle , les pistes les plus fortes étant moins renforcées que les plus faibles.
- 5 Une réinitialisation des pistes peut être effectuée .

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

1.3.2.3 Choix des paramètres

Pour l'algorithme AS, les auteurs préconisent que , bien que la valeur de Q ait peu d'influence sur le résultat final , cette valeur soit du même ordre de grandeur qu'une

estimation de la longueur du meilleur trajet trouvé. D'autre part, la ville de départ de chaque fourmi est typiquement choisie par un tirage aléatoire uniforme, aucune influence significative du placement de départ n'ayant pu être démontrée.

En ce qui concerne l'algorithme ACS, les auteurs conseillent d'utiliser $\tau_0 = (n \cdot L_{nn})^{-1}$, où n est le nombre de villes et L_{nn} la longueur d'un tour trouvé par la méthode du plus proche voisin.

Le nombre de fourmis m est un paramètre important; les auteurs suggèrent d'utiliser autant de fourmis que de villes (i.e. $m=n$) pour de bonnes performances sur le problème du voyageur de commerce. Il est possible de n'utiliser qu'une seule fourmi, mais l'effet d'amplification des longueurs différentes est alors perdu, de même que le parallélisme naturel de l'algorithme, ce qui peut s'avérer néfaste pour certains problèmes. En règle générale, les algorithmes de colonies de fourmis semblent assez peu sensibles à un réglage fin du nombre de fourmis.

1.3.3 Autres problèmes combinatoire

Les algorithmes de colonies de fourmis sont très étudiés depuis quelques années et il serait trop long de faire ici une liste exhaustive de toutes les applications et variantes qui ont été produites, même en se restreignant au domaine de l'optimisation. Dans les deux principaux champs d'application (problèmes NP – difficiles et problèmes dynamique), certains algorithmes ont cependant donné de très bons résultats. On peut notamment retenir des performances particulièrement intéressantes dans le cas de l'affectation quadratique [Stutzle and Hoos, 2000], de problèmes de planification [Merkle et al., 2000], de l'ordonnancement séquentiel [Gambardella and Dorigo 2000] du routage de véhicule [Gambardella et al. 1999], ou du routage sur réseau [Di Caro and Dorigo, 1998 b]. Il existe une littérature importante sur toutes sortes de problèmes : voyageur de commerce, coloriage de graphe, affectation de fréquence, affectation généralisée, sac à dos multidimensionnel, satisfaction de contraintes, etc.

1.3.4 Formalisation et propriétés d'un algorithme de colonie de fourmis

Une description élégante a été proposée [Dorigo and Stutzle, 2003], qui s'applique aux problèmes (combinatoires) où une construction partielle de la solution est possible. Ce cas, bien que restrictif, permet de dégager les apports originaux de ces méta heuristiques (dénommées ACO, pour 'Ant Colony Optimization', par les auteurs). Nous en avons choisi un extrait :

Une méta heuristique de colonie de fourmis est un processus stochastique construisant une solution , en ajoutant des composants aux solutions partielles .

Ce processus prend en compte (i) une heuristique sur l'instance du problème (ii) des pistes de phéromones changeant dynamiquement pour refléter l'expérience acquise par les agents.

Une formalisation plus précise existe [Dorigo and Stutzle , 2003] . Elle passe par une représentation du problème , un comportement de base des fourmis et une organisation générale de la méta heuristique . Plusieurs concepts sont également à mettre en valeur pour comprendre les principes de ces algorithmes , notamment la définition des pistes de phéromones ne tant que *mémoire adaptative* , la nécessité d'un réglage intensification/diversification et enfin l'utilisation d'une *recherche locale* . Nous traitons ci-après ces différents sujets.

1.3.4.1 Formalisation

Représentation du problème : Le problème est représenté par un jeu de solutions , une fonction objectif assignant une valeur à chaque solution et un jeu de contraintes . L'objectif est de trouver l'optimum global de la fonction objectif satisfaisant les contraintes . Les différents états du problème sont caractérisés comme une séquence de composants. On peut noter que , dans certains cas, un coût peut être associé à des états autres que des solutions .



FIG. 1.4 – Dans un algorithme de colonie de fourmis, les pistes de phéromone peuvent être associées aux composants (a) ou aux connexions (b) du graphe représentant le problème à résoudre.

Dans cette représentation , les fourmis construisent des solutions en se déplaçant sur un graphe $G=(C ,L)$, où les nœuds sont des composants de C et où l'ensemble L connecte les composants de C . Les contraintes de problème sont implémentées directement dans les règles de déplacement des fourmis (soit en empêchant les mouvements qui violent les contraintes , soit en pénalisant de telles solutions) .

Chapitre 1

Comportement des fourmis ; Les fourmis artificielles peuvent être caractérisée comme une procédure stochastique construisant des solutions sur le graphe $G=(C,L)$. En général, les fourmis tentent d'élaborer des solutions faisables mais, si nécessaire, elles peuvent produire des solutions infaisables. Les composants et les connexions peuvent être associées à des pistes de phéromones τ (mettant en place une mémoire adaptative décrivant l'état du système) et à une valeur heuristique η (représentant une information a priori sur le problème, ou venant d'une source autre que celle des fourmis ; c'est bien souvent le coût de l'état en cours) Les pistes de phéromones et la valeur de l'heuristique peuvent être associées soit aux composants, soit aux connexions (figure 1.4).

Chaque fourmi dispose d'une mémoire utilisée pour stocker le trajet effectué, d'un état initial et de conditions d'arrêt. Les se déplaçant d'après une règle de décision probabiliste fonction des pistes de phéromone locales, de l'état de la fourmi et des contraintes du problème. Lors de l'ajout d'un composant à la solution en cours, les fourmis peuvent mettre à jour la piste associée au composant ou à la connexion correspondante. Une fois la solution construite, elles peuvent mettre à jour la piste de phéromone des composants ou des connexions utilisées. Enfin, une fourmi dispose au minimum de la capacité de construire une solution du problème.

En plus des règles régissant le comportement des fourmis, un autre processus majeur a cours : **l'évaporation des pistes de phéromones**.

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

Selon les auteurs du formalisme ACO, il est possible d'implémenter d'autres processus nécessitant un contrôle centralisé (et donc ne pouvant être directement pris en charge par des fourmis), sous la forme de processus annexes. Ce n'est, à notre sens, que peu souhaitable ; en effet, on perd alors la caractéristique décentralisé du système. De plus, l'implémentation de processus annexes entre difficilement dans une formalisation rigoureuse.

1.3.4.2 Phéromones et mémoire

L'utilisation de la *stigmergie* est cruciale pour les algorithmes de colonies de fourmis.

Chapitre 1

Le choix de la méthode d'implémentation des pistes de phéromones est donc important pour obtenir les meilleurs résultats. Ce choix est en grande partie lié aux possibilités de représentation de l'espace de recherche, chaque représentation pouvant apporter une façon différente d'implémenter les pistes. Par exemple, pour le problème du voyageur de commerce, une implémentation efficace consiste à utiliser une piste τ_{ij} entre deux villes i et j comme une représentation de l'intérêt de visiter la ville j après la ville i . Une autre représentation possible moins efficace en pratique consiste à considérer τ_{ij} comme une représentation de l'intérêt de visiter i en tant que même ville.

En effet, les pistes de phéromone décrivent à chaque pas l'état de la recherche de la solution par le système, les agents modifient la façon dont le problème va être représenté perçu par les autres agents. Cette information est partagée par le biais des modifications de l'environnement des fourmis, grâce à une forme de communication indirecte : la stigmergie.

L'information est donc stockée un certain temps dans le système, ce qui a amené certains auteurs à considérer ce processus comme une forme de mémoire adaptative [Taillard, 1998, Taillard et al., 1998], où la dynamique de stockage et de partage de l'information va être cruciale pour le système.

1.3.4.3 intensification / diversification

Le problème de l'emploi relatif de processus de diversification et d'intensification est un problème extrêmement courant dans la conception et l'utilisation de méta heuristiques. Par intensification, on entend l'exploitation de l'information accumulée par le système à un moment donné. La diversification est au contraire l'exploration de régions de l'espace de recherche imparfaitement prises en compte. Bien souvent, il va s'agir de choisir où et quand 'injecter de l'aléatoire' dans le système (diversification) et / ou améliorer une solution (intensification).

Dans les algorithmes de type ACO, comme dans la plupart des cas, il existe plusieurs façons de gérer l'emploi de ces deux facettes des méta heuristiques d'optimisation. La plus évidente passe par le réglage via les deux paramètres α et β , qui déterminent l'influence relative des pistes de phéromone et de l'information heuristique. Plus la valeur de α sera grande, plus l'intensification sera importante, car plus les pistes auront une influence sur le choix des fourmis. A l'inverse, plus α sera faible, plus la diversification sera forte, car les fourmis

éviteront les pistes . Le paramètre β agit de façon similaire . On doit donc gérer conjointement les deux paramètres , pour régler ces aspects.

On peut également introduire des modifications de la gestion des pistes de phéromone . Par exemple , l'emploi de stratégies élitistes (les meilleurs solutions contribuent plus aux pistes) favorise l'intensification , alors qu'une réinitialisation de l'ensemble des pistes favorisera l'exploration .

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

Dans ces algorithmes fondés en grande partie sur l'utilisation de l'auto-organisation , ces deux approches peuvent être équivalentes, un changement de paramètre pouvant induire un comportement complètement différent du système , au niveau global.

1.3.4.4 Recherche locale et heuristiques

Les méta heuristiques de colonies de fourmis sont souvent plus efficaces quand elles sont hybridées avec les algorithmes de recherche locale . Ceux – ci optimisent les solutions trouvées par les fourmis , avant que celles- ci ne soient utilisées pour la mise à jour des pistes de phéromone .Du point de vue de la recherche locale , utiliser des algorithmes de colonies de fourmis pour engendrer une solution initiale est un avantage indéniable. Ce qui différencie une méta heuristique de type ACO intéressante d'un algorithme réellement efficace est bien souvent l'hybridation avec une recherche locale.

Une autre possibilité pour améliorer les performances est d'injecter une information heuristique plus pertinente . Cet ajout a généralement un coût élevé en terme de calculs supplémentaires .

Il faut bien noter que ces deux approches sont similaires de par l'emploi qu'elles font des informations de coût pour améliorer une solution . La recherche locale le fait de façon plus directe que l'heuristique , cependant que cette dernière est peut-être plus naturelle pour utiliser des informations a priori sur le problème.

1.3.4.5 Parallélisme

La structure même des méta heuristiques de colonies de fourmis comporte un parallélisme intrinsèque . D'une manière générale , les solutions de bonne qualité émergent du résultats des interactions indirectes ayant cours dans le système , pas d'un

codage explicite d'échanges . En effet, chaque fourmi ne prend en compte que des informations locales de son environnement (les pistes de phéromone) ; il est donc facile de paralléliser un tel algorithme .(i.e. le comportement des fourmis , l'évaporation et les processus annexes) peuvent également être implémentés de manière indépendante , l'utilisateur étant libre de décider de la manière dont ils vont interagir.

1.3.4.6 Convergence

Les méta heuristique peuvent être vues comme des modifications d'un algorithme de base : une recherche aléatoire .

Cet algorithme possède l'intéressante propriété de garantir que la solution optimale sera trouvée tôt ou tard , on parle alors de convergence . Cependant puisque cet algorithme de base est biaisé , la garantie de convergence n'existe plus .

Si , dans certains cas, on peut facilement être certain de la convergence d'un algorithme de colonies de fourmis (MMAS), le problème reste entier en ce qui concerne la convergence d'un algorithme ACO quelconque . Cependant , il existe une variante dont la convergence a été prouvée (Gutjahr,2000,Gutjahr, 2002] : le 'Graph-Based Ant System'(GBAS) . La différence entre GBAS et l'algorithme AS se situe au niveau de la mise à jour des pistes de phéromone , qui n'est permise que si une meilleure solution est trouvée . Pour certaines valeurs de paramètres , et étant donné $\epsilon > 0$ une 'faible ' valeur , l'algorithme trouvera la solution optimale avec une probabilité $P_t \geq 1 - \epsilon$, après un temps $t \geq t_0$ (où t_0 est fonction de ϵ).

1.3.5 Extensions

Devant le succès rencontré par les algorithmes de colonies de fourmis , de nombreuses pistes autres que celle de l'optimisation combinatoire commencent à être explorées ; par exemple , l'utilisation de ces algorithmes dans des problèmes continus et / ou dynamiques , ou encore la mise en relation de ce type d'algorithme dans un cadre d'intelligence en essaim et avec d'autres méta heuristiques .

1.3.5.1 Optimisation continue

Les Meta heuristiques sont bien souvent élaborées pour des problèmes combinatoires , mais il existe une classe de problèmes souvent rencontrée en ingénierie , où la fonction objectif est à variables continues et pour lesquels les méta heuristique peuvent être d'un grand secours (fonction non dérivable , multiples minimums locaux , grand nombre de variables , non-convexité , etc) .Plusieurs

Chapitre 1

tentatives pour adapter les méta heuristiques de colonies de fourmis au domaine continu sont apparues .

Outre les problèmes classiques d'adaptation d'une méta heuristique , les algorithmes de colonies de fourmis posent un certain nombre de problèmes spécifiques .Ainsi , le principal problème vient si l'on se place dans le formalisme ACO avec une construction de la solution composante par composant . En effet, un problème continu peut selon la perspective choisie présenter une infinité de composants , le problème de la construction est difficilement soluble dans ce cas . La plupart des algorithmes s'inspirent donc des caractéristiques d'auto-organisation et de mémoire externe des colonies de fourmis , laissant de côté la construction itérative de la solution ; cependant , le caractère probabiliste du formalisme ACO commence à être employé .

Plusieurs algorithmes de colonies de fourmis pour l'optimisation continue : CACO , un algorithme hybride non baptisé, API , ACO pour l'optimisation continue et CACS.

L'algorithme CACO : Le premier de ces algorithmes nommé CACO (Continuous Ant Colony Algorithm) , utilise deux approches : un algorithme de type évolutionnaire sélectionne et croise des régions d'intérêt, que des fourmis explorent et évaluent .

Une fourmi sélectionne une région avec une probabilité proportionnelle à la concentration en phéromone de cette région , de la même manière que dans le 'Ant System ' , une fourmi sélectionnerait une piste allant d'une ville à une autre :

$$p_i(t) = \frac{\tau_i^\alpha(t) n_i^\beta(t)}{\sum_{j=1}^N \tau_j^\alpha(t) n_j^\beta(t)}$$

Où N est le nombre de régions et $n_i^\beta(t)$ est utilisé pour inclure une heuristique spécifique au problème . Les fourmis partent alors du centre de la région et se déplacent selon une direction choisie aléatoirement , tant qu'une amélioration de la fonction objectif est trouvée. Le pas de déplacement utilisé par la fourmi entre chaque évaluation est donnée par :

$$\delta_r(t, R) = R \cdot (1 - u^{((1 - t/T)^c)}$$

Où R est le diamètre de la région explorée, $u \in [0,1]$ un nombre aléatoire, T le nombre total d'itérations de l'algorithme et c un paramètre de refroidissement (permettant de réduire le pas à chaque itération). Si la fourmi a trouvé une meilleure solution, la région est déplacée de façon à ce que son centre coïncide avec cette solution, et la fourmi augmente la quantité de phéromone de la région proportionnellement à l'amélioration trouvée (appelée ici 'fitness'). L'évaporation des ' pistes' se fait classiquement en fonction d'un coefficient.

