

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

Ministère de l'enseignement supérieur et de la recherche scientifique



**UNIVERSITE AMAR TELIDJI
LAGHOUAT**



FACULTE DES SCIENCES ET DE L'INGENIERIE

DEPARTEMENT DE GENIE INFORMATIQUE

**PROJET DE FIN D'ETUDES
POUR L'OBTENTION DU DIPLÔME**

D'INGENIEUR D'ETAT EN INFORMATIQUE

Option : Systèmes parallèles et distribués

Thème

**Optimisation d'un
contrôleur flou par les algorithmes génétiques.**

**Application : suivi d'une trajectoire par
un robot mobile.**

Réalisé par:

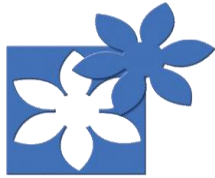
SEGHIER Amina

SOUIDA Kheira

Encadré par:

M^{lle} TAABA Kheira

N° d'ordre: /2008-PFE/DGI



REMERCIEMENTS

Nous aimerions remercier en premier lieu Dieu, le Tout Miséricordieux, qui nous a donné la force et la volonté d'achever ce projet.

Un grand remerciement à M^{elle} TAABA Khiera, notre directrice de thèse pour son soutien tout au long de la préparation de cette thèse ainsi que pour sa patience et son encouragement.

Nos remerciements s'adressent aux membres du Jury qui nous font l'honneur de participer à la soutenance.

Nos plus vifs remerciements à nos parents, notre famille, et nos amies.

Enfin, nous remercions toutes les personnes qui nous ont aidé et encouragé de près ou de loin, à la réalisation de ce travail et au long de notre parcours universitaire.

RÉSUMÉ

Dans ce travail, nous présentons une réalisation d'un contrôleur flou optimisé par les Algorithmes Génétiques. Pour le contrôleur flou, les fonctions d'appartenance, règles floues et défuzzification sont bien définies ; on s'intéresse pour l'optimisation sur la base des règles floues. Nous définissons aussi les opérations génétiques de sélection, croisement et de mutation utilisées dans l'Algorithme Génétique.

On applique cette technique de commande au robot mobile tricycle. Le contrôleur flou ajuste les paramètres d'un Algorithme Génétique utilisé pour estimer la position angulaire d'un robot mobile tricycle pour suivre la trajectoire définie.

Les entrées du contrôleur flou sont la distance entre le robot et l'objectif et l'angle d'orientation de ce dernier, la sortie est la variation de l'angle d'orientation.

Mots clés : Logique floue, contrôleur flou, Algorithme Génétique, Optimisation, Règle floue, robot mobile.

ABSTRACT

In this work, we present a realization of a fuzzy controller optimized by genetic algorithms. For the fuzzy controller, the membership functions, fuzzy rules, defuzzification are well defined and we focus on optimization of rules base. We also define the genetic operations of selection, crossing and of mutation used in the Genetic algorithm. We apply this technique to command mobile robot tricycle. The fuzzy controller adjusts the parameters of a Genetic Algorithm used to estimate the angular position of a robot mobile tricycle for follow-up the definite trajectory. The inputs of the fuzzy controller are the distance between the robot and the objective and the angle of orientation of this last, the output is the variation of the orientation angle.

Key words: Fuzzy logic, Optimization, Genetic algorithm, fuzzy rules, tricycle robot, Fuzzy Controller.

ملخص

في هذا العمل تطرقنا لدراسة التحكم بواسطة المنطق الضبابي لإعطاء المركبة المتحركة الإحداثيات اللازمة من أجل إتباع مسار المحدد مسبقا و من أجل إيجاد أحسن مسار لإتباع المسار المحدد مسبقا استخدمنا أسلوب من أساليب التحسين وهي الخوارزميات الجينية بحيث يصبح من الممكن استخراج المعرفة وتحسين القواعد الضبابية، وطبقناها على المتحكم بالمنطق الضبابي. ولمعرفة مدى صلاحية هذه التقنية قمنا بتطبيقها على مركبة ثلاثية العجلات عادية الشكل.

مفتاح الكلمات : المنطق الضبابي , المركبة المتحركة , أساليب التحسين, الخوارزميات الجينية , مركبة آلية

Liste des Figures

No table of contents entries found.Figure I-7 : fonction d'appartenance
singularise.....11

Figure I-8 : fonction d'appartenance des ensembles flous A et leur complément A^c 12

Figure I-9 : l'union de deux ensembles flous.....13

Figure I-10 : l'intersection de deux ensembles flous.....13

Figure I-11 : Recouvrement des ensembles flous.....14

Figure I-12 : Fonctions d'appartenance associées aux différents nombres flous Permettant de
décrire la variable linguistique température.....16

Figure I-13 : structure d'un régulateur flou.....21

Figure I-14 : fuzzification singleton.....21

Figure I-15 : fuzzification non- singleton.....22

Figure III-1 : structure d'un contrôleur flou.....25

Figure III-2 : les fonctions d'appartenances des entrées de l'erreur de la
position et l'erreur de l'angle25

Figure III-3 : fonction d'appartenance pour la sortie $\varphi(^{\circ})$ l'orientation.....25

Figure III-5 : principe de Les équations de mouvement.....25

Figure IV -1 : la fenêtre principale du logiciel.....26

Figure IV-2 : Menus du logiciel.....26

Figure IV-3 : l'initialisation de la commande floue.....26

Figure IV-4 : la simulation par la commande floue40

No table of contents entries found.

Sommaire

Remerciements

Résumé

Introduction générale1

Chapitre I : La Théorie des Ensembles flous et de la logique floue

I.1. Introduction	4
I.2. Aperçus de la logique floue	4
I.3 Théorie des ensembles flous	5
I.3.1 Concept de l'ensemble flou	5
I.3.2 Les ensembles flous	6
I.3.2.1 Similarités entre la logique classique et la logique floue.....	6
I.3.2.2 Définition d'un ensemble flou.....	7
I.3.2.3 Caractéristiques des ensembles flous.....	8
I.3.2.4 Représentation des ensembles flous	9
I.3.2.5 Opérations sur les ensembles flous.....	12
I.3.4 Recouvrement de deux ou plusieurs ensembles flous.....	14
I.3.5 Relation et composition floue	15
I.3.5.1 Relation floue	15
I.3.5.2 Composition des relations floues.....	15
I.4 Logique floue	16
I.4.1 les éléments de bases de la logique floue.....	16
I.4.2 Variables linguistiques.....	16
I.4.3 Opérateurs de la logique floue.....	17
I. 4.4 L'implication floue	18
I.4.5 Les règles floues	19
I. 4.5.1 Règle de Mamdani.....	20

I. 4.5.2 Règle de Takagi-Sugeno (TS).....	20
I.4.6 Le raisonnement et la prise de décision.....	20
I.4.7 Commande et réglage par la logique floue	21
I.4.7.1 Structure d'un contrôleur flou	21
I.4.7.1.1 La fuzzification.....	22
I.4.7.1.2 La base de connaissances	23
I.4.7.1.3 Le moteur d'inférence flou	24
I.4.7.1.4 La défuzzification	25
I.5 Types de contrôleur flou.....	26
I.5.1 Contrôleurs flou de type Mamdani.....	26
I.5.2 Contrôleurs flous de type Sugeno.....	26
I.6 conclusion.....	27

Chapitre II : les algorithmes génétiques

II.1 Introduction.....	28
II.1.1 Evolution naturelle et algorithmes génétiques	28
II.1.2 Terminologie	29
II.2 Principe de base d'un AG standard	31
II.3 Les operateurs génétiques simples	34
II.3.1 La sélection	34
II.3.2 Croisement.....	35
II.3.2 .a Croisement à un point.....	35
II.3.2 .b Croisement en deux points.....	36
II.3.2 .c Croisement aléatoire.....	36
II.3.3 Mutation.....	37
II.4 le codage.....	38
II.4.1 Le codage binaire	38
II.4.2 Le codage réel.....	38

II.5 Conclusion	39
Chapitre III : Optimisation d'un contrôleur flou par Algorithme Génétique	
III.1 introduction	40
III.2 structure du contrôleur flou	40
III.3 Optimisation par les algorithmes génétiques.....	44
III.3.1 Optimisation des règles floues par un Algorithme Génétique	46
A. Codage	47
B. Sélection	47
C. Croisement.....	48
D. Mutation	49
III.4 Modèle géométrique du robot tricycle	50
III.5 conclusion	52
Chapitre IV : Présentation de simulation	
IV.1 Introduction	53
IV.2 Environnement du travail.....	53
IV.3 Présentation du logiciel	53
IV.3.1 Menus du logiciel	53
IV.3.1.1 Le menu simulation par la commande floue	54
IV.3.1.2 le menu Optimisation pour les algorithmes génétiques	56
IV.3.2 exemple d'application.....	59
IV.4 Conclusion	64
Conclusion générale	
Annexe A	67
Annexe B.....	72
Bibliographie.....	79

Liste des tableaux

Tableau I-1	: Correspondances entre concepts en théories d'ensembles et en logiques	7
Tableau I-2	: Principales implications floues	18
Tableau I-3	: désignation standard des ensembles flous	23
Tableau I-4	: matrice d'inférences pour deux variables linguistiques.....	23
Tableau III-1	: la base de règles de contrôleur flou.....	43

INTRODUCTION GÉNÉRALE

De nos jours, des systèmes de régulation automatique du fonctionnement des processus sont intégrés dans de nombreuses applications, tant dans le domaine scientifique que technologique. Ces systèmes deviennent de plus en plus complexes, les performances des régulateurs utilisés ne cessent de s'améliorer.