Des modifications ont été apportées par Wodrich et al pour améliorer les performances de l'algorithme original. Ainsi, en plus des fourmis 'locales' de CACO, des fourmis 'globales' vont explorer l'espace de recherche (figure 1.5) pour éventuellement remplacer les régions peu intéressantes par de nouvelles régions non explorées. Les régions sont également affectées d'un âge, qui augmente si aucune amélioration n'est découverte. De plus, le paramètre t dans le pas de recherche des fourmis $\delta r(t, R)$ est défini par l'âge de la région explorée.

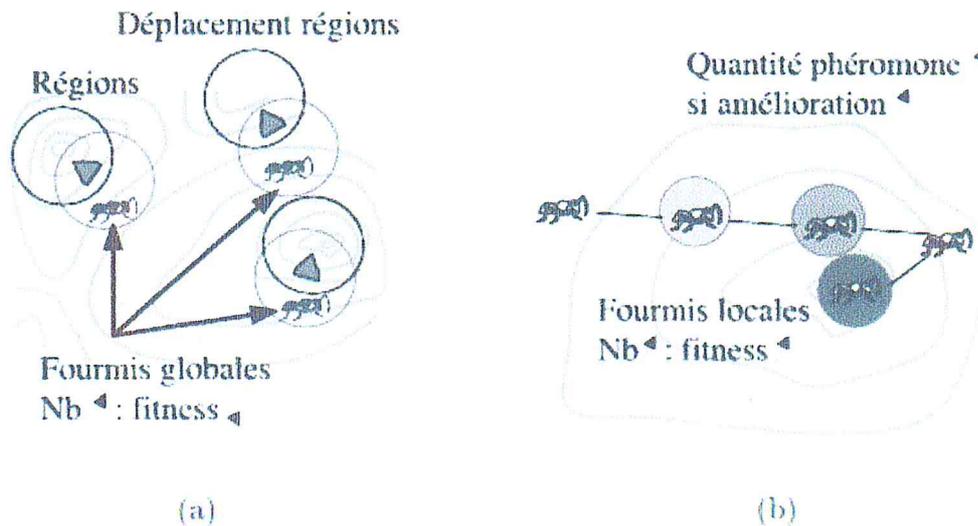


FIG. 1.5 – L'algorithme CACO : les fourmis globales (a) participent au déplacement des régions que les fourmis locales (b) évaluent.

Chapitre 1

Une refonte de l'algorithme [Mathur et al ., 2000] a été opérée par d'autres auteurs en vue de relier plus finement CACO avec le paradigme des colonies de fourmis et d'abandonner la liaison avec l'algorithme évolutionnaire . On parle ainsi par exemple de diffusion pour définir la création de nouvelles régions.

Une méthode hybride : Une approche semblable utilisant à la fois une approche de fourmis et d'algorithmes évolutionnaire a été proposée par Ling et al.[Ling et al . , 2002] .L'idée principale de cette méthode est de considérer les écarts entre deux individus sur chaque dimension comme autant de parties d'un chemin où les phéromones sont déposées , l'évolution des individus étant prise en charge par des opérateurs de mutation et de croisement . D'un certain point de vue , cette méthode tente donc de reproduire le mécanisme de construction de la solution , composant par composant .

La méthode procède précisément selon l'algorithme 1.2 . Chaque fourmi x_i de la population contenant m individus est considérée comme un candidat à l'élément $x_{i,e}^*$ de la solution optimale. L'idée est d'utiliser le chemin entre les éléments $x_{i,e}$ et $x_{j,e}$ noté (i, j) pour déposer une piste de phéromone dont la concentration est notée $\tau_{ij}(t)$ au pas de temps t .

Les auteurs ont proposé une version 'adaptative', où les probabilités de mutation et de croisement sont variables. Malheureusement , cet algorithme n'a pas encore été complètement testée.

L'algorithme API : Dans tous les algorithmes évoqués jusqu'ici , le terme 'colonies de fourmis ' s'entend de par l'utilisation de la stigmergie comme processus d'échange d'information .

Il existe cependant un algorithme adapté au cas continu qui s'inspire du comportement de fourmis primitives (ce qui ne veut pas dire inadaptées) de l'espèce *Pachycondyla apicalis*, et qui ne fait pas usage de la communication indirecte par piste de phéromones : l'algorithme API .

Dans cette méthode , on commence par positionner un nid aléatoirement sur l'espace de recherche , puis des fourmis sont distribuées aléatoirement dans l'espace . Ces fourmis vont alors explorer localement leur 'site de chasse ' en évaluant plusieurs points dans un périmètre donné (voir figure 1.6) . Chaque fourmi mémorise le meilleur point trouvé. Si , lors de l'exploration de son site de chasse , elle trouve un meilleur point , alors elle reviendra sur ce site , sinon, après un certain nombre

Chapitre 1

d'explorations, elle choisira un autre site. Une fois les explorations des sites de chasse terminées, des fourmis tirées au hasard comparent deux à deux (comme peuvent le faire les fourmis réelles durant le comportement de 'tandem running' ou course en couple, leurs meilleurs résultats, puis mémorisent le meilleur des deux sites de chasse. Le nid est finalement réinitialisé sur le meilleur point trouvé après un temps donné.

Algorithme 1.2 Un algorithme de colonies de fourmis hybride pour le cas continu

1. A chaque itération, sélectionner pour chaque fourmi une valeur initiale dans le groupe de valeurs candidates avec la probabilité :

$$p_{ij}^k(t) = \frac{\tau_{ij}(t)}{\sum \tau_{ir}(t)}$$

2. Utiliser des opérateurs de mutation et de croisement sur les m valeurs, afin d'obtenir m nouvelles valeurs ;
3. Ajouter ces nouvelles valeurs au groupe de valeurs candidates pour le composant $x_{i,e}$;
4. Les utiliser pour former m solutions de la nouvelle génération ;
5. Calculer la 'fitness' de ces solutions ;
6. Quand m fourmis ont parcouru toutes les arêtes, mettre à jour les pistes de phéromone des valeurs candidates de chaque composant par :

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \sum \tau_{ij}^k$$

7. Si la $k^{\text{ème}}$ fourmi choisit la $j^{\text{ème}}$ valeur candidate du groupe de composants, alors $\Delta\tau_{ij}^k(t+1) = Wf_k$, sinon $\Delta\tau_{ij}^k=0$, en désignant par W une constante et par f_k la 'fitness' de la solution trouvée par la $k^{\text{ème}}$ fourmi ;
8. Effacer les m valeurs ayant les plus basses intensités de phéromone dans chaque groupe de candidats .

La mémoire des sites des fourmis est remise à zéro, et l'algorithme effectue une nouvelle itération.

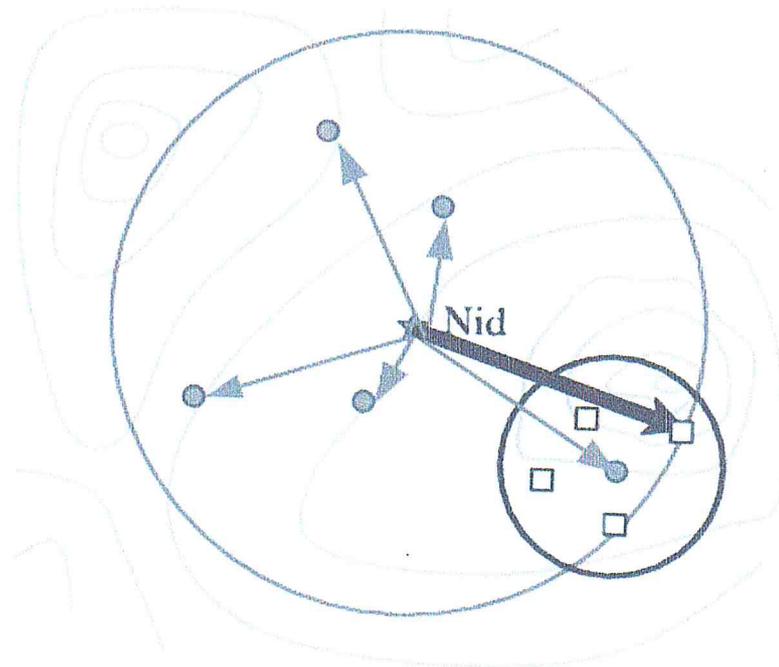


FIG. 1.6 – L’algorithme API : une méthode à démarrage multiple inspirée par une espèce de fourmi primitive. Les fourmis (cercles pleins) explorent des sites de chasse (petits carrés) dans un périmètre (grand cercle) autour du nid. Le nid est déplacé sur le meilleur point au moment de la ré-initialisation (flèche en trait gras).

ACO pour l’optimisation continue : Cet algorithme [Socha , 2004] tente de maintenir la construction itérative des solutions dans le cas de variables continues , en adoptant un point de vue différent des précédents .En effet, il prend le parti de considérer que les composants de toutes les solutions sont formés par les différentes variables optimisées . De plus , plutôt que de considérer l’algorithme du point de vue de l fourmi , il prend le parti de se placer au niveau de la colonie , si les fourmis n’étant plus que des points à évaluer .

Dans cette méthode, dénommée “ACO pour l’optimisation continue “, on tire aléatoirement une population de fourmis dans une distribution de probabilité à chaque itération . De cet ensemble de points ne sont conservés que les meilleurs points , qui servent alors à construire une “meilleure “distribution de probabilité .

La distribution de probabilité utilisée est un “amalgame pondéré de noyaux normaux”, soit un ensemble de distribution normales combinées :

$$P(x)=\sum_{j=1}^K w_j \cdot \left(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \right)$$

Chapitre 1

en désignant par k le nombre de noyaux utilisés, μ et σ^2 la moyenne et la variance d'un noyau et w_i la pondération.

Chaque distribution est nommée " mise à jour de phéromone " et consiste à

Algorithme 1.3 Algorithme du type colonies de fourmis, utilisant l'estimation de distribution pour l'optimisation en variables continues.

Construire la distribution de probabilité initiale : $\tau_i^0 = p_i^0(x_i)$, $i \in \{1 \dots n\}$

Tant que (critère d'arrêt non atteint) ;

 Pour chaque fourmi, de $a = 1$ à m ;

 Pour chaque variable, de $i = 1$ à n ;

 Choisir aléatoirement une valeur x_i selon la distribution $P_i(x_i)$;

 Ajouter à la solution en construction :

$$S^a = \{s_1^a, \dots, s_{i-1}^a\} \cup \{x_i\}$$

 Fin

Fin

Mémoriser les k meilleures solutions : $S^* = \{S_1^*, \dots, S_k^*\}$

Reconstruire la distribution de probabilité selon les meilleures solutions : $\tau = P(S^*)$

Fin

Renforcer ou diminuer l'influence des noyaux correspondant aux solutions. Le principe de la méthode est présenté sur l'algorithme 1.3.

L'algorithme CACS : Cette méthode appelée " Continuous Ant Colony System " [Pourtakdoust and Nobahari, 2004] est très proche de la précédente, bien qu'ayant été présentée simultanément. En effet, dans CACS comme dans ACO pour l'optimisation continue, le cœur de l'algorithme consiste à faire évoluer une distribution de probabilité.

Chapitre 1

Le principe de la méthode est le même que celui présenté sur l'algorithme 1.3. Dans CACS, la distribution utilisée est dite "normale", mais la formule de la distribution diffère légèrement de la formule classique (équation 1.5) et la variance utilisée est en fait un nouvel indice de dispersion (voir équation 1.6).

$$P(x) = e^{-\frac{(x-x_{min})^2}{2\sigma^2}} \quad (1.5)$$

en désignant par x_{min} le mode de la distribution et σ^2 l'indice de dispersion suivant :

$$\sigma^2 = \frac{\sum_{j=1}^m \frac{1}{f_j - f_{min}} (x_j - x_{min})^2}{\sum_{j=1}^m \frac{1}{f_j - f_{min}}} \quad (1.6)$$

En désignant par m le nombre de fourmis, f_j la valeur de la fonction associée à la fourmi j , et f_{min} la meilleure valeur trouvée.

Dans CACS, la seule distribution utilisée est centrée sur le mode de la distribution de l'itération précédente, et non sur la moyenne.

En résumé dans le cas continu : On a pu voir que deux types d'algorithmes sur cinq étaient en fait plus ou moins hybridés avec un algorithmes de type évolutionnaire, et qu'un troisième n'utilisait pas la métaphore "classique" des colonies de fourmis. Les deux derniers algorithmes utilisent une approche différente des précédents, plus centrée sur l'aspect probabiliste que comportemental des algorithmes à colonies de fourmis. D'une manière générale, la recherche en est encore à ses débuts et les algorithmes produits n'ont pas atteint leur maturité, et ne sont donc pas encore vraiment compétitifs par rapport à d'autres classes de méta heuristiques plus élaborées sur les problèmes continus.

1.3.5.2 Problèmes dynamiques

Un problème est dit dynamique s'il varie dans le temps, i.e. si la solution optimale ne possède pas les mêmes caractéristiques durant le temps de l'optimisation. Ces problèmes soulèvent des difficultés spécifique, du fait qu'il faut approcher au mieux la meilleure solution à chaque pas de temps.

La première application des algorithmes de colonies de fourmis à des problèmes dynamiques concernait l'optimisation du routage sur des réseaux de type téléphonique [Schoonderwoerd et al., 1996]. Cependant, l'algorithme proposé n'ayant pas fait l'objet d'études et de comparaisons approfondies, il est difficile d'en tirer des enseignements.

Une autre application a été exposée par White et al [White et al., 1998, Bieszczad and White, 1990] sur des problèmes similaires. Une application a été décrite pour des problèmes de

Chapitre 1

routage sur des réseaux internet (figure 1.7) : l'algorithme AntNet [Di Caro and Dorigo , 1997] . Cette méta heuristique a fait l'objet de plusieurs études et semble avoir prouvé son efficacité sur plusieurs problèmes tests.

Chacun de ces algorithmes utilise pour mettre à jour des tables de routage probabiliste des fourmis pour explorer le réseau de façon à ce que l'information pertinente soit la fréquence de passage des fourmis sur chaque nœud . D'une manière générale , l'aspect distribué et flexible des algorithmes de colonies de fourmis semble très bien adapté à des problèmes dynamiques .

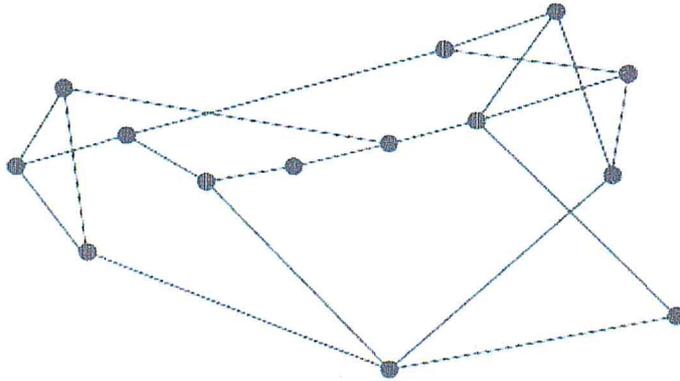


FIG. 1.7 – Exemple de réseau utilisé pour tester l'algorithme AntNet : NFSNET (chaque arête représente une liaison orientée).

Chacun de ces algorithmes utilise pour mettre à jour des tables de routage probabilistes des fourmis pour explorer le réseau de façon à ce que l'information pertinente soit la fréquence de passage des fourmis sur chaque nœud. d'une manière générale, l'aspect distribué et flexible des algorithmes de colonies de fourmis semble très bien adapté a des problèmes dynamiques.

Chapitre 2

Introduction

POSITION DU PROBLEME

Cette note a pour but de montrer comment sur le plan informatique, on peut intégrer au mieux les équations du mouvement d'un solide autour de son axe d'inertie (satellite en particulier). c'est ce qu'on appelle les équations d'attitude qui sont particulièrement utiles dans la mise en place d'un SCAO (système de contrôle d'attitude et d'orbite) notamment dans le cas où l'orientation peut être quelconque.

Le repérage de l'attitude d'un corps dans l'espace est donnée par les angles ;

D'Euler \rightarrow précession ψ , nutation θ , rotation propre φ .

De cardan \rightarrow lacet ψ , tangage Θ et roulis φ

Ces deux repérages possèdent des indéterminations ou des non définitions des angles dans certaines configurations. de plus la mise sous forme canonique des équations du mouvements s'avère très difficile voire impossible. il se pose donc un problème sérieux, sur le plan numérique dans l'étude d'un mouvement général.

Généralement les angles d'Euler ou équivalents sont utilisés dans les équations aux petites variations linéarisées, autour des positions d'équilibre ou les angles sont toujours parfaitement définis (ce n'est pas le cas de $\theta=0$ en particulier pour les angles d'Euler) .

Les développements qui suivent présentent une théorie mathématique qui permet de s'affranchir des configurations angulaires singulières.

L'attitude d'un satellite, c'est-à-dire son orientation, est équivalente à la connaissance d'un repère R qui lui est lié, rapportée à un repère de référence R_0 inertiel ou pas .

Or, il existe toujours une rotation qui permet de transformer R_0 en R . cette rotation peut être caractérisée de nombreuses manières :

3 scalaires, des angles par exemple de Cardan ou d'Euler, avec des singularités de définitions dans certaines configurations.

Une matrice de rotation P ou de passage avec 9 scalaires reliés par 6 relations (trois vecteurs colonnes unitaires et trois vecteurs lignes unitaires , de plus $P^t = P^{-1}$)

La caractérisation géométriques de la rotation par un axe unitaire et un angle , soit 4 scalaires reliés par une relation (axe unitaire).

Un être nouveau que nous allons définir par 4 scalaires avec une relation de norme 1 imposée, le quaternion, objet de cette étude.

Dans tous les cas, il y'a trois inconnues.

INTRODUCTION DES QUATERNIONS

Rappel sur la rotation instantanée

La figure ci-dessous montre :

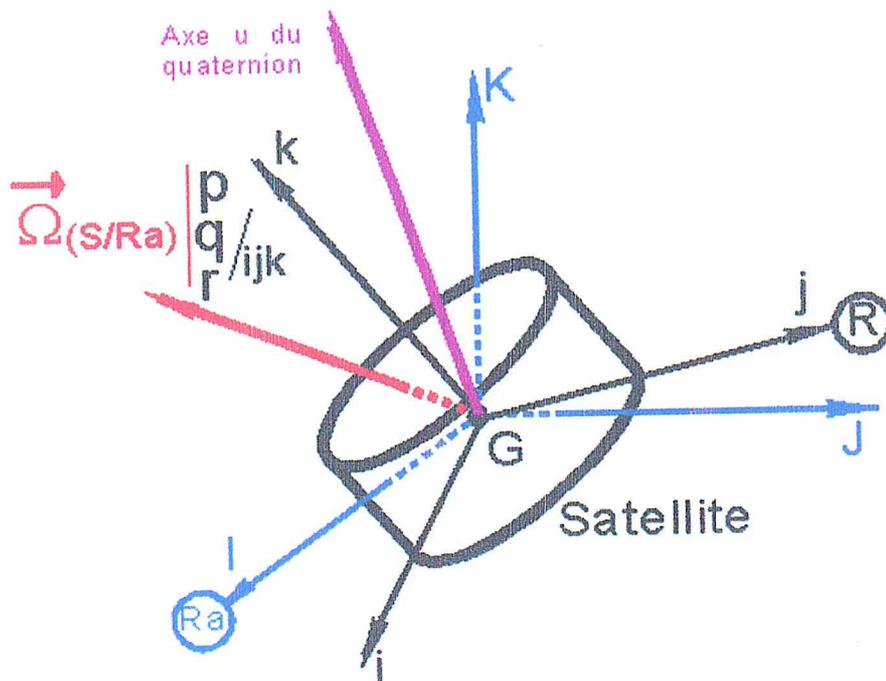
La base X Y Z de référence qui est en général inertielle mais ce peut être le repère orbital.

La base mobile x y z en général liée au solide S en mouvement.

Le vecteur rotation instantanée Ω du solide par rapport au repère inertielle. Dans le cas où la base mobile est liée au corps, c'est la rotation de cette base par rapport a la base absolue.

On note classiquement $\Omega = (p, q, r)$ la matrice des composantes de $\Omega(S/Ra)$ sur la base mobile x y z

$$\vec{\Omega} = p\vec{x} + q\vec{y} + r\vec{z}$$



Introduction des quaternions et de l'algèbre des quaternions

Rotation et quaternion élémentaire

Géométriquement il est clair que quel que soit la configuration des bases X Y Z et x y z, il existe deux rotations (au sens géométrique du terme) qui permettent de passer de la base X Y Z a la base x y z

Chapitre 2

Rotation d'axe le vecteur $\vec{u}=(\alpha,\beta,\gamma)$ et d'angle noté θ ($0 < \theta < 2\pi$)

Rotation d'axe le vecteur $-\vec{u} = -(\alpha, \beta, \gamma)$ et d'angle $2\pi-\theta$ ($0 < 2\pi - \theta < 2\pi$)

L'idée a donc été de créer un être mathématique nouveau, a quatre composantes Q, appelé quaternions représentant cette transformation géométrique, a ne pas confondre avec la rotation au sens de la mécanique qui est un vecteur matérialisant la vitesse angulaire et l'axe de rotation.

$$Q = \left(\cos \frac{\theta}{2}, u \sin \frac{\theta}{2} \right) = \left(\cos \frac{\theta}{2}, \alpha \sin \frac{\theta}{2}, \beta \sin \frac{\theta}{2}, \gamma \sin \frac{\theta}{2} \right)$$

Ou $u(\alpha,\beta,\gamma)$ est l'unitaire de l'axe de la rotation et θ l'angle de rotation.

Définition de l'algèbre des quaternions

L'algèbre des quaternions est le sous espace vectoriel H de l'ensemble des matrices carrées d'ordre 4 engendré par la base {e, i, j, k}

$$e = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \quad j = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad k = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}$$

on peut encore noter ces quatre éléments particuliers

$$e=(1000), i=(0100), j=(0010), k=(0001)$$

un quaternion est donc noté sous forme complexe, dans la base canonique le (e,i,j,k).

$$Q = q_0 e + q_1 i + q_2 j + q_3 k$$

ou les composantes de Q sont réelles, e est l'élément neutre de la multiplication .

la première composante $q_0 e$ est appelée partie réelle

la deuxième composante $q_1 i + q_2 j + q_3 k$ a trois termes est appelée partie imaginaire ou partie pure.

Une partie imaginaire est assimilable a un vecteur.

Un quaternion pur a une partie réelle nulle.

L'ensemble P des quaternions purs est un sous- espace vectoriel de dimension 3. Tout quaternion se décompose de manière unique en sa partie réelle et sa partie pure.

Cet ensemble P isomorphe à R^3 est assimilable a l'ensemble des vecteurs de R^3 sur lequel il est aisé de construire un espace affine de la mécanique.

Operations dans l'algèbre des quaternions :

Conjugaison

Le conjugué de Q est noté \bar{Q}

$$Q = q_0 e + q_1 i + q_2 j + q_3 k \quad \bar{Q} = q_0 e - q_1 i - q_2 j - q_3 k$$

Relations remarquables

$$\text{conjugué}(Q.Q') = \text{conjugué}(Q') \cdot \text{conjugué}(Q)$$

$$Q \text{ quaternion pur} \Leftrightarrow \bar{Q} = -Q$$

Structure d'algèbre : on définit sur l'ensemble H des quaternions des opérations classiques

Chapitre 2

Une multiplication par un scalaire de définition classique.

Une opération d'addition associative et commutative de définition évidente

Une multiplication associative mais non commutative en général, distributive par rapport à l'addition et satisfaisant aux règles de calcul suivantes :

$$\begin{aligned} i^2=j^2=k^2=-e, e^2=e \\ e.i=i.e=i, e.k=k.e=k, e.j=j.e=j \\ i.j=-j.i=k \\ j.k=-k.j=i \\ k.i=-i.k=j \end{aligned}$$

on notera sur les trois dernières relations l'analogie avec le produit vectoriel dans R^3 .

produit scalaire

$$\begin{aligned} \vec{Q} \cdot \vec{Q} &= \text{Re}(Q \cdot \bar{Q}') \\ \text{norme}(Q) &= |Q| = \sqrt{\text{Re}(Q \cdot \bar{Q})} \\ |Q \cdot Q'| &= |Q| |Q'| \\ \text{Si } Q \neq 0 \quad Q^{-1} &= \frac{\bar{Q}}{|Q|^2} \end{aligned}$$

Nb : le produit (multiplication dans l'algèbre) de deux quaternions purs (c'est-à-dire éléments de P) s'explique aussi à l'aide du produit scalaire (écriture à l'aide des vecteurs) des quaternions et du produit vectoriel par analogie avec celui de R^3 .

$$Q \cdot Q' = -(\vec{Q} \cdot \vec{Q}')e + \vec{Q} \wedge \vec{Q}'$$

Les opérations entre vecteurs sont celles classiques de R^3

NB ;

les vecteurs de base sont orthogonaux un à un et unitaires.

3°) ensembles S des quaternions de norme 1

Ce sont des quaternions qui nous intéressent plus particulièrement pour représenter les rotations en tant qu'isométries.

On appelle S le sous espace de H des quaternions de norme 1

$$Q \cdot Q' = -(\vec{Q} \cdot \vec{Q}')e + \vec{Q} \wedge \vec{Q}' \quad \text{équation de redondance}$$

$$Q \in S \Leftrightarrow \text{norme}(Q) = |Q| = \sqrt{\text{Re}(Q \cdot \bar{Q})} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1$$

Soit Q un élément de S posons

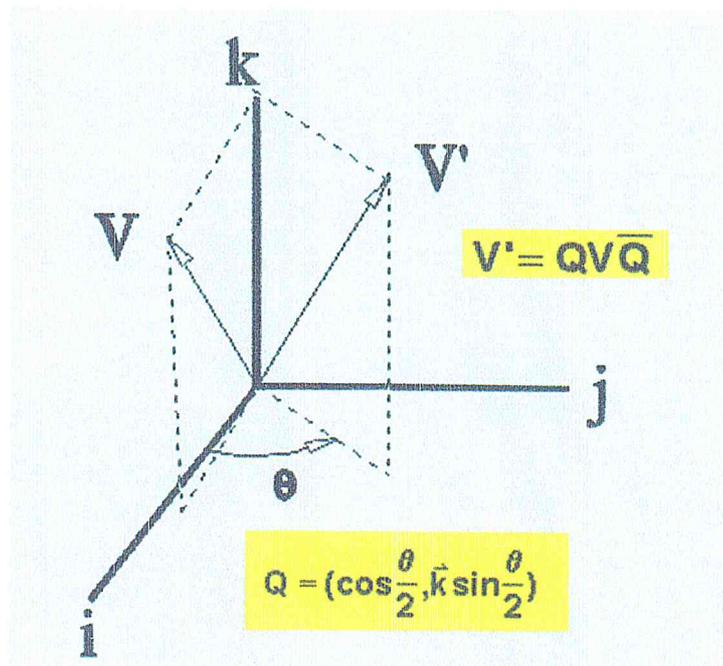
$$p \in P \Rightarrow f(p) \Rightarrow Qp\bar{Q}$$

On montre alors que l'application f est un endomorphisme orthogonal de P qui se décompose de manière unique sous la forme

$$Q = \cos \frac{\theta}{2} + u \sin \frac{\theta}{2}$$

Où u est un élément de S ou encore un unitaire.

Ci-dessous la rotation d'angle θ et d'axe k qui transforme v en v', nous aurons en termes de quaternions :



On établit donc une analogie simple (bijection) entre les éléments de S et les rotations de l'espace R^3 d'axe u et d'angle θ .

En termes plus rigoureux il y a homomorphie de groupe entre S et le groupe des rotations de R^3 . Autant dire à toute rotation géométrique on peut associer un quaternion Q et donc l'application linéaire R_Q qui la représente.

Dynamique d'un satellite et quaternions d'attitude

Dans le paragraphe précédent, nous avons associé à toute rotation géométrique (déplacement) un quaternion Q , d'un espace S muni d'une algèbre.

Les développements qui suivent se proposent de relier la rotation instantanée à la dérivée d'un quaternion.

Dérivation et rotation instantanée Ω

Nous désignons par I, J, K les quaternions de base de l'espace de référence absolu et ceux i, j, k d'une base R (liée au satellite, par exemple, mais ce n'est pas une nécessité). Nous rappelons que la rotation instantanée ne concerne que le mouvement de R par rapport à R_a .

Le quaternion Q représente la rotation géométrique transformant à l'instant t la base I, J, K en i, j, k . Ce quaternion, variable a priori, est donc fonction du temps t .

Dérivation temporelle d'un quaternion $Q(t)$

Cette dérivation est définie comme pour les vecteurs, par ses composantes dans i, j, k par :

Chapitre 2

$$\frac{dQ}{dt} = \dot{Q} = \dot{q}_0 e + \dot{q}_1 I + \dot{q}_2 J + \dot{q}_3 K$$

Nous aurions aussi pu la définir par ses composantes dans I,J,K. En pratique la dérivée d'un quaternion, tout comme le quaternion lui-même n'a pas de lien avec la base, puisqu'elle caractérise l'évolution d'une transformation géométrique intrinsèque. C'est une notion intrinsèque. On veillera à ne pas confondre avec la dérivée d'un vecteur qui, elle est associée à une base, tout comme la vitesse.