Lorsque le système est fortement non-linéaire, imprécis ou très complexe, il est parfois impossible de définir un modèle mathématique de son fonctionnement. Du coup, le défi principal est de formuler une logique qui ne nécessite pas de connaître le fonctionnement exact du système, mais simplement la façon de le commander. Le principe de régulation, basé sur des concepts relativement simples, permet de faire intervenir dans le contrôleur des connaissances acquises par un expert humain. Les connaissances sont exploitées sous une forme linguistique par l'intermédiaire de règles comme « si condition alors action ». Une méthode de raisonnement (inférence) utilise ces règles pour définir les commandes qui sont envoyées au système. C'est dans ce contexte que la logique floue a été fondée.

Les applications industrielles de la logique floue ont longtemps été restreintes à la commande de petits systèmes qu'il était possible de régler empiriquement. Il s'agissait typiquement de régulateurs flous à deux entrées : l'erreur par rapport à une consigne et sa variation. La table de règles établie par des « experts ». Il existe en effet un assez grand nombre d'implémentations différentes nécessitant à chaque fois un traitement particulier. Deux types de SIF (système d'inférence flou) sont majoritairement utilisés : Mamdani et de Takagi-Sugeno.

Les règles floues représentent une connaissance humaine, exprimée en langage naturel, à l'aide de mots vagues, mal définis, « flous ». [Glo 99]

Lorsque les règles de commande ne sont pas connues précisément, la conception du contrôleur flou peut être assimilée à un problème d'optimisation. De nombreuses techniques d'optimisation ont été décrites dans la littérature (méthodes linéaires, programmation dynamique . . . etc). Ces techniques ont trouvé des applications dans plusieurs domaines scientifiques. Dans notre contexte, nous avons choisi d'utiliser les algorithmes génétiques afin d'optimiser la base des règles floues du contrôleur flou.

L'algorithme génétique est une méthode d'exploration de l'espace des solutions possibles d'un problème exploitant des mécanismes similaires à ceux de la sélection naturelle.

A chaque itération, l'algorithme génétique évalue un ensemble limité de solutions du problème, et retient les meilleures d'entre elles (phase de sélection). Ces solutions sont combinées (phase de croisement) et éventuellement légèrement modifiées (phase de mutation) avant d'être à nouveau évaluées. A l'issue de ce processus itératif, on aboutit sous certaines conditions à la solution optimale du problème.

La logique floue, les réseaux de neurones et les algorithmes génétiques constituent une voie prometteuse pour aborder le problème de régulation. Ces méthodes font intervenir des mécanismes qui ont été observés et étudiés dans des domaines de recherche initialement très éloignés de l'informatique : linguistique et raisonnement humain pour la logique floue, physiologie humaine et animale pour les réseaux de neurones ou encore génétique et sélection naturelle pour les algorithmes génétiques /évolutionnistes.

Elles sont pour la plupart basées sur un apprentissage qui permet de définir de façon itérative le meilleur jeu de paramètres pour une structure donnée de contrôleur. Pour l'instant, les chercheurs se sont penchés principalement sur les approches suivantes :

- Optimisation des fonctions d'appartenance,
- Optimisation des règles floues,
- Optimisation simultanée des fonctions d'appartenance et des règles floues.

Dans ce cadre, nous présentons notre travail qui a pour objectif la réalisation d'un logiciel du contrôleur flou optimisé par les algorithmes génétiques. Cette thèse est organisée en quatre chapitres:

Dans le premier chapitre, consacré à la commande floue, nous commençons par énoncer les fondements de la logique floue. Nous voyons comment elle permet d'exprimer selon un formalisme unique des informations très diverses (données incertaines ou imprécises, connaissances exprimées sous forme linguistique, . . . etc). Ensuite, nous présentons en détails les méthodes de raisonnement flou (propositions, implications et inférence) qui constituent la base de la commande floue. Enfin, nous décrivons la structure générale d'un contrôleur flou.

Dans le deuxième chapitre, nous décrivons le principe des algorithmes génétiques. Sur la base d'un exemple simple, nous analysons l'intérêt des différentes phases de traitement :

sélection des individus, croisement et mutation. Nous exposons ensuite le théorème des schémas, qui explique de façon théorique le bien-fondé de cette méthode d'optimisation.

Nous décrivons enfin plusieurs méthodes de représentation des informations traitées par un algorithme génétique sous la forme d'un codage binaire, ou par des nombres réels ou en base n.

Le troisième chapitre est consacré à la description d'un contrôleur flou par un algorithme génétique. Nous présentons les différentes étapes de la logique floue (fuzzification ; inférence floue ; défuzzification) et même pour les algorithmes génétiques (sélection ; croisement ; mutation). On applique ce contrôleur sur un robot mobile tricycle.

Enfin, dans le quatrième chapitre, on représente le logiciel qui a été réalisé et implémenté en Delphi. Les résultats seront discutés et illustrés par des exemples.

CHAPITRE I



*La Théorie des Ensembles flous
et de la logique floue*

1.1. Introduction :

De nos jours, la logique floue (en anglais « fuzzy logic ») est un axe de recherche important sur lequel se focalisent de nombreux scientifiques. Elle trouve des applications dans de multiples domaines, tant public (appareils photos, machines à laver, fours à micro-onde,...), que industriel (réglage et commande de processus complexes liés à l'énergie, aux transports, à la transformation de la matière, à la robotique, aux machines-outils,...).

Les bases théoriques de la logique floue ont été formulées en 1965 par le professeur Lotfi Asker Zadeh d'origine iranienne, actuellement à l'Université de Berkeley en Californie. Il a introduit la notion de sous-ensemble flou pour fournir un moyen de représentation et de manipulation des connaissances imparfaitement décrites, vagues ou imprécises. A cette époque, la théorie de la logique floue n'a pas été prise au sérieux (à l'exception de quelques experts).

A partir de 1985, ce sont les Japonais qui ont commencé à utiliser largement la logique floue pour des applications industrielles afin de résoudre des problèmes de réglage et de commande.

1.2. Aperçus de la logique floue :

Le terme de logique floue a deux acceptions. Au sens strict (acception la plus généralement utilisée et à laquelle nous nous référons ici), il s'agit d'une logique formelle du raisonnement approximatif basée sur la théorie des ensembles flous. Au sens général, le terme de logique floue est synonyme à celui de théorie des ensembles flous. La logique floue et la théorie des ensembles flous ont été introduites principalement par Lotfi Asker Zadeh. Ses concepts reposent sur des notions développées antérieurement en philosophie par Jan Lukasiewicz (logique à valeurs multiples) et Max Black (description d'opérations sur des ensembles flous). La logique floue généralise la logique classique avec des variables logiques et des formules logiques prenant des degrés de valeur de vérité quelconques entre 0 (faux) et 1 (vrai) inclusivement: 0 correspond à totalement faux, 1 correspond à totalement vrai, et les valeurs entre 0 et 1 exclusivement,]0,1[, correspondent aux degrés de vérité floue entre totalement faux et totalement vrai.

La logique classique avec ses valeurs de vérité booléennes de 0 et 1 est considérée comme un cas particulier de la logique floue [**Lab 07**].

I.3 Théorie des ensembles flous :

La théorie des ensembles flous est une théorie mathématique du domaine de l’algèbre abstraite. Elle a été développée afin de représenter mathématiquement l'imprécision relative à certaines classes d'objets et sert de fondement à la logique floue.

I.3.1 Concept de l'ensemble flou

Le concept de l’ensemble flou a été introduit pour éviter les passages brusques d’une classe à une autre et autoriser des éléments à n’appartenir complètement ni à l’une ni à l’autre, ou encore à appartenir à chacune. [BMS 01]

Afin de mettre en évidence le principe fondamental de la logique floue, on présente un exemple simple de la classification des personnes en trois ensembles «jeune», «entre deux âges», «âgé». Selon la logique classique (logique de booléenne binaire), qui n'admet pour les variables que les deux valeurs 0 et 1, toutes les personnes âgées de moins de 25 ans sont alors considérées des jeunes et toutes les personnes âgées de plus de 50 ans comme des vieux (figure (I-1)).