Remarque : $2\dot{Q}\bar{Q}$ est un quaternion pur en utilisant les règles opératoires sur les quaternions.

Introduction de la rotation instantanée Ω

Nous allons montrer sur un exemple simple, puis de manière générale que le quaternion :

$$2\dot{Q}.\bar{Q} = -2\bar{Q}\dot{Q} = \Omega$$

Purement imaginaire ou pur représente le vecteur rotation instantanée du repère R (ou du solide si la base i,j,k est attachée à ce solide), exprimée dans la base d'écriture du quaternion.

Exemple

Soit le quaternion représentant la rotation géométrique d'angle θ autour de l'axe fixe K.

$$Q = \cos\frac{\theta}{2}e + OI + OJ + \sin\frac{\theta}{2}K = (\cos\frac{\theta}{2}, \vec{K} \sin\frac{\theta}{2})$$

Et son dérivé, défini comme plus haut

$$\frac{dQ}{dt} = \frac{\dot{\theta}}{2} (-\sin\frac{\theta}{2}e + OI + OJ + \cos\frac{\theta}{2}K)$$

On démontre aisément que le vecteur rotation instantanée du mouvement absolu, exprimé dans la base (I,J,K) est :

$$\Omega = 2\dot{Q}\bar{Q} = \dot{\theta}K$$

Généralisation

La généralisation se fait en utilisant les quaternions $Q(t)$ et son conjugué, représentant les rotations R_Q et $R_{\bar{Q}}$ son inverse, variables, de R^3 , donc fonction du temps t.

$[i(t), j(t), k(t)]$ est la base mobile déduite de la base canonique de R^3 , $[I, J, K]$ inertielle, par la rotation associée à $Q(t)$. Nous notons Ω_s et Ω_l les quaternions représentant Ω dans chaque base.

Ainsi en exprimant i comme image de I, puis en le dérivant, naturellement dans la base inertielle, il vient :

$$\begin{aligned} i(t) &= QI\bar{Q} \\ \frac{di}{dt} &= \dot{Q}I\bar{Q} + QI\dot{\bar{Q}} \end{aligned}$$

Mais $2\dot{Q}\bar{Q}$ est pur, donc en posant :

$$Pu[2\dot{Q}\bar{Q}] = 2\dot{Q}\bar{Q} = \Omega \in P$$

Chapitre 2

Nous obtenons ainsi la dérivée absolue de i par :

$$\begin{aligned}\frac{di}{dt} &= \dot{Q}\bar{Q}i + iQ\dot{\bar{Q}} = \dot{Q}\bar{Q}i - iQ\dot{\bar{Q}} \\ \frac{di}{dt} &= \text{Pu}[2\dot{Q}\bar{Q}i] = \text{Pu}[2\dot{Q}\bar{Q}]\Lambda\vec{i} = \vec{\Omega}\Lambda\vec{i}\end{aligned}$$

Nous retrouvons la définition classique du vecteur rotation de la mécanique et découvrons que ce vecteur est donné par :

$$\Omega_I = 2\dot{Q}\bar{Q} = p_I I + q_I J + r_I K \Leftrightarrow \vec{\Omega}_S = p_I \vec{X} + q_I \vec{Y} + r_I \vec{Z}$$

Où le vecteur rotation est exprimé, en tant que quaternion, dans la base de référence absolue (I,J,K) des quaternions de S et ensuite comme vecteur sur la base absolue (X,Y,Z) de R^3 .

$$\dot{Q} = \frac{1}{2} \Omega_I Q \text{ en axes inertiels}$$

Rappelons que la rotation instantanée est une notion intrinsèque ne dépendant que du mouvement et pas de la base d'expression du vecteur.

Remarque importante

NB : il faut bien faire la différence entre le quaternion et le vecteur associé. Rappelons aussi que la rotation est une donnée intrinsèque qui ne dépend que du mouvement d'une base par rapport à l'autre. Ce sont les expressions des quaternions ou des vecteurs qui changent avec la base.

Il faut bien comprendre que dans l'espace affine R^3 , le vecteur rotation est une donnée intrinsèque. Donc si le quaternion Q transforme la base des quaternions I, J, K (ou des vecteurs X, Y, Z) en la base des quaternions (i, j, k) (ou des vecteurs x, y, z), alors le vecteur Ω devient Ω' , n'ayant pas changé de coordonnées et pour retrouver Ω intrinsèque, il faut effectuer la rotation inverse de quaternion conjugué de Q .

Donc exprimé dans la base mobile, le vecteur rotation, en tant que quaternion, a un quaternion Ω_S qui se déduit de Ω_I par la rotation inverse de Q donc par la rotation de quaternion conjugué de Q

$$\Omega_S = R_{\bar{Q}}(\Omega_I) = \bar{Q}[2\dot{Q}\bar{Q}]Q = 2\bar{Q}\dot{Q}$$

$$\Omega_S = p_i + q_j + r_k \Leftrightarrow \vec{\Omega}_S = p\vec{x} + q\vec{y} + r\vec{z}$$

Où le vecteur rotation est exprimé, en tant que quaternion, dans la base de référence relative i, j, k des quaternions de S et ensuite comme vecteur sur la base relative x, y, z de R^3 .

Donc comme plus haut, mais dans la base relative on a :

$$\Omega_S = 2\bar{Q}\dot{Q} = p_i + q_j + r_k \Leftrightarrow \vec{\Omega}_S = p\vec{i} + q\vec{j} + r\vec{k}$$

Conclusion :

Le même vecteur rotation s'exprime dans chaque base comme ci-dessous

$$\Omega_I = 2\dot{Q}\bar{Q} = \begin{bmatrix} p_I \\ q_I \\ r_I \end{bmatrix} I, J, K \quad : \text{rotation absolue exprimée en axes inertiels I, J, K ou X, Y, Z}$$

Chapitre 2

$$\Omega_S = 2\bar{Q}\dot{Q} = \begin{bmatrix} P_S \\ q_S \\ r_S \end{bmatrix} \text{ i,j,k : rotation absolue exprimée en axes mobiles I,j,k ou x,y,z liés au satellite .}$$

satellite .

Calcul de la matrice de passage

Appelons $(P_{X,Y,Z \Rightarrow x,y,z})$ la matrice de passage de la base de référence absolue à la base mobile relative , liée au satellite . Nous savons que les anciennes composantes s'expriment à l'aide de P et des nouvelles composantes .

Soit Q un quaternion unité élément de S , donc associé à une rotation R_Q :

$$(\vec{X}, \vec{Y}, \vec{Z}) \rightarrow [\vec{x} = R_Q(\vec{X}) \vec{y} = R_Q(\vec{Z})]$$

Où

$$\text{Traduit en quaternions , appliqué à V : } R_Q(\vec{V}) = Q \Lambda \bar{Q}$$

Pour calculer la matrice de changement de base on écrit :

$$x = R_Q(X) = P_{11}X + P_{21}Y + P_{31}Z$$

$$y = R_Q(Y) = P_{12}X + P_{22}Y + P_{32}Z$$

$$z = R_Q(Z) = P_{13}X + P_{23}Y + P_{33}Z$$

Si nous faisons le calcul pour la première relation , à partir du transformé de X qui donne le vecteur x , il vient tous calculs terminés , dans la base absolue que :

$$P = \begin{bmatrix} 2(q_0^2 + q_1^2) - 1 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & 2(q_0^2 + q_2^2) - 1 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & 2(q_0^2 + q_3^2) - 1 \end{bmatrix}$$

Les éléments de P sont notés P_{ij} et a_{ij} ceux de P^{-1}

Nb : vous pourrez vérifier que l'écriture de P est la même que ce soit avec Q exprimée dans I J K ou Q

Equation de comportement du quaternion en axes satellite

A l'expérience , les équations en axes satellite sont les plus importantes dans les applications pratiques , puisqu'en mécanique , il apparait clairement que les axes satellites sont évidemment mieux placés par rapport au corps que ceux du référentiel fixe .

L'évolution temporelle du quaternion est donnée par l'équation :

$$\Omega_S = 2\bar{Q}\dot{Q} = pi + qj - rk$$

$$\dot{Q} = \frac{1}{2}Q\Omega_S \text{ multiplication des quaternions}$$

$$\dot{Q} = \frac{1}{2}(q_0e - q_1i + q_2j + q_3k)(pi + qj - rk) = \begin{bmatrix} -pq_1 - qq_2 - rq_3 \\ pq_0 - qq_3 + rq_2 \\ pq_3 + qq_0 - rq_1 \\ -pq_2 + qq_1 + rq_0 \end{bmatrix}$$

Donnant ainsi une relation fondamentale matricielle d'évolution du quaternion exprimé en axes satellite avec une matrice M_3 qui se calcule avec les composantes de Ω_S :

Chapitre 2

$$\dot{Q} = \frac{dQ}{dt} = \frac{1}{2} M_S \cdot Q = \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

Equations du mouvement d'un satellite autour de son centre d'inertie

La résolution du problème de l'attitude d'un corps dans l'espace consiste donc à résoudre en parallèle sept équations totalement couplées :

1. Une équation vectorielle équivalente à 3 équations différentielles scalaires , résultant théorème du moment cinétique appliqué au centre d'inertie du solide et fournissant p,q,r , donc la rotation instantanée absolue .

2. Une équation vérifiée par le quaternion Q, équivalente à 4 équations différentielles scalaires , traduisant les équations d'évolution du quaternion d'attitude , nécessitant la connaissance de p , q , r calculés en parallèle . Son intégration fournit donc le quaternion donnant l'attitude du satellite .

$\dot{\Omega} = p\hat{x} + q\hat{y} + r\hat{z}$	Conditions : repère x, y, z lié au satellite et moment des forces en G, connu sur les axes satellite
$\dot{\Omega}(t=0) = \dot{\Omega}_0$	
$\frac{dQ}{dt} = \frac{1}{2} M_S \cdot Q$ $M_S = \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix}$ $Q(t=0) = Q_0$	Equation d'évolution du quaternion d'attitude Q et conditions initiales permettant le calcul de Q. La connaissance de Q permet le calcul de la matrice de passage de la base fixe à celle du satellite.
$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$	Equation de redondance, permettant la vérification du bon fonctionnement de l'algorithme de calcul
$\frac{d}{dt} \bar{I} \dot{\Omega} + \dot{\Omega} \wedge \bar{I} \dot{\Omega} = \bar{M}_G$	Théorème du moment cinétique, donnant p, q, r grâce aux conditions initiales sur Ω .

La dernière relation est une équation redondante à vérifier et surtout destinée à rappeler qu'il faut veiller à normer le quaternion d'attitude tout au long du calcul.

Nb1 : il est recommandé de normer le quaternion Q tout au long des calculs d'intégration lorsque nécessaire . De plus , il vaut mieux travailler en axes mobiles qu'en axes fixes (simple respect des symétries du solide ou de ses axes principaux) :

L'attitude du satellite en référentiel absolu , s'en déduit alors par l'intermédiaire de la matrice de passage P , elle-même fonction de Q .

Chapitre 2

$$P = \begin{matrix} & \begin{matrix} \hat{x} & \hat{y} & \hat{z} \end{matrix} \\ \begin{matrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{matrix} & \begin{bmatrix} 2(q_0^2 + q_1^2) - 1 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 - q_0q_3) & 2(q_0^2 + q_2^2) - 1 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & 2(q_0^2 + q_3^2) - 1 \end{bmatrix} \end{matrix}$$

Changement de base XYZ → xyz

Si l'on s'intéresse à un mouvement rapporté à un référentiel non absolu, il suffit d'effectuer les changements de repère, concernant les positions et les vitesses.

Nb 2 : on peut aussi évaluer le vecteur rotation relatif du satellite par rapport à ce repère R, et calculer alors la matrice M_S , qui permet de déterminer le quaternion d'attitude Q du satellite par rapport à R.

Par exemple, si le repère relatif à une rotation instantanée connue dans le repère absolu ($\Omega_x, \Omega_y, \Omega_z$), alors il faudra remplacer p, q, r par p*, q*, r* dans le calcul de M_S .

$$\begin{aligned} \hat{\Omega}_{(Sat/R)} &= \hat{\Omega}_{(Sat/R_a)} - \hat{\Omega}_{(R/R_a)} \\ &\text{soit} \\ \begin{bmatrix} p^* \\ q^* \\ r^* \end{bmatrix} &= \begin{bmatrix} p \\ q \\ r \end{bmatrix} - \begin{bmatrix} P_{11} & P_{21} & P_{31} \\ P_{12} & P_{22} & P_{23} \\ P_{13} & P_{23} & P_{33} \end{bmatrix} \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} \end{aligned}$$

Initialisation du quaternion d'attitude

Formules d'initialisation

Il paraît évident que l'initialisation doit être à l'aide des angles de repérage et donc de la matrice P de passage qui s'exprime en fonction de ces angles.

Nous supposons donc connue la matrice P par ses coefficients P_{ij} .

On peut calculer le quaternions d'attitude avec la matrice P, par des relations suivantes :

$$\begin{aligned} 4q_1^2 &= 1 + P_{11} - P_{22} - P_{33} \\ 4q_2^2 &= 1 + P_{22} - P_{33} - P_{11} \\ 4q_3^2 &= 1 + P_{33} - P_{11} - P_{22} \\ 4q_0^2 &= 1 - P_{11} + P_{22} + P_{33} \\ 4q_1q_2 &= P_{12} + P_{21} \\ 4q_2q_3 &= P_{23} + P_{32} \\ 4q_3q_1 &= P_{31} + P_{13} \\ 4q_0q_3 &= P_{21} - P_{12} \\ 4q_0q_2 &= P_{13} - P_{31} \\ 4q_0q_1 &= P_{32} - P_{23} \end{aligned}$$

Une série de test est nécessaire sur les éléments diagonaux pour calculer le quaternion à une indétermination près de signe. Dans tous les cas, on supposera cependant que q_0 est dans l'intervalle [0,1].

En pratique, dès que le signe d'une composante de Q est choisi, il détermine automatiquement le signe des autres, mais la matrice P de passage n'en est pas affectée. Il n'y a donc pas lieu de s'inquiéter, et si q_0 est différent de 0, nous prendrons toujours cette composante positive sinon ce sera q_1 etc....

Quaternions et gradient de gravité sur un satellite

Chapitre 2

La gravité, à ne pas confondre avec la pesanteur au sens terrestre du terme, ne s'applique pas, en général, au centre d'inertie d'un satellite, contrairement à ce que l'on pense. La ligne d'action est certes décalée d'une quantité infime par rapport à ce point privilégié, mais cela suffit à créer sur tout corps non inertiellement sphérique, un couple appelé couple du gradient de gravité. Le propos n'est pas de démontrer ici son mode de calcul.

Mise en œuvre de l'étude

La simulation et l'étude du mouvement ne sont pas faciles avec le quaternion d'attitude. Pour l'interprétation, il faut donc revenir à la représentation initiale avec les angles classiques, via la matrice de passage P ou son inverse A.

Nb : il faudra également probablement exprimer certaines composantes de forces à l'aide du quaternions d'attitude. Un exemple est donné ci-dessous.

C'est le prix à payer pour avoir une intégration numérique paraît-il très stable

1°) Repères- Notations- Angles de cardan :

Nous nous intéressons par exemple, au problème de la stabilisation d'un satellite d'imagerie spatiale, qui doit rester dans une position invariable par rapport au sol survolé.

O Xa Ya Za est un repère galiléen, qu'il n'est pas utile de préciser outre mesure.

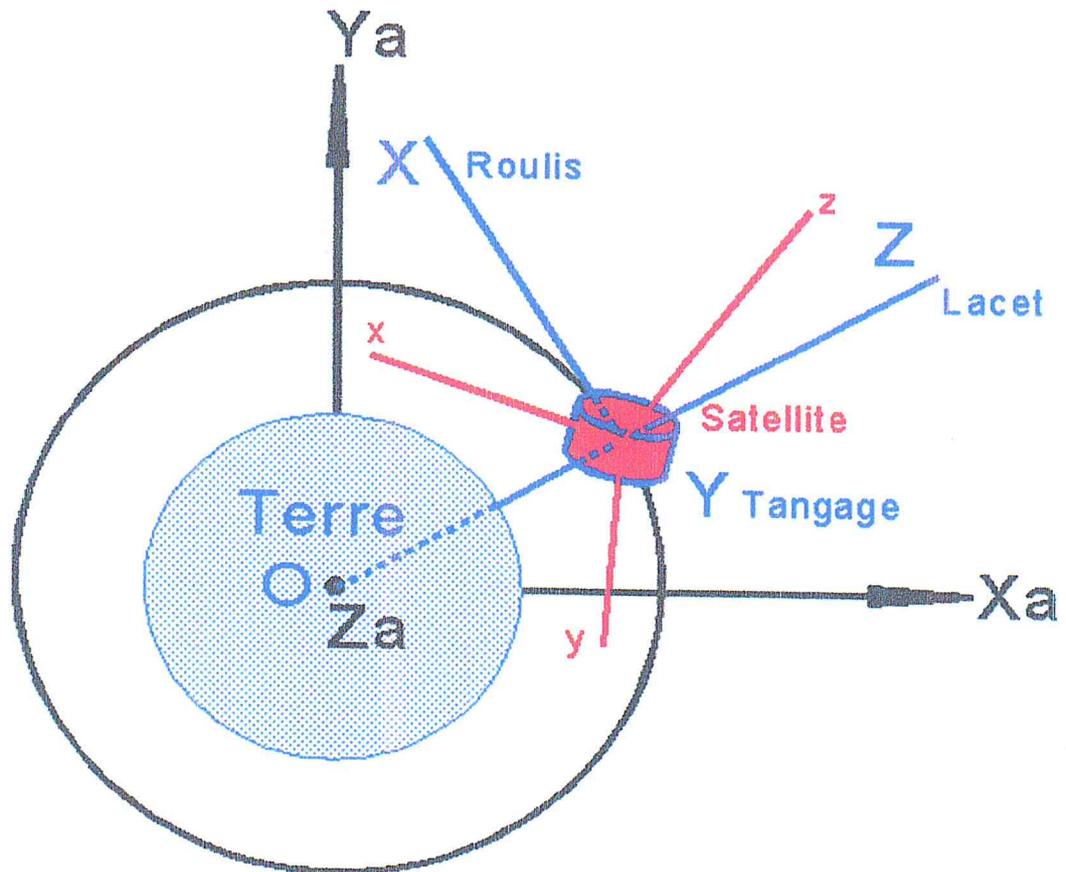
Le satellite S est en orbite, supposée circulaire, de rayon r_0 . On appelle **REPERE ORBITAL Ro** le repère d'origine S et d'axes X Y Z, avec :

X axe dit de roulis, unitaire de la vitesse orbitale, tangent à l'orbite.

Y axe de tangage, unitaire du moment cinétique, normal à la trajectoire.

Z axe de lacet, suivant le géocentrique.

On désigne par S x y z le repère R, principal d'inertie pour le satellite, avec I_R, I_T, I_L les **moments principaux d'inertie**.



Comme un satellite d'imagerie doit garder des axes fixes par rapport à ce repère orbital R_o , nous sommes amenés à définir des angles particuliers, qui pour la cas d'espèce resteront voisins de 0, tout au long de la vie du satellite, tant que l'asservissement de stabilisation fonctionne.

Les repères sont définis comme suit, après avoir indiqué que l'axe a est la projection sur le plan horizontal X, Y de l'axe x . La succession de repères est :

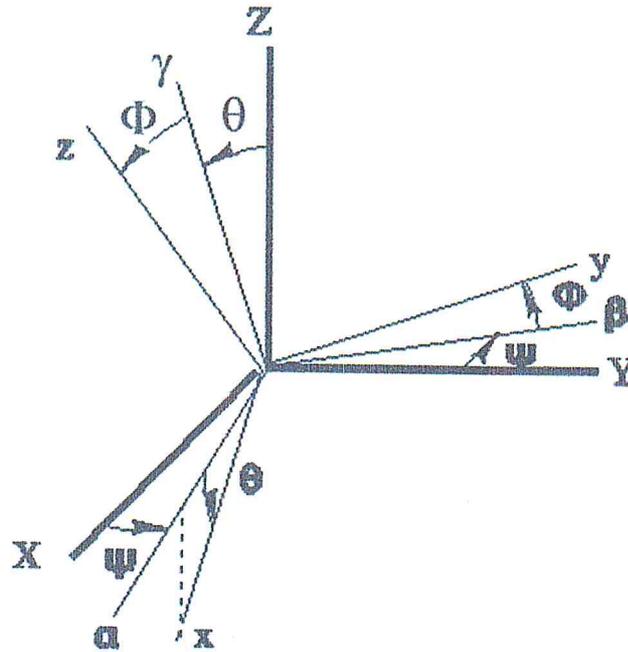
$$XYZ \rightarrow Y \rightarrow abZ \rightarrow q \rightarrow Xbg \rightarrow F \rightarrow xyz$$

Nous avons ainsi défini les angles conventionnels, également appelés ANGLES DE CARDAN :

Lacet y mesuré autour de Z

Tangage q mesuré autour de b (voisin de Y lorsque les angles sont petits)

Roulis f mesuré autour de x (voisin de X lorsque les angles sont petits)



La matrice P de passage de XYZ à xyz s'explique classiquement :

$$P = \begin{bmatrix} \cos\theta\cos\psi & -\sin\psi\cos\Phi + \cos\psi\sin\Phi\sin\theta & \sin\psi\sin\Phi + \cos\psi\cos\Phi\sin\theta \\ \cos\theta\sin\psi & \cos\psi\cos\Phi + \sin\psi\sin\Phi\sin\theta & -\cos\psi\sin\Phi + \sin\psi\cos\Phi\sin\theta \\ -\sin\theta & \sin\Phi\cos\theta & \cos\Phi\cos\theta \end{bmatrix}$$

S à comparer à son expression déduite du quaternion $Q = (q_0 q_1 q_2 q_3)$ qui représente la rotation globale composée des 3 rotations élémentaires définies plus haut :

$$P(\text{XYZ} \rightarrow \text{xyz}) = \begin{bmatrix} 2(q_0^2 + q_1^2) - 1 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & 2(q_0^2 + q_2^2) - 1 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & 2(q_0^2 + q_3^2) - 1 \end{bmatrix}$$

Ce qui permet d'une certaine façon d'exploiter les résultats pour évaluer les conditions initiales ou les angles en fonction du quaternion.

Exprimé dans les axes satellites *le vecteur rotation instantanée absolue* a pour composantes :

$$P = \dot{\Phi} - \dot{\psi} \sin\theta + \omega_0 \sin\psi \cos\theta$$

$$q = \dot{\psi} \cos\theta \sin\Phi + \dot{\theta} \cos\Phi + \omega_0 (\cos\psi \cos\Phi + \sin\theta \sin\Phi \sin\psi)$$

$$r = \dot{\psi} \cos\theta \cos\Phi - \dot{\theta} \sin\Phi + \omega_0 (\sin\theta \sin\psi \cos\Phi - \cos\psi \sin\Phi)$$

Où $\omega_0 = \sqrt{\frac{\mu}{r_0^2}}$ est la vitesse angulaire orbitale

Chapitre 2

Vous noterez qu'il s'agit de la rotation galiléenne mais que la rotation à annuler quand on souhaite obtenir un pointage terre parfait est la rotation relative au repère orbital de composantes en roulis, tangage et lacet.

$$p_0 = \dot{\Phi} - \dot{\psi} \sin \theta$$

$$q_0 = \dot{\psi} \cos \theta \sin \Phi + \dot{\theta} \cos \Phi$$

$$r_0 = \dot{\psi} \cos \theta \cos \Phi - \dot{\theta} \sin \Phi$$

C'est d'ailleurs cette rotation qu'il faut utiliser dans l'équation différentielle donnant le quaternion Q de l'attitude dans XYZ.

2°) Expression du quaternion en fonction des angles de Cardan :

Les quaternions associés aux rotations élémentaires et à la rotation résultante sont :

$$\left[\begin{array}{l} Rot(\psi, \vec{Z}) \leftrightarrow Q_\psi = \left[\cos \frac{\psi}{2} + \sin \frac{\psi}{2} \vec{Z} \right] \\ Rot(\theta, \vec{\beta}) \leftrightarrow Q_\theta = \left[\cos \frac{\theta}{2} + \sin \frac{\theta}{2} \vec{\beta} \right] \\ Rot(\Phi, \vec{X}) \leftrightarrow Q_\Phi = \left[\cos \frac{\Phi}{2} + \sin \frac{\Phi}{2} \vec{X} \right] \end{array} \right] \Rightarrow Q = Q_\Phi Q_\theta Q_\psi$$

Tous calculs effectués, on obtient les relations remarquables suivantes

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos \frac{\Phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\Phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \\ \sin \frac{\Phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} - \cos \frac{\Phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \\ \cos \frac{\Phi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\Phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} \\ \sin \frac{\Phi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} - \cos \frac{\Phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} \end{bmatrix}$$

qui permettent une initialisation sans problèmes du quaternion Q.

3°) Calcul du gradient de gravité en termes de quaternions

On vérifie que le gradient de gravité s'exprime à l'aide du quaternion d'attitude sous la forme , avec roulis Φ , tangage θ , lacet ψ .

$$\vec{F} / \vec{x}\vec{y}\vec{z} = 3\omega_0^2 \vec{Z} \Lambda \vec{I} \vec{Z} = \begin{bmatrix} 3\omega_0^2 (I_L - I_T) \sin \Phi \cos \Phi \cos^2 \theta \\ -3\omega_0^2 (I_R - I_L) \sin \theta \cos \theta \cos \Phi \\ -3\omega_0^2 (I_T - I_R) \sin \Phi \cos \theta \sin \theta \end{bmatrix}$$

Chapitre 2

Comme nous savons calculer les matrices de passage $P(XYZ \rightarrow xyz)$ et P^{-1} donne :

$$\vec{Z} = \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \end{bmatrix} = \begin{pmatrix} -\sin\theta \\ \sin\Phi \cos\theta \\ \cos\Phi \cos\theta \end{pmatrix} = \begin{pmatrix} \vec{Z}_{\bar{x}} \\ \vec{Z}_{\bar{y}} \\ \vec{Z}_{\bar{z}} \end{pmatrix} = \begin{pmatrix} 2(q_1 q_3 - q_0 q_2) \\ 2(q_2 q_3 + q_0 q_1) \\ 2(q_0^2 + q_3^2) - 1 \end{pmatrix}$$

et donc l'expression du couple exact dû au gradient de gravité :

$$\vec{\Gamma} / \vec{x}\vec{y}\vec{z} = 3\omega_0^2 \vec{Z} \wedge \vec{I} \vec{Z} = 3\omega_0^2 \begin{bmatrix} (I_L - I_T)a_{23}a_{33} \\ (I_R - I_L)a_{33}a_{13} \\ (I_T - I_R)a_{13}a_{23} \end{bmatrix} = 3\omega_0^2 \begin{bmatrix} (I_L - I_T)P_{32}P_{33} \\ (I_R - I_L)P_{33}P_{31} \\ (I_T - I_R)P_{31}P_{32} \end{bmatrix}$$

II) Equations du mouvement :

On note $\vec{C} = C_X \vec{X} + C_Y \vec{Y} + C_Z \vec{Z}$ le couple de commande éventuel appliqué en contrôle d'attitude.

Avec les notations présentées plus haut, on obtient les équations du théorème du moment cinétique.

$$I_R \frac{dp}{dt} + (I_L - I_T)qr = \Gamma_x + C_X$$

$$= 6\omega_0^2 (I_L - I_T)(q_2 q_3 + q_0 q_1)(2q_0^2 + 2q_3^2 - 1) + C_X$$

$$I_T \frac{dq}{dt} + (I_R - I_L)rp = \Gamma_y + C_Y$$

$$= 6\omega_0^2 (I_R - I_L)(q_1 q_3 - q_0 q_2)(2q_0^2 + 2q_3^2 - 1) + C_Y$$

$$I_L \frac{dr}{dt} + (I_T - I_R)pq = \Gamma_z + C_Z$$

$$= 12\omega_0^2 (I_T - I_R)(q_2 q_3 + q_0 q_1)(q_1 q_3 - q_0 q_2) + C_Z$$

$$\frac{dQ}{dt} = \frac{1}{2} M_S Q$$

$$Q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad \text{avec} \quad \begin{bmatrix} P_{21} = 2(q_1 q_2 + q_0 q_3) \\ P_{22} = 2(q_0^2 + q_2^2) - 1 \\ P_{23} = 2(q_3 q_2 - q_0 q_1) \end{bmatrix}$$

$$M_S = \begin{bmatrix} 0 & -p + \omega_0 P_{21} & -q + \omega_0 P_{22} & -r + \omega_0 P_{23} \\ p - \omega_0 P_{21} & 0 & r - \omega_0 P_{23} & -q + \omega_0 P_{22} \\ q - \omega_0 P_{22} & -r + \omega_0 P_{23} & 0 & p - \omega_0 P_{21} \\ r - \omega_0 P_{23} & q - \omega_0 P_{22} & -p + \omega_0 P_{21} & 0 \end{bmatrix} r$$

Chapitre 2

III) Formalisation du problème

On propose :

De prendre comme variable un vecteur d'état X à 7 coordonnées $X = [p \ q \ r \ q_0 \ q_1 \ q_2 \ q_3]$

D'écrire le système différentiel sous une forme canonique où F fonction vectorielle ne dépend que des inerties, de la pulsation orbitale et du vecteur d'état X , ce qui rendra l'adaptation plus facile pour d'autres applications :

$$\frac{dX}{dt} = F(X) + BU$$

B ne dépend que des inerties et U est le vecteur d'entrée du système, dont les trois premières composantes sont les couples normalisés en axes satellite.

$$B = \begin{bmatrix} \frac{1}{I_r} & 0 & 0 \\ 0 & \frac{1}{I_t} & 0 \\ 0 & 0 & \frac{1}{I_l} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad U = \begin{bmatrix} C_X / I_R \\ C_Y / I_T \\ C_Z / I_L \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Chapitre 3

3.1 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.