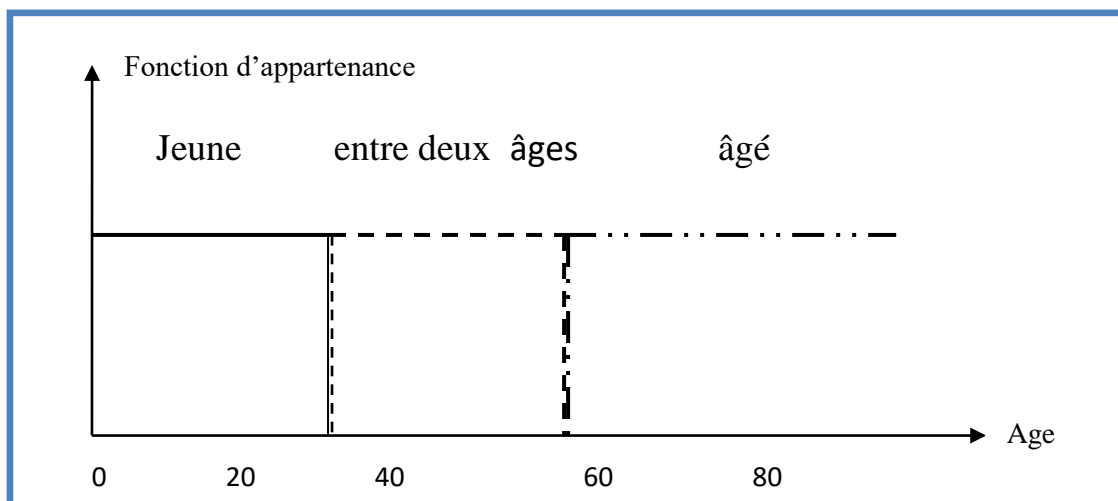


Figure I-1: Classification des personnes en trois ensembles selon

la logique classique

Cependant, une telle logique de classification ne tient pas en compte de la complexité et de la diversité du problème. Pourquoi une personne, lorsqu'elle a eu 50 ans, doit-elle être considérée comme appartenant à l'ensemble âgé? En réalité, un tel passage se fait progressivement et individuellement.

La logique floue, dont les variables peuvent prendre n'importe quelle valeur entre 0 et 1,

permet de tenir compte de cette réalité. Les limites ne varient pas soudainement, mais progressivement. La figure (I-2) montre une classification possible pour l'exemple précédent, cette fois-ci à l'aide de la logique floue. Ainsi une personne de 25 ans appartient à l'ensemble «jeune» avec une valeur $\mu=0.75$ de la fonction d'appartenance et à l'ensemble «entre deux âges» avec $\mu=0.25$. Par contre une personne âgée de 65 ans appartient avec une valeur $\mu=1$ de la fonction d'appartenance à l'ensemble «âgé».

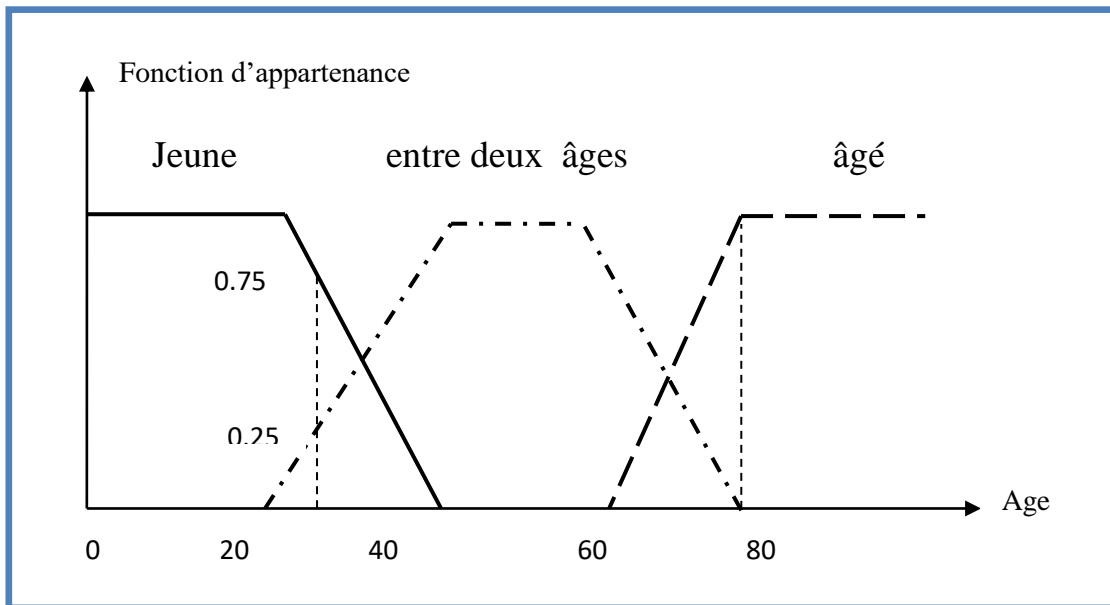


Figure I-2: Classification des personnes en trois ensembles selon la logique floue

I.3.2 Les ensembles flous :

Les ensembles flous, développés par Lotfi Zadeh, ont été introduits afin de modéliser la représentation humaine des connaissances, et ainsi améliorer les performances des systèmes de décision qui utilisent cette modélisation. Les ensembles flous sont utilisés soit pour modéliser l'incertitude et l'imprécision, soit pour représenter des informations précises sous forme lexicale assimilable par un système expert.

I.3.2.1 Similarités entre la logique classique et la logique floue:

La logique classique et la logique floue sont formalisables dans le cadre de théories d'ensembles: la théorie des ensembles classiques permet de formaliser la logique classique et la théorie des ensembles flous permet de formaliser la logique floue.

La théorie des ensembles flous et la théorie des ensembles classiques sont basées sur des concepts de base similaires et qui peuvent être mis en correspondance avec les concepts utilisés par la logique floue et la logique classique (voir tableau suivant) : **[Lab 07]**

Quelques concepts standards utilisés en théories des ensembles (classiques ou flous)	Concepts correspondants utilisés en logique (classique et floue)
\cup (union)	\vee (disjonction)
\cap (intersection)	\wedge (conjonction)
$-$ (différence)	\neg (négation)
\subseteq (inclusion)	\Rightarrow (implication)
X (ensemble universel)	1 (vrai)
\emptyset (ensemble vide)	0 (faux)

Tableau I-1: Correspondances entre concepts en théories d'ensembles et en logiques

I.3.2.2 Définition d'un ensemble flou :

Un ensemble classique A de X est défini par une fonction caractéristique μ_A qui prend la valeur 0 pour les éléments de X qui n'appartiennent pas à A et la valeur 1 pour ceux qui appartiennent à A . $\mu_A : \mu_A(x) \rightarrow [0,1]$

Définition:

Un ensemble flou A de X est défini par une fonction d'appartenance qui associe à chaque élément x de X , de degré $\mu_A(x)$ compris entre 0 et 1, avec lequel x appartient à A :

$$x : \mu_A(x) \rightarrow [0,1] \dots\dots\dots(I-1)$$

Dans le cas particulier où μ_A ne prend que des valeurs égales à 0 ou 1, l'ensemble A est un ensemble classique de X . Un ensemble classique est donc un cas particulier d'un ensemble flou.

Pour illustrer le cas ensemble flou, considérons l'ensemble des fréquences variant des plus basses fréquences aux plus hautes fréquences. Soient les deux sous-ensembles A et A' représentant les basses fréquences et les moyennes fréquences, respectivement. Une fréquence égale à 80 Hz peut appartenir aux deux ensembles A et A' avec différents degrés d'appartenance (figure I-3). Dans ce contexte, la notion du flou est apparente, puisque les termes « basses » et « moyennes fréquences » sont imprécises (pas nets). [BMS 01]

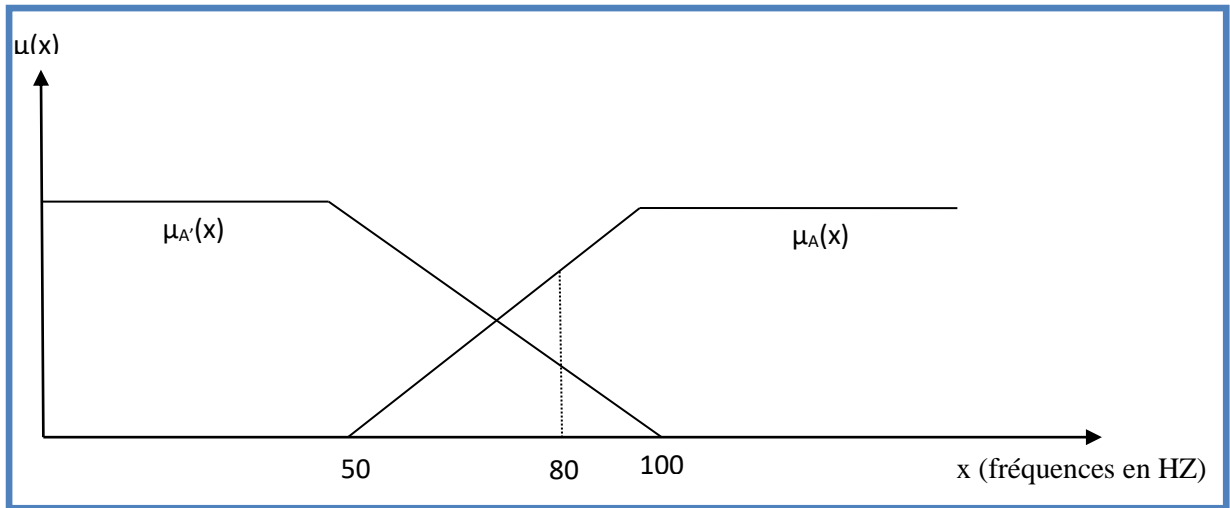


Figure I-3 : Exemple d'ensemble flou.