3.2 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.

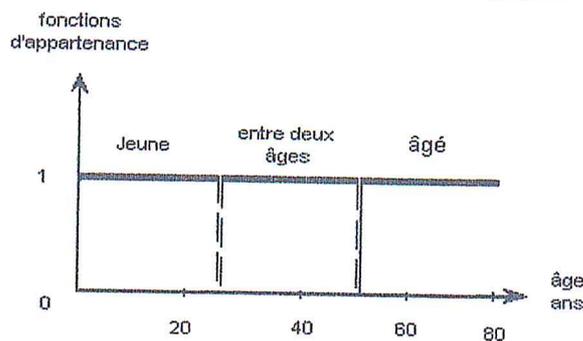


Figure 3.1 : Classification des personnes en trois sous-ensembles selon la logique classique.

La logique floue, dont les variables peuvent prendre n'importe quelle valeur entre 0 et 1 (on parle alors de *fonction d'appartenance* μ), 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.

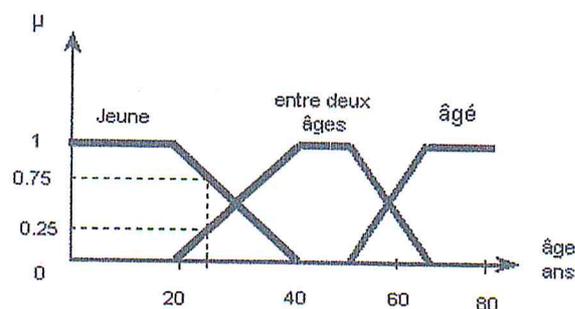
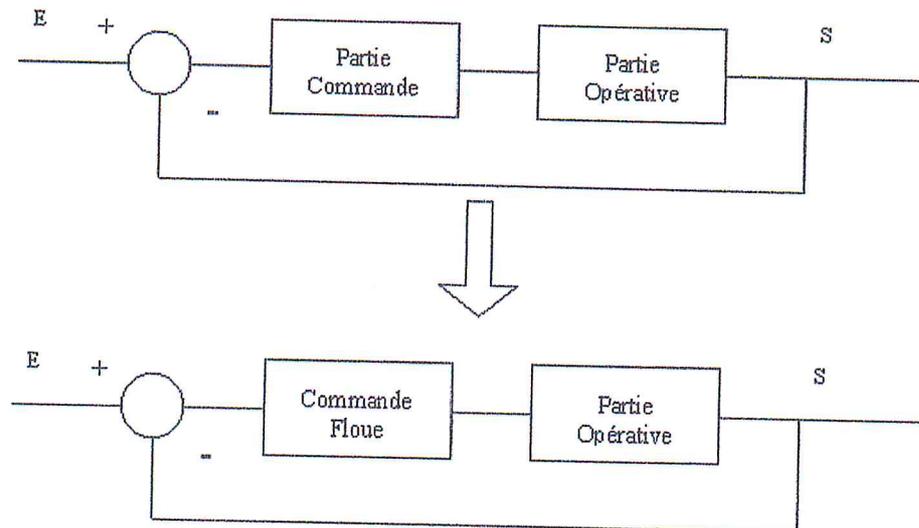


Figure 3.2 : Classification des personnes en trois sous-ensembles selon la logique floue

Chapitre 3

3.3 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.

Chapitre 3

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.

3.4 Bases générales de la logique floue

Les éléments de base de la logique floue sont :

3.4.1 Les sous-ensemble Flou [7] [15] [18]

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

- A est un *sous-ensemble* de U .
- A , un label linguistique, caractérisant qualitativement une partie des valeurs de x .
- μ_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* $\{A\}$.

3.4.2 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.

3.4.2.1 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).

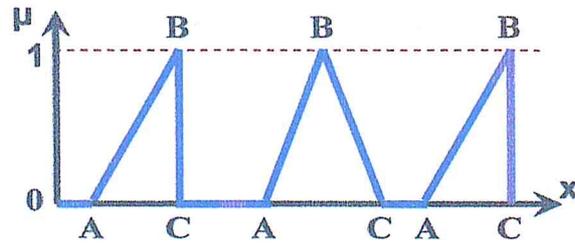


Figure III.3 : Fonctions d'appartenance de forme triangulaire.

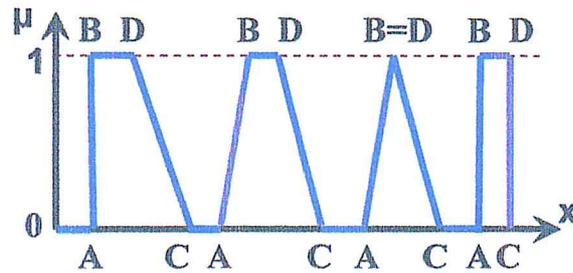
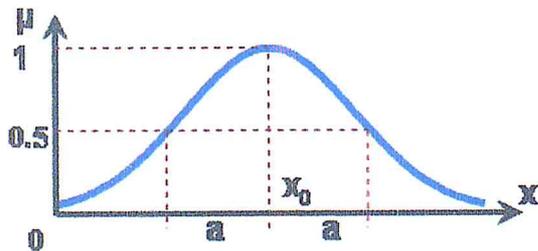


Figure III.4 : Fonction d'appartenance de forme trapézoïdale.

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.

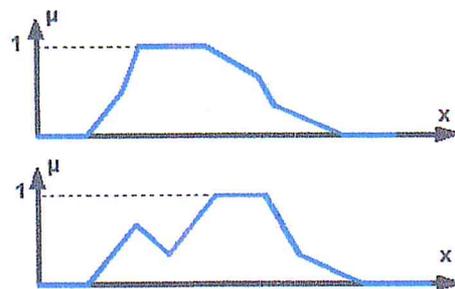


Figure III.6.: Fonctions d'appartenances composées de segments de droites (fonction convexe et concave).

3.4.2.2 Considérations générales sur les fonctions d'appartenances [22]

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

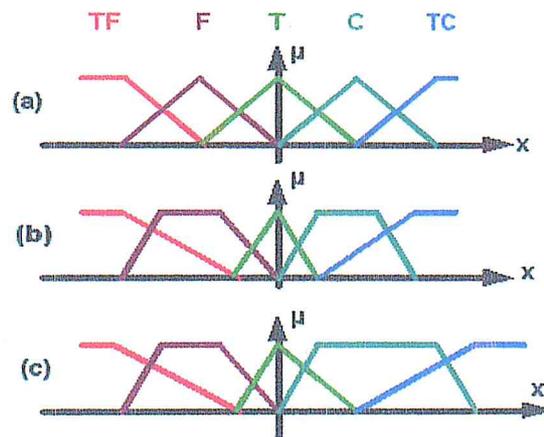


Figure III.7 : Structure des fonctions d'appartenance :

- (a) symétrique et équidistante ; (b) symétrique et non équidistante ;
(c) non symétrique et non équidistante.

3.4.3 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.

3.4.3.1 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

Chapitre 3

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 :

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

(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 $\mu_{x_1}, \mu_{x_2}, \dots, \mu_{x_n}$, dont les valeurs numériques (facteurs d'appartenance) sont en général différentes pour chaque concurrent.

3.4.3.2 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 :

$$\begin{array}{l} \text{Opération := Si condition 1, Alors opération 1, OU} \\ \text{Si condition 2, Alors opération 2, OU} \\ \vdots \\ \text{Si condition m, Alors opération m.} \end{array}$$

(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

Chapitre 3

ces variables. En général, il faut aussi introduire des fonctions d'appartenance pour les opérateurs, surtout si ces dernières sont exprimées par des variables floues.

3.4.4 Opérateurs de la logique floue

Les variables linguistiques sont liées entre elles au niveau des inférences par des opérateurs « ET » ou « OU ». Il s'agit d'*opérateurs de la logique floue* qui interviennent sur les fonctions d'appartenance représentant les variables linguistiques. De plus, il existe un opérateur « NON » (complément, négation, inverse).

Remarque : Traduites sous forme de règles « min » et « max », les nouveaux opérateurs ainsi définis présentent une certaine affinité avec les règles de calcul pour ces opérateurs dans le cas de la logique classique.

3.4.4.1 Représentation de l'opérateur NON par le complémentaire de A

Le complémentaire de A, \bar{A} peut être exprimé par la fonction d'appartenance :

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) ; \forall x \in U$$

(III.4)

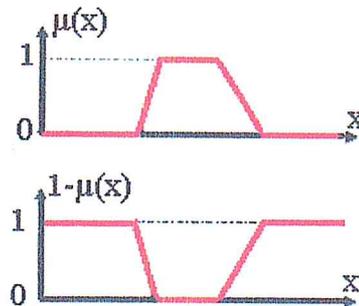
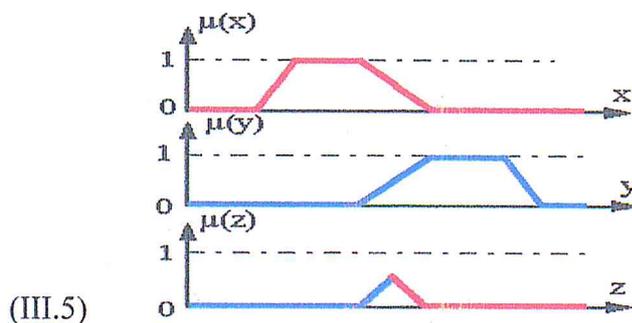


Figure III.8 : Opérateur NON.

3.4.4.2 Opérateur ET

Le sous-ensemble A ET B = $A \cap B$ est représenté par la fonction d'appartenance :

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



(III.5)

Figure III.9 : Opérateur ET.

3.4.4.3 Opérateur OU

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

$$\mu_{A \cup B}(z) = \max\{\mu_A(x), \mu_B(y)\}; \forall x, y, z \in U \quad (\text{III.6})$$

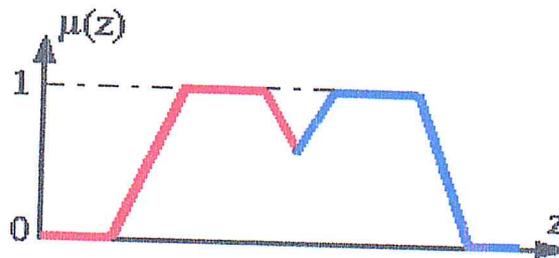


Figure III.10 : Opérateur OU.

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

$$T : [0,1] \times [0,1] \rightarrow [0,1]$$

$$x, y \rightarrow z = x T y$$

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 :

$$S : [0,1] \times [0,1] \rightarrow [0,1]$$

$$x, y \rightarrow z = x S y$$

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.

3.5 Structure d'une commande floue

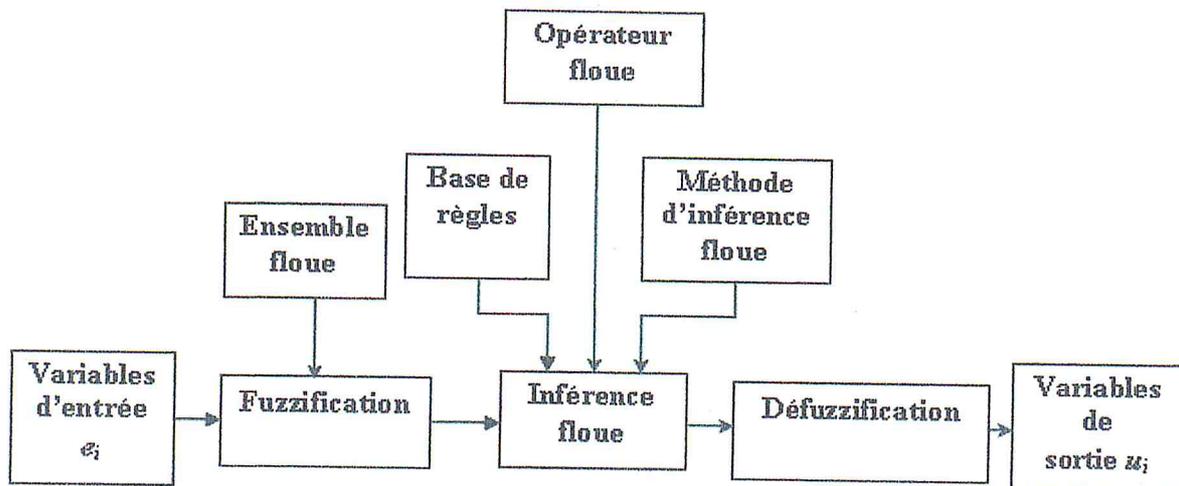
3.5.1 Généralité [12] [14]

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.



3.5.2 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 à priori (possible apparition de règles d'inférence contradictoires).

3.5.3 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 m 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 :

$$r_{\max} = \prod_{i=1}^m m_i$$

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*.

3.5.4 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 :

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

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)).

Chapitre 3

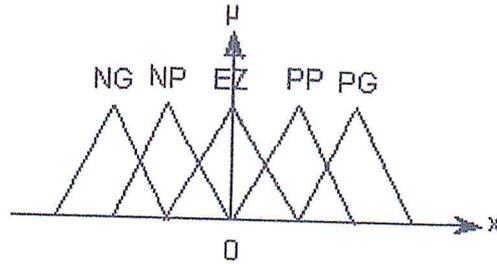


Figure III.11 : Représentations des cinq sous-ensembles flous.

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

⋮

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 :

Chapitre 3

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

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 :

$$\mu_{res}(x_r) = \max[\mu_{r_1}(x_r), \mu_{r_2}(x_r), \dots, \mu_{r_m}(x_r)] \quad (\text{III.10})$$

➤ 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 $\mu_{res}(z)$.

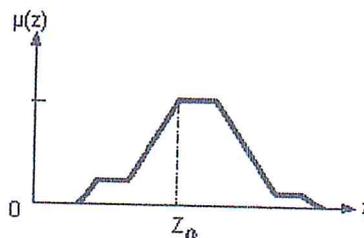
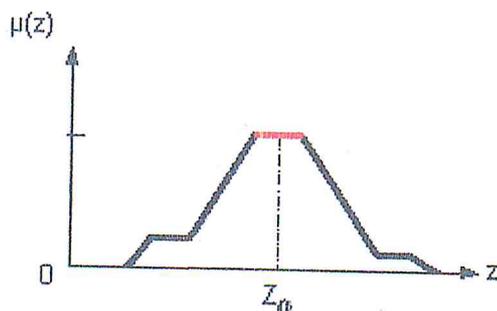


Figure III.13 : Défuzzification par le premier maximum.

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



$$Z_0 = \sum_{i=1}^l \frac{r_i}{l} \quad (\text{II.11})$$

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

Chapitre 3

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.

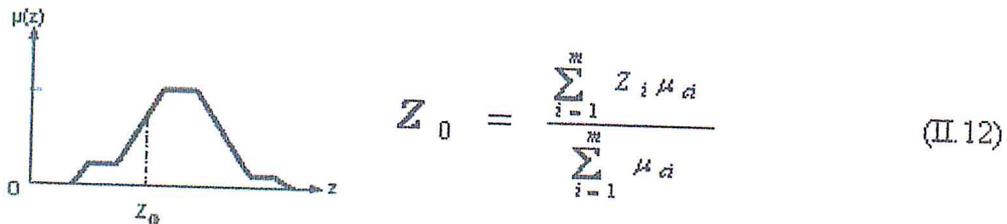


Figure III.15 : Défuzzification par la méthode de la moyenne pondérée.

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 $\mu_{res}(z)$.

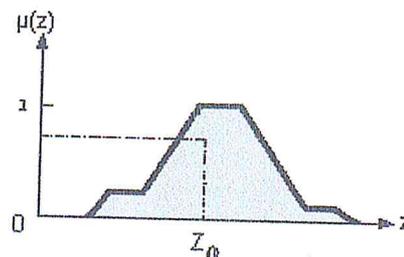


Figure III.16 : Défuzzification par le centre de gravité.

Pour le cas d'inférence max-min on obtient

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

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

Chapitre 3

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 à 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.

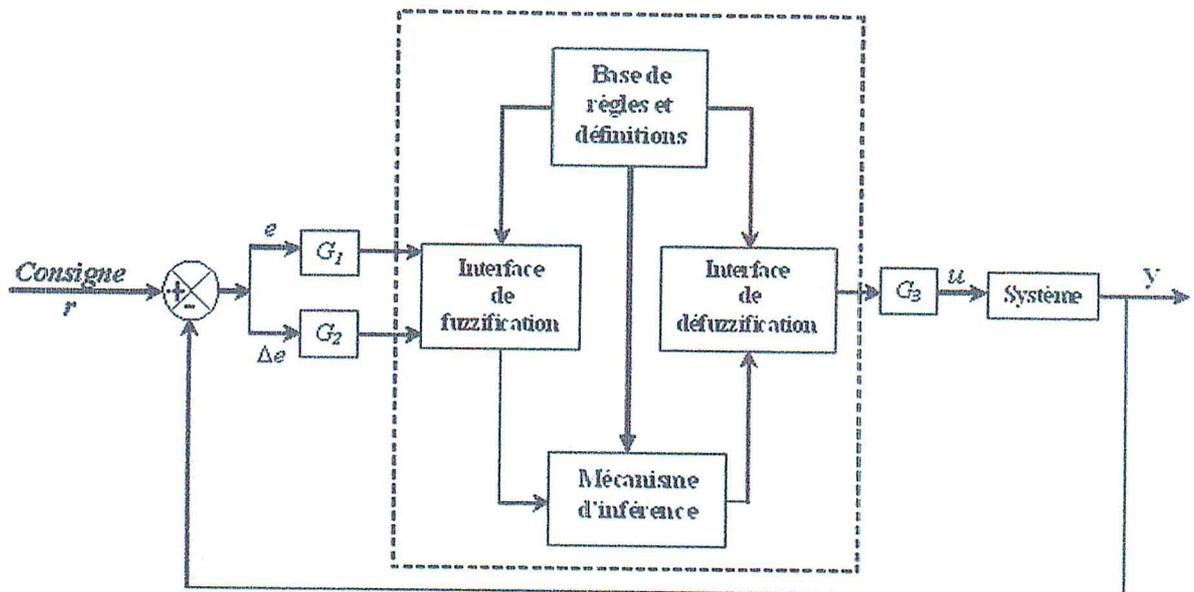


Figure IV.1 : Structure d'une commande floue.

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 Δe .

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, permettrons d'approcher les consignes souhaitées.

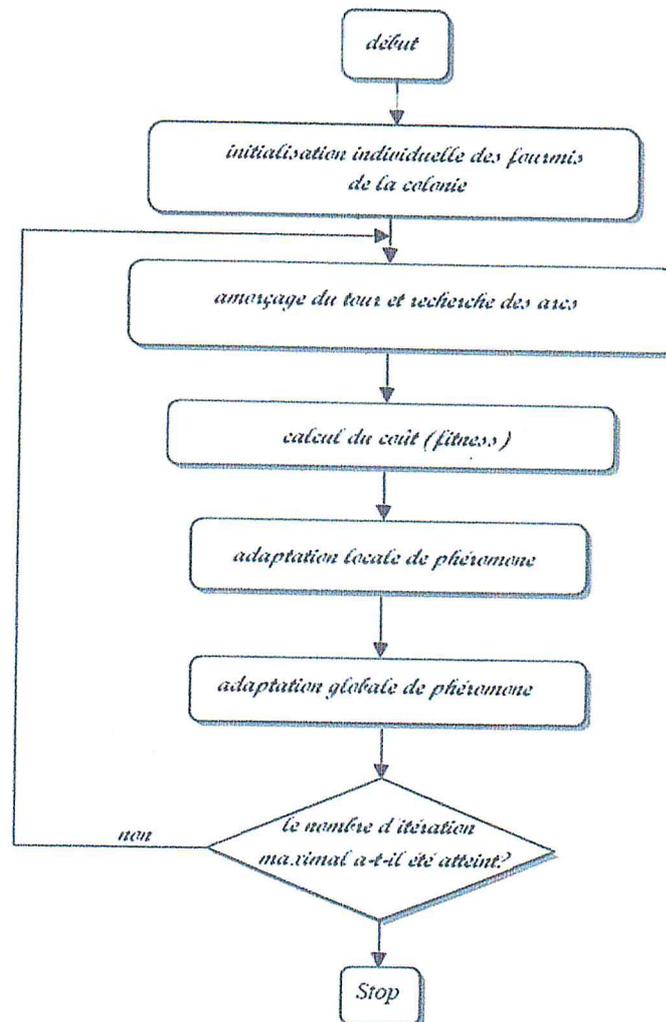


figure 4.2 : Organigramme des colonies de fourmis représentant la méthodologie d'optimisation des paramètres d'un contrôleur flou

Vecteur associé à la structure des données

Le vecteur d'un ensemble flou peut être subdivisé en deux types de séquences :

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

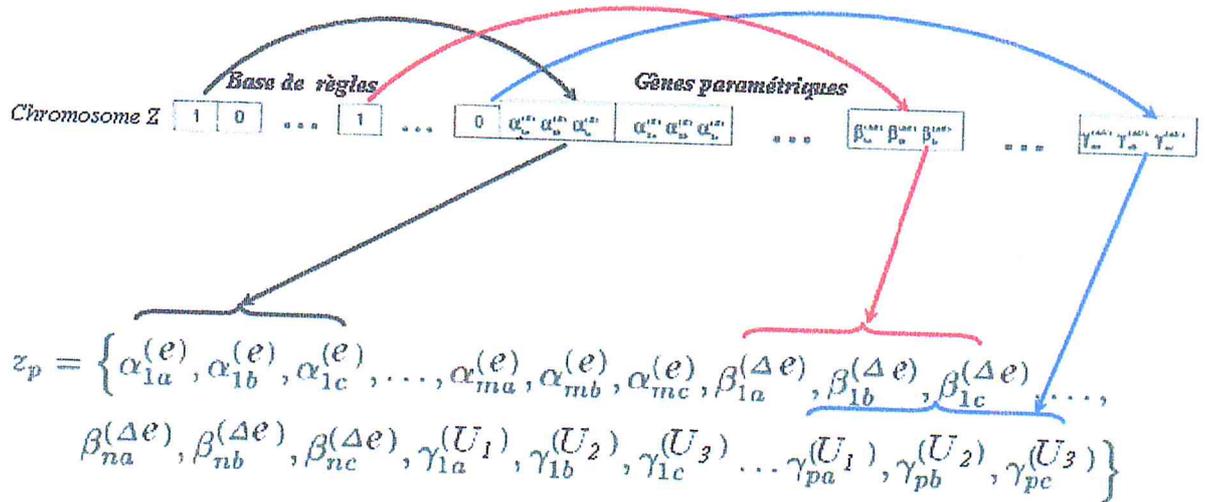


Figure IV.4 : Structure des fonctions d'appartenance du contrôleur flou.

Tel que :

- m, n et p : le nombre maximum de sous-ensemble flou de $e, \Delta e$ et u respectivement.
- Les paramètres du $i^{\text{ème}}$ sous-ensemble flou d'entrée de e .
- Les paramètres du $j^{\text{ème}}$ sous-ensemble flou de variation d'entrée Δe .
- Les paramètres du $k^{\text{ème}}$ sous-ensemble flou de sortie u .

IV.3 Les facteurs d'échelle (gain de normalisation)

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.

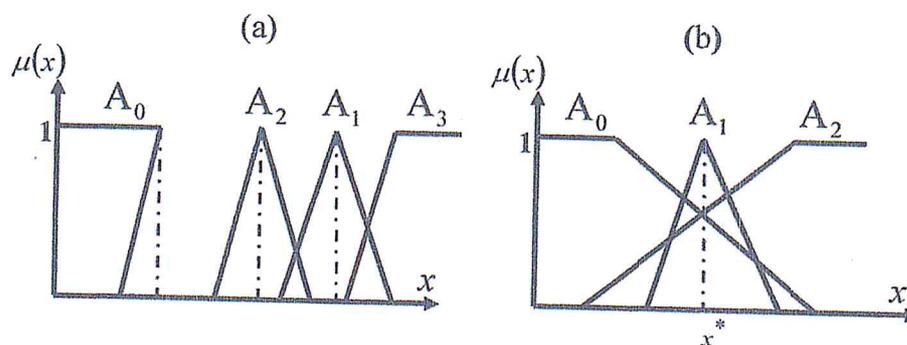


Figure IV.5 : Les méfaits d'une absence de contraintes sur les fonctions d'appartenance.

Chapitre 4

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 δ 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 :

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

Le vecteur de sortie est le suivant :

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

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

IV.5.1 Étude du mouvement libre du satellite

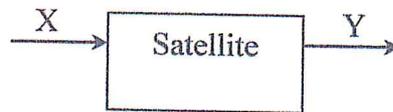


Figure IV.6 : Représentation du mouvement libre du satellite.

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

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

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:

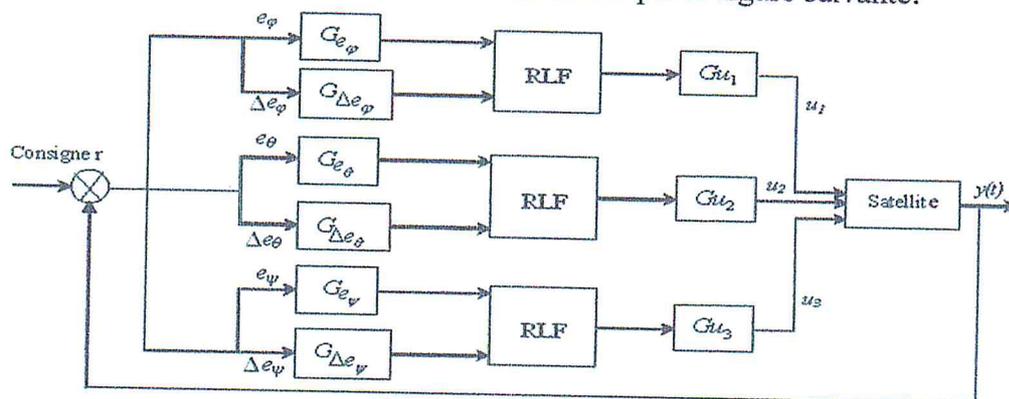


Figure IV. 7: Schéma fonctionnel d'un système asservi.

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_ϕ et sa variation Δe_ϕ ;
 Régulateur (2) : l'erreur e_θ et sa variation Δe_θ ;
 Régulateur (3) : l'erreur e_ψ et sa variation Δe_ψ
 Les variables de la sortie sont : u_1 , u_2 et u_3 .

Chapitre 4

- 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 :

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

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

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

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

La table de décision est représenté par :

		$u(t)$	$\Delta e(t)$		
			N	Z	P
$e(t)$	N	P	P	Z	
	Z	P	Z	N	
	P	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 séquence de 17 composantes (son équivalent géométrique aurait été associé à un vecteur de 17 composantes).

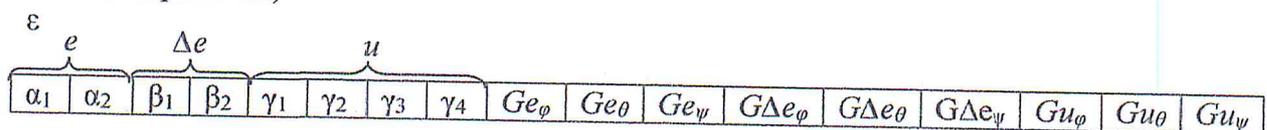


Figure IV.9 : séquence des composantes pour trois fonctions d'appartenance.

Chapitre 4

Les huit premières composantes représentent les projections des fonctions d'appartenance (e , Δe , u) sur l'axe de l'univers du discours normalisé. Pour éviter les méfaits d'une absence de contraintes sur les fonctions d'appartenance, nous imposons les conditions suivantes :

$$-1 < \alpha_1 < \alpha_2 < 0 \dots \dots \dots (1)$$

$$-1 < \beta_1 < \beta_2 < 0 \dots \dots \dots (2)$$

$$-1 < \gamma_1 < \gamma_2 < \gamma_3 < \gamma_4 \dots \dots \dots (3)$$

Tandis que les neuf dernières composantes, elles représentent les gains; tel que :

	ε_1	ε_2	ε_3
$G\varepsilon$	[0 1/]	[0 1/]	[0 1/]
$G\Delta\varepsilon$	[3 12]	[3 6]	[8 12]
Gu	[15 30]	[15 60]	[2 3]

Tableau IV.2 : Domaines d'appartenance des gains de normalisations de ε et $\Delta\varepsilon$ 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.