Un ensemble flou A dans X peut être représenté comme un ensemble de paires, constituées de l'élément x et de la valeur de sa fonction d'appartenance :

$$A = \{(x, \mu_A(x)) / x \in X\} \dots\dots\dots(I-2)$$

Si X est continu, A est noté par:

$$A = \int_x \mu_A(x)/x \dots\dots\dots(I-3)$$

Si X est discret, A est noté par:

$$A = \sum_x \mu_A(x)/x \dots\dots\dots(I-4)$$

I.3.2.3 Caractéristiques des ensembles flous :

Les caractéristiques d'un ensemble flou A de X les plus utiles sont :

- Le support de A , noté $\text{supp}(A)$, est la partie de X sur laquelle la fonction d'appartenance de A n'est pas nulle :

$$\text{Supp}(A) = \{x \in X / \mu_A(x) \neq 0\} \dots\dots\dots(I-5)$$

Dans le cas où le support de l'ensemble flou A est un point unique dans X tel que $\mu_A(x) = 1$, A est appelé singleton flou.

Exemple : si $x = \{1, 2, 3, 4, 5, 6, 7\}$

$$A = 0.1 / 3 + 0.5 / 4 + 0.8 / 5 + 1 / 6$$

Donc $\text{Supp}(A) = \{3, 4, 5, 6\}$

- La hauteur, notée $h(A)$ de l'ensemble flou A de X est la plus grande valeur prise par sa fonction d'appartenance

$$h(A) = \text{Max } \mu_A(x) \dots\dots\dots(I-6)$$

- L'ensemble flou A de X est normalisé si sa hauteur $h(A)$ est égale à 1, et si $h(A) < 1$ A est dit sous normal.

L'opérateur $\text{norm}(A)$ dénote la normalisation de l'ensemble flou :

$$\hat{A} = \text{norm}(A) \Leftrightarrow \mu_{\hat{A}}(x) = \mu_A(x) / h(A) \quad \forall x \in X$$

Exemple : si $X = \{1, 2, 3\}$

$$A = 0.1 / 1 + 0.2 / 2 + 1 / 3$$

$$B = 0.1 / 1 + 0.6 / 2 + 0.9 / 3$$

Alors A est normal et B est sous normal.

- Le noyau de A , noté $\text{noy}(A)$, est l'ensemble des éléments de X pour lesquels la fonction d'appartenance de A vaut 1 :

$$\text{noy}(A) = \{x \in X / \mu_A(x) = 1\} \dots\dots\dots(I-7)$$

- La cardinalité de l'ensemble flou A de X est définie par:

$$\text{Card}(A) = |A| = \sum_{x \in X} \mu_A(x) \dots\dots\dots(I-8)$$

Exemple : Si $A = 1/X1 + 0.8/X2 + 0.6/X3 + 0.2/X4 + 0/X5$

$$\text{Alors Card}(A) = 1 + 0.8 + 0.6 + 0.2 = 2.6$$

Si A est un ensemble ordinaire de X , sa hauteur est égale à 1 ; avec un noyau égal à son support, sa cardinalité est le nombre d'éléments qui le composent.

I.3.2.4 Représentation des ensembles flous

Les fonctions d'appartenance

Une fonction d'appartenance est une courbe qui définit comment chaque point dans l'espace d'entrée est tracé à une valeur d'appartenance (ou degré d'appartenance) entre 0 et 1. Ces fonctions d'appartenance sont de différentes formes, parmi lesquelles nous citons **[BMS 01]**:

La fonction trapézoïdale :

Est caractérisée par quatre paramètres (a, b, c, d) :

$$\mu_A(x, a, b, c, d) = \begin{cases} 0 & x \leq a \\ \frac{x - a}{b - a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{d - x}{d - c} & c \leq x \leq d \end{cases} \dots\dots\dots(I-9)$$

$$\mu_A(x, a, b, c, d) = \max (0, \min ((x-a)/(x-b), 1,(d-x)/(d-c))).$$

Les paramètres a et d localisent les pieds du trapèze et les paramètres b et c localisent les sommets.(figure(I-4))

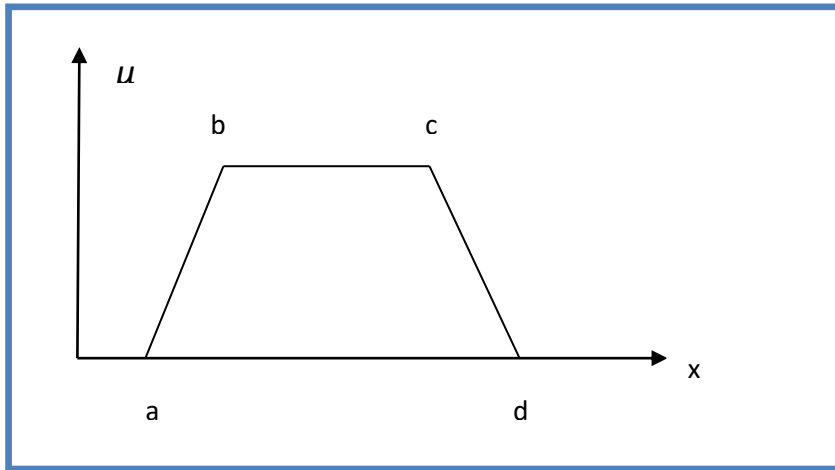


Figure 1-4 : fonction d'appartenance trapézoïdale

La fonction triangulaire :

Elle est caractérisée par trois paramètres (a, b, c) :

$$\mu_A(x, a, b, c) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & c \leq x \end{cases} \dots\dots\dots(I-10)$$

$$\mu_A (x, a, b, c) =\max (\min ((x-a/b-a), (c-x/c-b)), 0)$$

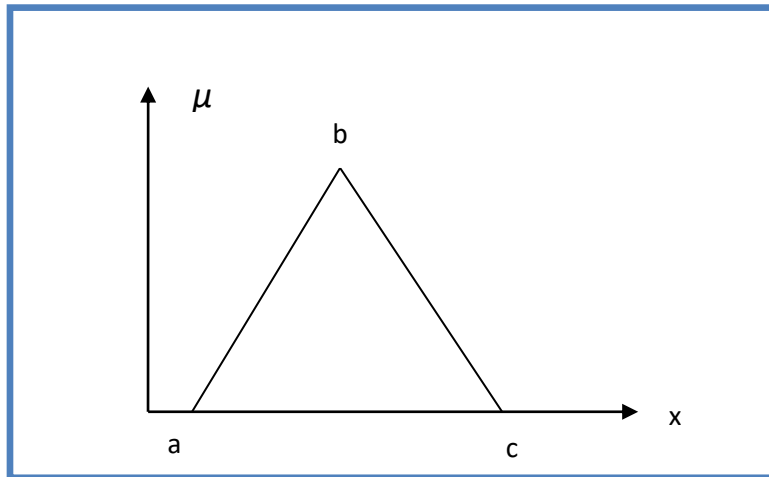


Figure 1-5 : fonction d'appartenance triangulaire

La fonction gaussienne :

Symétrique et elle dépend de deux paramètres c et σ , le centre et largeur de la fonction gaussienne respectivement :

$$\mu_A(x, c, \sigma) = \exp(-1/2 ((x-c) / \sigma)^2) \dots\dots\dots(I-11)$$

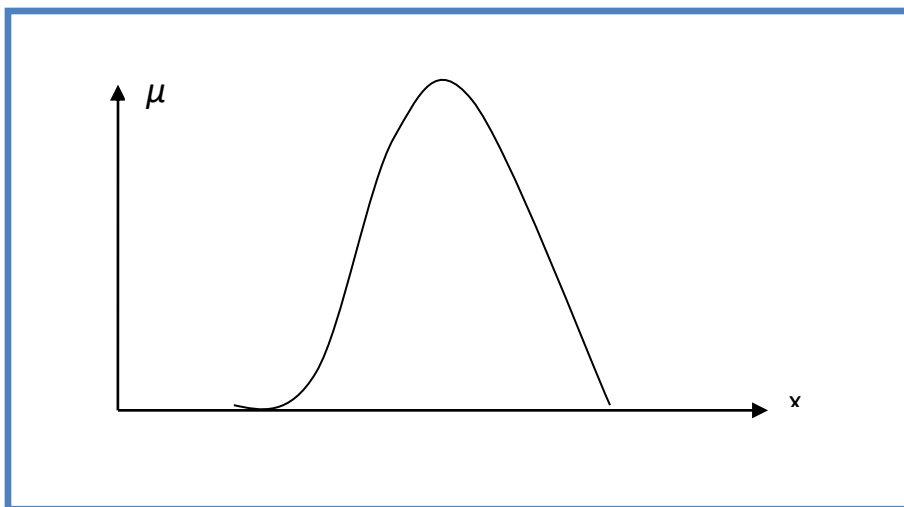


Figure 1-6 : fonction d'appartenance gaussienne

La fonction singularise :

$$\mu_A(x) = \begin{cases} 1 & x = x_0 \\ 0 & x \neq x_0 \end{cases} \dots\dots\dots(I-12)$$

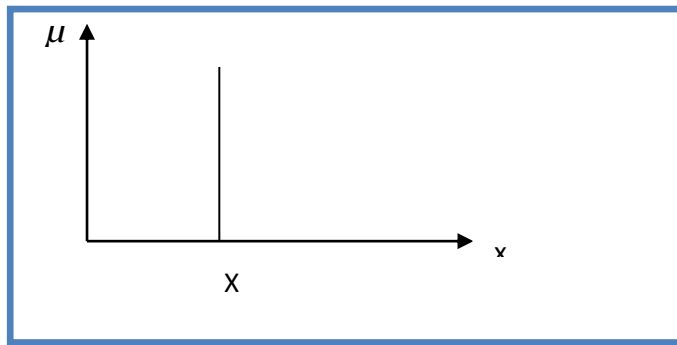


Figure 1-7 : fonction d'appartenance singularise

1.3.2.5 Opérations sur les ensembles flous :

Un ensemble flou est vide si et seulement si sa fonction d'appartenance est zéro sur X.

a) Egalité des ensembles flous :

Deux ensembles flous A et B sont égaux, $A = B$, si et seulement si leurs fonctions d'appartenance sont égales pour tout x dans X.

$$\forall x \in X \quad \mu_A(x) = \mu_B(x) \dots\dots\dots(I-13)$$

Exemple: on suppose $x = \{1, 2, 3\}$

$$A = 0.1 / 1 + 0.5 / 2 + 1 / 3$$

$$B = 0.2 / 1 + 0.5 / 2 + 1 / 3$$

$$C = 0.2 / 1 + 0.5 / 2 + 1 / 3$$

Donc : $B = C$ mais $A \neq C$ et $A \neq B$.

b) Complément d'un ensemble flou :

Le complément d'un ensemble flou A est dénoté par A^c et est défini comme l'ensemble flou de X de fonction d'appartenance :

$$\forall x \in X \quad \mu_{A^c}(x) = 1 - \mu_A(x) \dots\dots\dots(I-14)$$

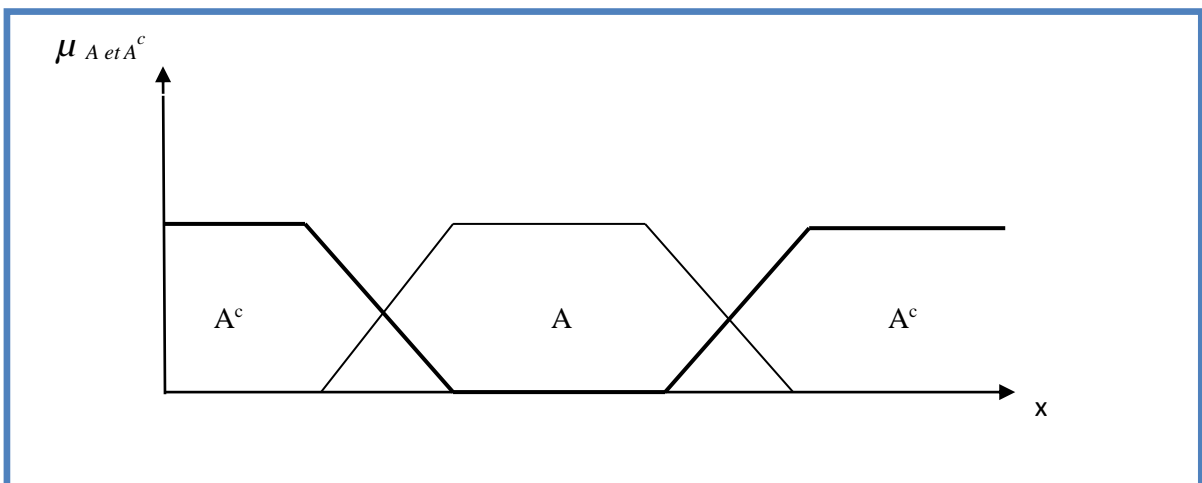


Figure I-8 : fonction d'appartenance des ensembles flous A et leur complément A^c

c) **Inclusion** : A est inclus dans B , si et seulement si : $\mu_A(x) \leq \mu_B(x)$

$$A \subset B \Leftrightarrow \mu_A(x) \leq \mu_B(x) \dots\dots\dots(I-15)$$

d) **Union des ensembles flous** :

L'union de deux ensembles flous A et B avec fonctions d'appartenances respectivement $\mu_A(x)$ et $\mu_B(x)$ est un ensemble flou C, écrit comme $C = A \cup B$ avec une fonction d'appartenance notée par :

$$\forall x \in X \quad \mu_C(x) = \max(\mu_A(x), \mu_B(x)) \dots\dots\dots(I-16)$$

Ou, dans une forme abrégée $\mu_C = \mu_A \vee \mu_B$

On note que U a la propriété associative, c'est-à-dire : $A \cup (B \cup C) = (A \cup B) \cup C$.

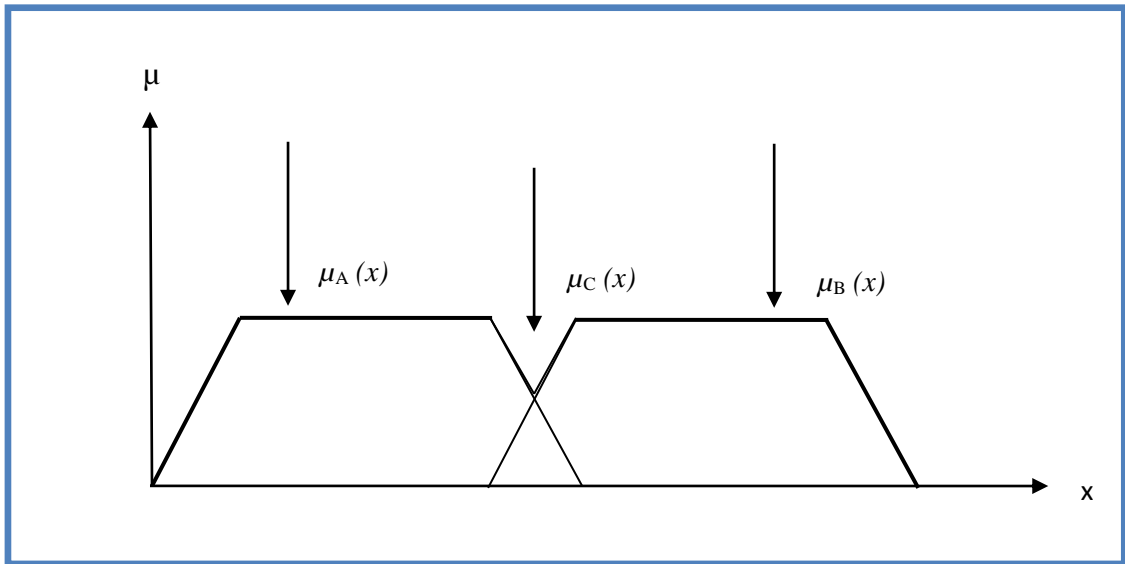


Figure I-9 : l'union de deux ensembles flous

e) **Intersection des ensembles flous** :

L'intersection de deux ensembles flous A et B avec fonctions d'appartenances respectivement $\mu_A(x)$ et $\mu_B(x)$ est un ensemble flou C, écrit comme $C = A \cap B$ avec une fonction d'appartenance notée par :

$$\forall x \in X \quad \mu_C(x) = \min(\mu_A(x), \mu_B(x)) \dots\dots\dots(I-17)$$