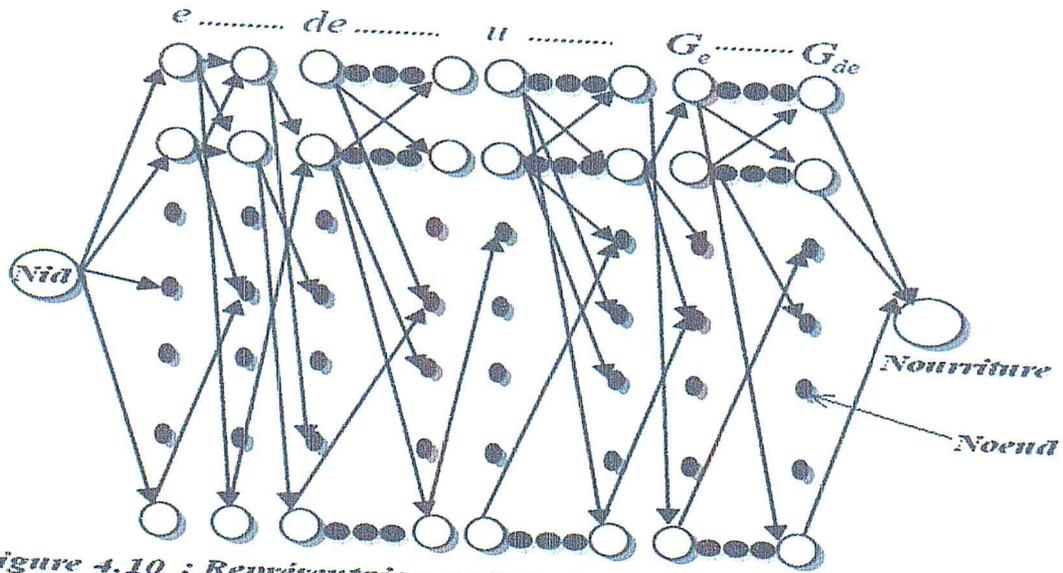


Figure 4.10 : Représentation nodale des possibilités de parcours

Fonction d'adaptation, d'ajustement ou de fitness

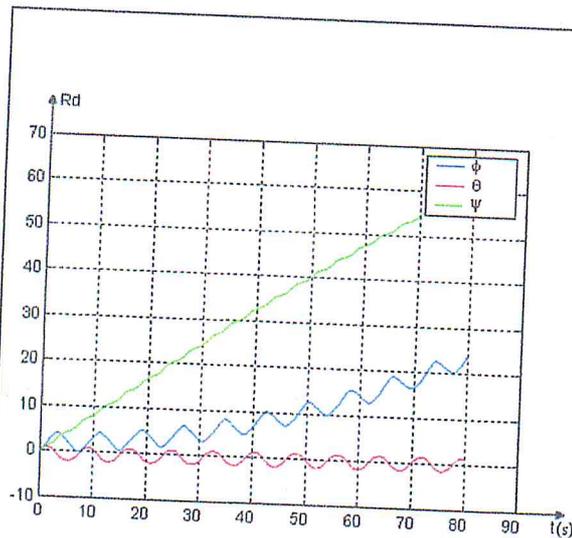
Cette fonction représentera l'énergie mécanique totale à fournir au système, pour obtenir l'attitude désirée. elle mesure ' la consommation ' du système pour une commande donnée. Cette dernière devant être optimisée par minimisation répond a l'architecture analytique suivante :

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

IV.8 résultats et interprétations pour mini satellite

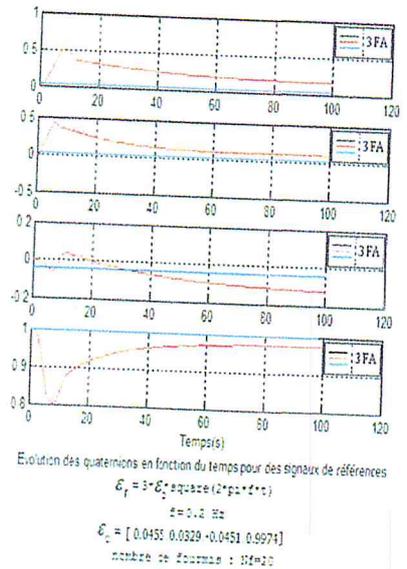
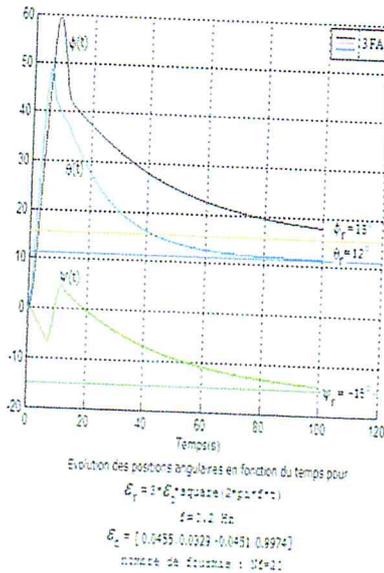
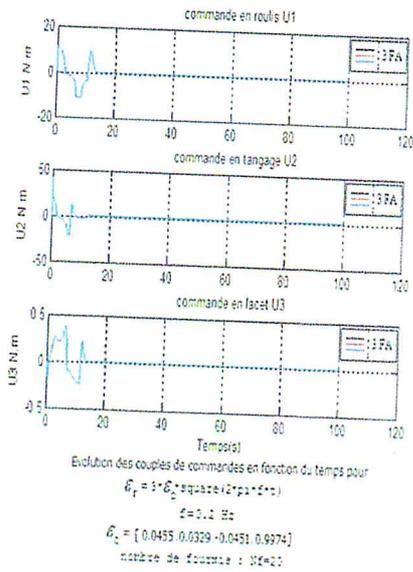
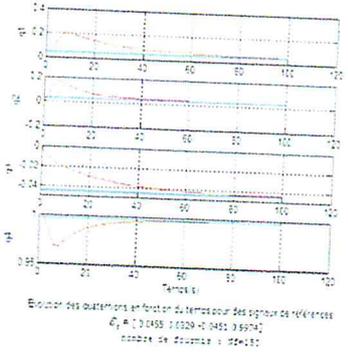
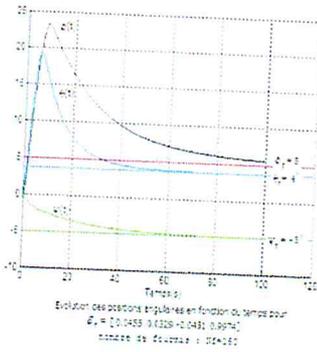
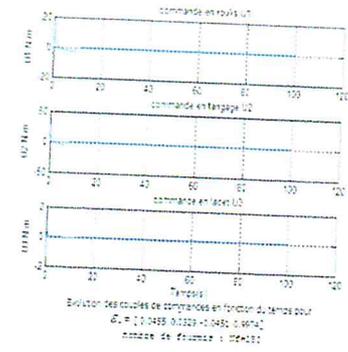
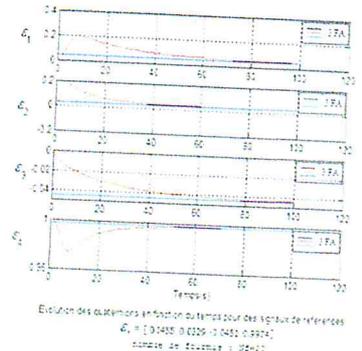
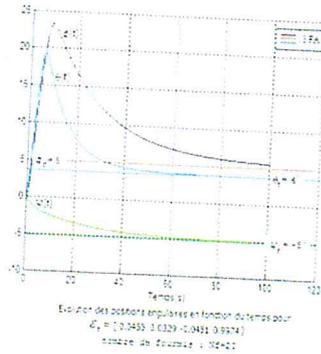
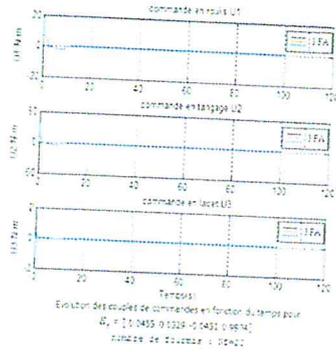
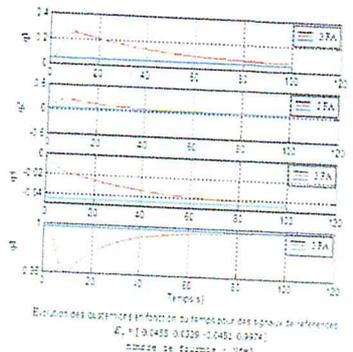
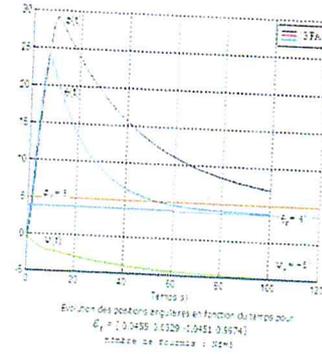
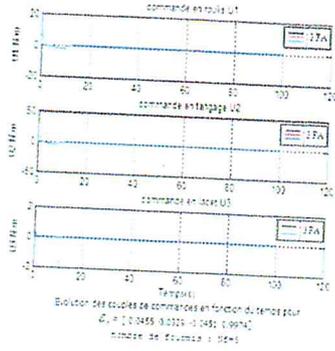
• Les paramètres de simulations pour Mini satellite sont :
 $I_R=132 \text{ Kg.m}^2$; $I_T=137 \text{ Kg.m}^2$; $I_L=7 \text{ Kg.m}^2$; orbite circulaire ; masse de satellite d'environ $m_S=150 \text{ Kg}$.

Mouvement libre



La simulation du mouvement libre du mini satellite montre que la variable angulaire φ évolue de manière croissante selon une loi sinusoïdale de la même manière que θ , sauf que cette dernière est stable, à amplitude fixe. Tandis que l'angle ψ oscille à faible amplitude suivant une droite croissante.

Figure IV.14 : Cas du mini satellite



ملخص

تقدير الموقف (أو التوجه) في مساحة 3-الأبعاد هو موضوع الذي يدفع المجتمع ، مع تطبيقات مختلفة مثل السيطرة في (S)العلمي منذ عقود (على الأقل منذ منتصف 60 ، المواقع الأقمار الصناعية والطائرات والروبوتات الجوي وتحت الماء، والتطبيقات المتعلقة بالصحة (مراقبة الناس في بيئتهم، وإعادة التأهيل وإعادة التأهيل) أو حتى الانفجار من التطبيقات المتعلقة الأفلام والألعاب المتحركة. هذه المناطق الماضيين (الصحة والمرح) بحاجة إلى تصميم أنظمة يمكن الاعتماد عليها، وتوفير تقديرا للموقف (وأحيانا موقف) "الدقيق"، والتعاقد، والمحمولة ومستقل، لذلك كفاءة في استخدام الطاقة. وقد أدت هذه القيود الباحثين والمصممين لهذه الأنظمة الأجهزة المستندة إلى الاستشعار (مايكرو MEMS التقاط المواقع أن أنتقل إلى تقليديا triaxial الكهربائية نظام ميكانيكي)، وجهاز دخلت الآن مرحلة التسارع (أي 3 التسارع شنت على ثلاثة محاور متعامدة بعضها بعضا، مع حساسية مماثلة من triaxial والدوران triaxial أجهزة الاستشعار بعد كل محور القياس)، والمغناطيسية ، في إشارة تتعلق الاستشعار، والجاذبية والتسارع triaxial التدابير التسارع ، يتيسر المجال Triaxial الخاص هيئة جامدة الذي تعلق أجهزة الاستشعار. المغناطيسية سرعة دوران triaxial المغناطيسي الأرضي والاضطرابات المغناطيسية، ويقيس جيروسكوب جسم صلب. انصهار كل هذه البيانات يمكن ارجاعه الى موقف هيئة جامدة في الفضاء 3 الابعاد. الغرض من مشروعنا هو سيطرة خوارزمية مستعمرة النمل. والميزة الرئيسية الصناعية ليكون الأمثل من خلال خوارزمية مستعمرة النمل. مما يجعل فرضية عن استخدام هذه الأدوات في سياق تصميم نظام التحكم في الحركة هو أن معرفة خصائص أجهزة الاستشعار المستخدمة يمكن ضمان مجموعة من الحلول، وبالتالي، فمن الضوضاء معروف ولكن المحدود اللاخطية أو الاضطرابات الخارجية. ويتم تحقيق أداء الناتج الممكن للتحقق من ذلك، لتكوين نظرا لنمذجة النظام، وسيتم تحقيق أداء الناتج (أي دقة من الخوارزميات المستخدمة) ضمان ذلك.

Summary

The estimate of the attitude (or orientation) in a 3-dimensional space is a topic that drives the scientific community for decades (since at least the mid-60s), with applications as varied as controlling in attitude satellites and aircraft, aerial robotics and underwater, health related applications (monitoring of people in their environment, re-education and rehabilitation) or even the explosion of applications related to films and animated games. These last two areas (health and fun) need to design reliable systems, providing an estimate of the attitude (and sometimes position) "precise", compact, portable and autonomous, so energy efficient. These constraints have led researchers and designers of these attitudes capture systems to turn to MEMS sensor-based devices (Micro Electro Mechanical System), the device now embarking traditionally triaxial accelerometer (ie 3 accelerometers mounted on three mutually orthogonal axes, with a similar sensitivity of the sensor after each measurement axis), a triaxial magnetometer and a triaxial gyro. The triaxial accelerometer measures, in the reference related to the sensor, gravity and acceleration own rigid body on which the sensors are attached. Triaxial magnetometer measures the Earth's magnetic field and the magnetic disturbances, and triaxial gyroscope measures the speed of rotation of the rigid body. The fusion of all these data can be traced back to the attitude of the rigid body in 3-dimensional space. The purpose of our project is the control by fuzzy controller orientation (or attitude) of a satellite to be optimized by ant colony algorithm. The main advantage of using these tools in the context of designing a motion control system is that knowing the characteristics of the sensors used can be guaranteed a set of solutions, making an assumption about the unknown but limited noise of nonlinearities or external disturbances. Thus, it is possible to verify that, for a given configuration of a modeling of the system, the output performance (i.e. the accuracy of the algorithms used) will be achieved so guaranteed.

Résumé

L'estimation de l'attitude (ou orientation) dans un espace à 3 dimensions est un sujet qui anime la communauté scientifique depuis plusieurs décennies (depuis au moins le milieu des années 60), avec des applications aussi variées que le contrôle en attitude des satellites et des aéronefs, la robotique aérienne et sous-marine, les applications liées à la santé (suivi des personnes dans leur environnement, rééducation et réhabilitation) ou bien encore l'explosion des applications liées aux films et aux jeux d'animation. Ces deux derniers domaines (santé et amusement) nécessitent de concevoir des systèmes fiables, fournissant une estimation de l'attitude (et parfois de la position) "précise", peu encombrant, portables et autonomes, donc peu gourmands en énergie. Ces contraintes

- [1] ABERKANE, AIT OUAZZOU 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
- [23] <http://fr.wikipedia.org/wiki/Optimisation>

Bibliographie

[24] http://artemmis.univ-mrs.fr/cybermeca/Formcont/mecaspa/SCAO/IND_SCAO.htm

[25] http://spot4.cnes.fr/spot4_fr/attitude.htm