Ou, dans une notation abrégée

$$\mu_C = \mu_A \wedge \mu_B$$

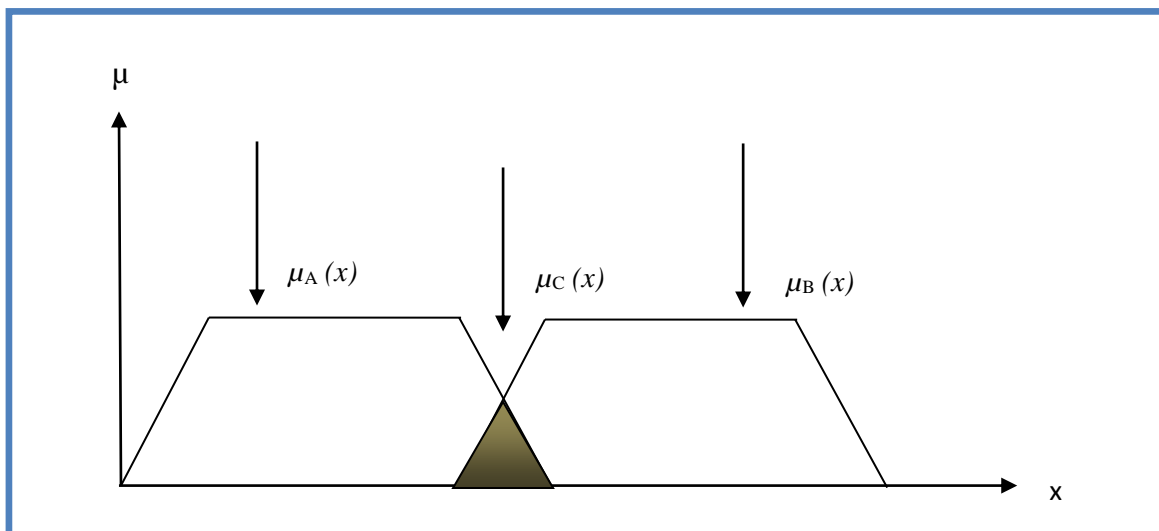


Figure I-10 : l'intersection de deux ensembles flous

a) **Produit cartésien flou:**

L'union et l'intersection sont définies pour des ensembles flous contenus dans le même univers de discours. Le produit cartésien flou est utilisé pour quantifier les opérations sur plusieurs univers de discours.

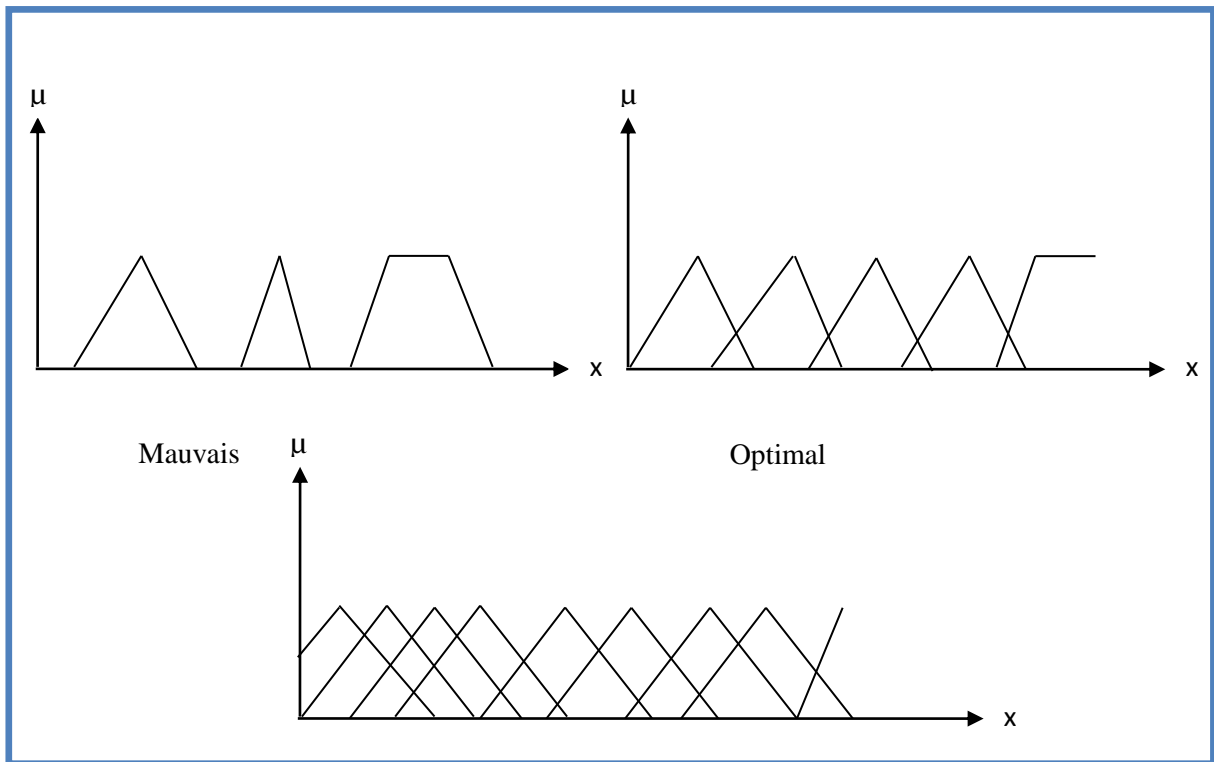
Soient A, A_2, \dots, A_n des ensembles flous définis sur les univers de discours X_1, X_2, \dots, X_n respectivement; leur produit cartésien est un ensemble flou (souvent appelé relation floue), noté par $A = A_1 \times A_2 \times \dots \times A_n$ avec une fonction d'appartenance définie par :

$$\mu_A(x) = \mu_{A_1}(x) * \mu_{A_2}(x) * \dots * \mu_{A_n}(x) \quad \dots \dots \dots (I-18)$$

I.3.4 Recouvrement de deux ou plusieurs ensembles flous: (chevauchement)

Plusieurs études ont montré que le recouvrement des ensembles flous influence le comportement du système de régulation. Le schéma ci-dessous donne trois exemples :

- Si les ensembles ne présentent absolument aucun recouvrement, le système de régulation en logique floue ne fonctionnera jamais correctement.
- Si les ensembles sont bien répartis, le système affichera le fonctionnement le plus stable.
- S'il existe un recouvrement entre plus de deux ensembles, il sera difficile d'énoncer correctement les règles.



Fonctionnement instable

Figure I-11 : Recouvrement des ensembles flous

I.3.5 Relation et composition floue :

I.3.5.1 Relation floue :

De même qu’une relation nette, une relation floue représente le degré de présence ou d’absence d’une association entre deux ou plusieurs ensembles flous [BMS 01].

Une relation $R(A,B)$, à deux ensembles A et B dans l’espace $A \times B$ est caractérisée par La fonction d’appartenance :

$$\mu_R(x,y) \text{ où } x \in A \text{ et } y \in B \dots\dots\dots(I-19)$$

$$\mu_R(x,y) \in [0,1]$$

I.3.5.2 Composition des relations floues :

La composition de deux relations floues $P(A,B)$ et $Q(B,C)$ est notée par :

$$R(A,C) = P(A,B) \circ Q(B,C), \text{ définie dans } A \times C.$$

La composition max-min des relations $P(A,B)$ et $Q(B,C)$ est définie par la fonction

d'appartenance $\mu_{P \circ Q}(x,z)$, telle que:

$$\mu_{P \circ Q}(x,z) = \max [\min (\mu_P(x,y), \mu_Q(y,z))] \dots\dots\dots(I-20)$$

La composition max-produit des relations $P(A,B)$ et $Q(B,C)$ est définie par la fonction

d'appartenance $\mu_{P \circ Q}(x,z)$, telle que:

$$\mu_{P \circ Q}(x, z) = \max [\mu_P(x, y), \mu_Q(y, z)] \dots\dots\dots(I-21)$$

I.4 Logique floue :

Les premières tentatives pour assouplir la logique classique ont concerné la manipulation de degrés de vérité autre que le vrai et le faux absolu. Une première extension de la logique classique à été proposée par Lukasiewicz comme logique à trois valeurs de vérité représentées par 0 (le faux), 1 (le vrai) et 1/2 (le doute). Après cela de nombreuses autres logiques multivalentes ont ensuite été proposées, pour des valeurs de vérités appartenant à l'intervalle [0,1], 0 correspond au faux absolu, 1 correspond au vrai absolu.

Mais de telles extensions de la logique classique ne suffisent pas pour traiter des connaissances imprécises, puisqu'elles n'admettent une gradualité que dans les valeurs de vérité et qu'elles n'autorisent pas une proposition à être énoncée d'une manière imprécise. C'est dans cette perspective que la logique floue a été élaborée.

I.4.1 les éléments de bases de la logique floue: [Koua 06]

Les éléments de base de la logique floue sont :

- Les variables linguistiques.
- Les inférences.
- Les opérateurs.

Dans la logique floue les algorithmes conventionnels sont remplacés par une série de règles linguistiques : Si, alors, ...Elle se prête donc très bien au réglage et à la commande de processus ainsi qu'au raisonnement humain.

I.4.2 Variables linguistiques: [Ham 07]

La description d'une certaine situation, d'un phénomène ou d'un procédé contient en général des expressions floues. Afin de permettre un traitement numérique, il est indispensable de les soumettre à une définition à l'aide de fonctions d'appartenance. Dans ce contexte, on attribue à chaque valeur de variable linguistique des fonctions d'appartenance dont la valeur varie entre 0 et 1. Le plus souvent, on utilise pour les fonctions d'appartenance des formes trapézoïdales ou triangulaires. Cependant parfois, on fait appel à d'autres formes telles que les formes de cloche et les formes obtenues à partir de fonctions trigonométriques. En toute généralité, la fonction d'appartenance est désignée par $\mu_A(x)$. L'argument "x" se rapporte à la variable linguistique. Pour une meilleure compréhension, prenant l'exemple de réglage de température " T " d'un local. Dans le cas le plus simple, on peut distinguer deux valeurs "froid" et "chaud" de la variable linguistique température. Elles forment deux ensembles

flous (Figure I-12). Ainsi une température de 16° appartient avec un facteur d'appartenance $\mu = 0,7$ à l'ensemble "froid" et avec $\mu = 0,3$ à l'ensemble "chaud".

On écrit alors $\mu_{\text{froid}}(T = 16^\circ) = 0,7$ et $\mu_{\text{chaud}}(T=16^\circ)=0,3$.

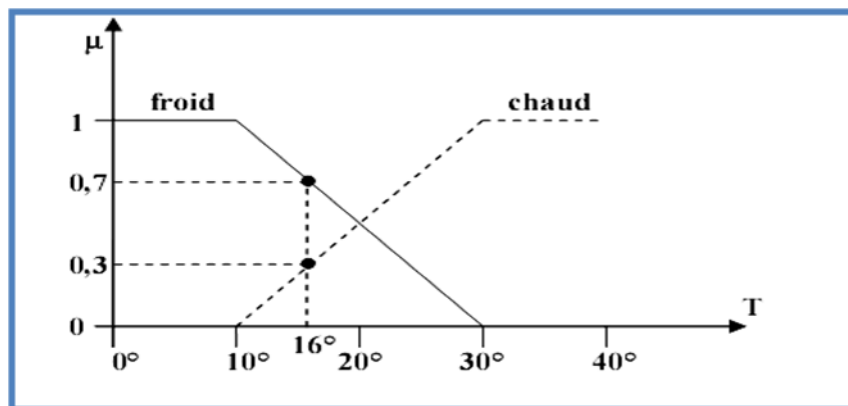


Figure I-12 : Fonctions d'appartenance associées aux différents nombres flous permettant de décrire la variable linguistique température

I.4.3 Opérateurs de la logique floue

Comme on l'a vu à la section précédente, les variables linguistiques sont liées entre elles au niveau des inférences par des opérateurs (*NON*, *ET*, *OU*). Il s'agit d'opérateurs de la logique floue qui interviennent sur les fonctions d'appartenances représentant les variables linguistiques.

Ces opérateurs ressemblent aux opérateurs de la théorie des ensembles flous :

Opérateur NON:

L'opérateur **NON** peut être exprimé par les fonctions d'appartenance de la manière suivante: $\mu_B(x)=1- \mu_A(x)$ (I-22)

A noter qu'il s'agit de l'opérateur **NON**, appelé aussi « complément », « négation » ou « inverse »

Opérateur ET:

Dans le cas de la logique floue, l'opérateur **ET** est réalisé dans la plupart des cas par la fonction du minimum. Il est appliqué aux fonctions d'appartenances de la manière suivante:

$$\mu_c(x) = \min[\mu_A(x), \mu_B(x)] \dots \dots \dots (I-23)$$

Opérateur OU:

La réalisation de l'opérateur **OU** au niveau de la logique floue se fait en général par la formation du maximum:

$$\mu_c(x) = \max[\mu_A(x), \mu_B(x)] \dots \dots \dots (I-24)$$

Dans la plupart des cas, ces opérateurs donnent des résultats convenables, surtout pour le réglage et la commande par logique floue.

I. 4.4 L'implication floue :

Considérons $p : \ll x \text{ est } A \gg$, $q : \ll y \text{ est } B \gg$ deux propositions floues. Notons a, b les valeurs de vérité possibles de p et q respectivement. Une implication floue, que l'on note $I(a, b)$, est une fonction :

$$I(a, b) : [0, 1] \times [0, 1] \rightarrow [0, 1] \dots\dots\dots (I-25)$$

qui définit la valeur de vérité de la proposition floue «si p alors q ». L'implication floue est également notée : $A \Rightarrow B \dots\dots\dots (I-26)$

Les valeurs de vérité sont exprimées par : $a = \mu_A(x)$, $b = \mu_B(y)$.

La fonction $I(a, b)$ est donc équivalente à la fonction d'appartenance, que l'on note $\mu_R(x, y)$, d'une relation floue définie entre X et Y . Cette fonction d'appartenance s'exprime, pour tout (x, y) de $X \times Y$, en utilisant la fonction d'appartenance $\mu_A(x)$ intervenant dans la prémisse et celle $\mu_B(y)$ intervenant dans la conclusion de la règle :

$$\mu_R(x, y) = I(\mu_A(x), \mu_B(y)) \dots\dots\dots (I-27)$$

Ainsi, la définition d'une implication floue peut se ramener à celle d'une fonction d'appartenance. Les implications floues le plus souvent employées sont précisées dans le tableau (I-2) :

Nom	Notation	Valeur de vérité
Lukasiewicz	$\mu_R(x,y)$	$\text{Min}(1 - \mu_A(x) + \mu_B(y), 1)$
Kleen-Dienes	$\mu_R(x,y)$	$\text{Max}(1 - \mu_A(x), \mu_B(y))$
Zadeh	$\mu_R(x,y)$	$\text{Max}(1 - \mu_A(x), \text{min}(\mu_A(x), \mu_B(y)))$
Reichenbach	$\mu_R(x,y)$	$1 - \mu_A(x) + \mu_A(x) \cdot \mu_B(y)$
Mamdani	$\mu_R(x,y)$	$\text{Min}(\mu_A(x), \mu_B(y))$
Larsen	$\mu_R(x,y)$	$\mu_A(x) \cdot \mu_B(y)$
Turksen 88	$\mu_R(x,y)$	$\text{Max}[\text{min}(\mu_A(x), \mu_B(y) \cdot (1 - \mu_A(x))), \text{max}(\mu_B(y), (1 - \mu_A(x)))]$
Gogen	$\mu_R(x,y)$	$\begin{cases} 1 & \text{si } \mu_A(x)=0 \\ \text{Min}(\mu_B(y) / \mu_A(x), 1) & \text{si non} \end{cases}$
Brouwer-Godel	$\mu_R(x,y)$	$\begin{cases} 1 & \text{si } \mu_A(x) \leq \mu_B(y) \\ \mu_B(y) & \text{si non} \end{cases}$
Rescher-Gaines	$\mu_R(x,y)$	$\begin{cases} 1 & \text{si } \mu_A(x) \leq \mu_B(y) \\ 0 & \text{si non} \end{cases}$

Tableau I-2 : Principales implications floues

Parmi les implications définies dans le tableau (I-2), les implications de Mamdani et de Larsen sont les plus connues. [WeW 98]

I.4.5 Les règles floues :

Une règle floue est une relation exprimée à l'aide d'une implication, par exemple :

SI « Erreur est POSITIVE » ET « Variation d'erreur est NEGATIVE » ALORS « la commande est GRANDE ».

Cette règle est constituée de trois propositions :

- La proposition 1 est « Erreur est POSITIVE »
- La proposition 2 est « Variation d'erreur est NEGATIVE »

Ces deux propositions forment la prémisse de la règle.

- La proposition 3 est « la commande est GRANDE », forme la conclusion de la règle.

Alors la forme générale de la règle est :

« Si condition, Alors action » ou aussi « Si prémisse, Alors conséquence ».

Dans la théorie des systèmes flous il existe deux types de règles :

I. 4.5 .1 Règle de Mamdani :

La règle utilisée par Mamdani est de la forme :

Si x_1 est A_1 , et x_2 est A_2 , etet x_n est A_n , alors y est B (I-28)

Où : A_1, A_2, \dots, B sont des ensembles flous caractérisant la partie prémisse et la partie conséquence, définies par les fonctions d'appartenances $\mu_{A_i}(x_i)$ et $\mu_B(y)$, respectivement.

La règle(1) peut être aussi écrite comme : x est A alors y est B . où A : est l'ensemble flou défini par le produit cartésien, la règle floue est interprétée comme une implication floue de la forme : $A \rightarrow B$.

I. 4.5.2 Règle de Takagi-Sugeno (TS) :

A la différence de celle de Mamdani, la règle de TS implique une conséquence qui n'est pas un ensemble flou, mais une fonction, soit :

Si x_1 est A_1 et x_2 est A_2 et ...et x_n est A_n alors y est $B(x)$.

Où : $B(x)$ est une fonction analytique des entrées x , de la même manière, la règle de TS peut être vue comme une implication floue de la forme : $A \rightarrow B(x)$

I.4.6 Le raisonnement et la prise de décision :

En général, plusieurs valeurs de variable linguistique, convenablement définie par des fonctions d'appartenance, sont liées entre elles par des règles, à fin de tirer des conclusions. On parle alors de déduction floues, ou inférence dans ce contexte. On peut distinguer deux genres de règle d'inférence.

- Inférence avec une seule règle :

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 : $y = [x_1 \text{ ou } (x_2 \text{ et } x_3) \text{ ou } \dots] \text{ et } x_n$.

- Inférence avec plusieurs règles :

Se présente lorsque une où plusieurs variables nécessitent une prise de décision. Les règles peuvent alors être exprimées sous la forme :

Opération= Si condition 1, Alors opération 1, ou
 Si condition 2, Alors opération 2, ou

 Si condition m, Alors opération m.

1.4.7 Commande et réglage par la logique floue :

De façon générale, un système de commande a pour objectif de piloter l'entrée d'un processus afin d'obtenir un fonctionnement correct de ce dernier.

La mesure réalisée sur un système est prise en compte par l'intermédiaire d'une variable linguistique, qui est issue d'une analyse par un expert humain. Ensuite, l'action à réaliser est déduite à la fois d'un ensemble de règles de commande et de l'état du système, qualifié par la variable linguistique. Enfin, la commande finale du système est créée en utilisant les conclusions de la déduction.

1.4.7.1 Structure d'un contrôleur flou :

Un contrôleur flou peut être vu comme un système expert fonctionnant à partir d'une représentation d'une connaissance divisée en un fichier de faits et un fichier de règles en plus d'un moteur d'inférence qui permet de consulter ces fichiers et produire des conclusions.

La figure I-13 montre le schéma synoptique général d'un contrôleur flou, et cela quelque soit le type d'application du contrôleur. On trouve généralement la même configuration qui est constituée de quatre blocs essentiels, à savoir :

- une base de connaissances sous forme de règles floues et paramètres des fonctions d'appartenance.
- un bloc de décision ou moteur d'inférence.
- une fuzzification qui consiste en la transformation des entrées précises en degrés d'appartenance.
- une défuzzification qui consiste en la transformation des résultats flous en des sorties précises.

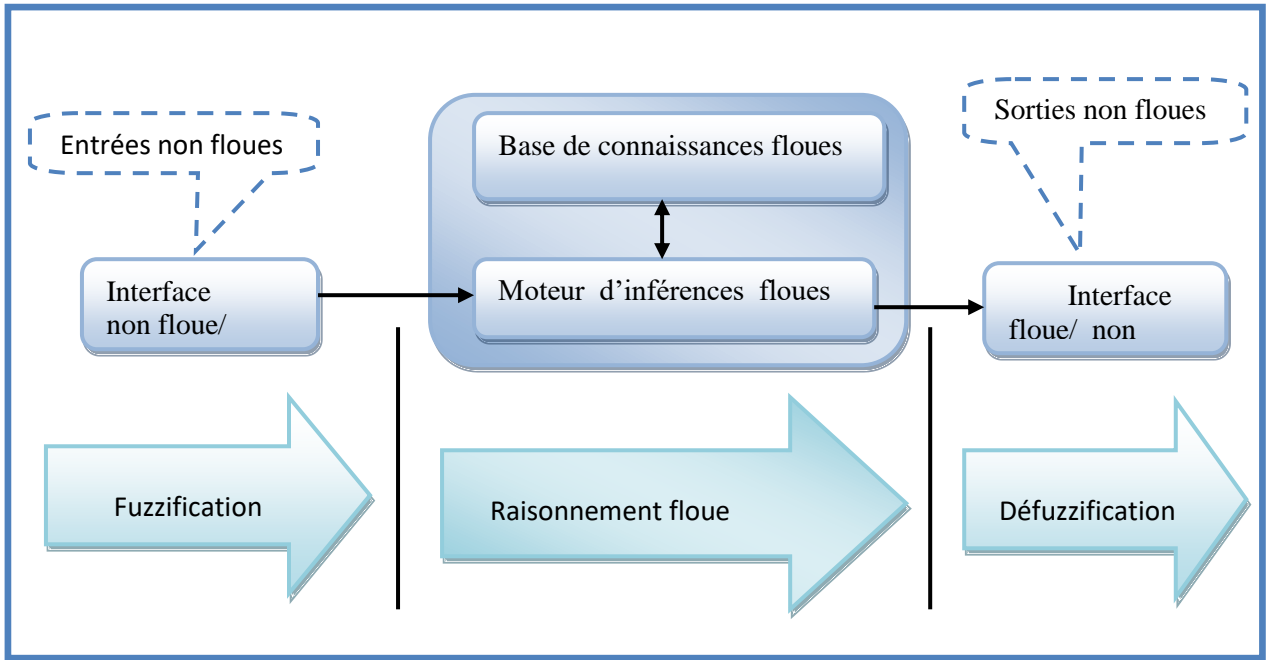


Figure I-13 : structure d'un régulateur flou

1.4.7.1.1 La fuzzification :

La stratégie de cette opération consiste à convertir les grandeurs d'entrée mesurées ou estimées en variable linguistique, exprimées par des termes flous. Ce processus est basé sur la collection des fonctions d'appartenances, et de relier des données numériques à des ensembles flous, qui sont définies par des concepteurs de système. La fuzzification [WeW 98] est une étape clé dans tout contrôleur flou. Il y a au minimum deux choix de cette transformation :

• **La fuzzification singleton :**

L'opérateur de fuzzification convertit l'entrée numérique $x_0 \in X$ en singleton flou dans X tel que (Figure I-14)

$$\mu_{x_0}(x) = 1 \quad \text{si } x = x_0 \dots\dots\dots (I-29)$$

$$\mu_{x_0}(x) = 0 \quad \text{si } x \neq x_0 \dots\dots\dots (I-30)$$

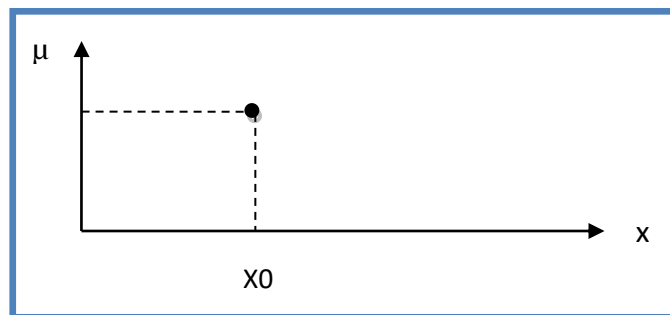


Figure1-14: fuzzification singleton

• **fuzzification non- singleton:**

La mesure de la variable est incertaine, par exemple à cause de bruit, le sous ensemble flou x_0 doit être représenté par un fait imprécis, on utilise une méthode de fuzzification qui associe à la variable mesurée x_0 une fonction d'appartenance telle que :

$$\mu_{x_0}(x) = \max \left(0, 1 - \left| \frac{x - x_0}{\varepsilon} \right| \right) \dots\dots\dots (I-31)$$

Ce sous ensemble comprend donc la mesure x_0 avec une appartenance unité et les valeurs voisines de x_0 comme montre la figure (I-15) ci- dessous.

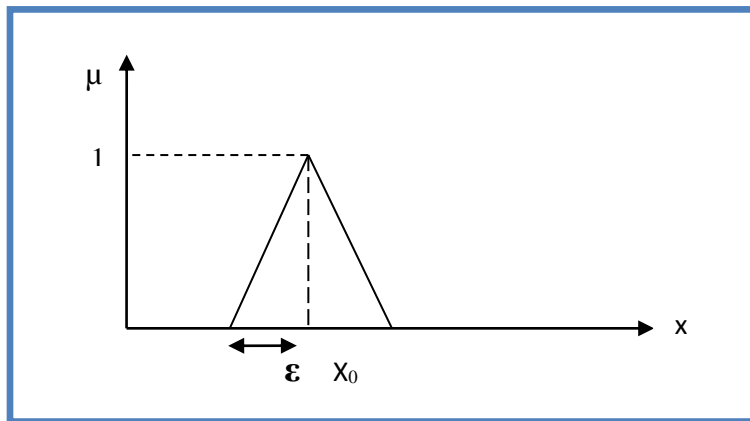


Figure I-15 : fuzzification non- singleton

En général, on attribue à chaque entrée du système un vecteur d'appartenance de largeur m (nombre d'ensembles flous), ou la $i^{\text{ème}}$ valeur indique la valeur du $i^{\text{ème}}$ ensemble flou à cette entrée ; où la $i^{\text{ème}}$ valeur représente une mesure de degré pour lequel l'entrée considérée appartient au $i^{\text{ème}}$ ensemble flou.

I.4.7.1.2 La base de connaissances:

Cette base est composée de l'ensemble des informations qu'on utilise dans les processus du contrôle, qui permet d'établir l'ensemble de fonctions d'appartenance et les règles floues.

- **Base de données:** un nombre des ensembles flous répartis sur l'univers de discours caractérisant les variables linguistiques des entrées et des sorties.
- **Base de règle :** Les connaissances sont représentées sous forme de règle « SI-ALORS ». Une règle floue est dite « mémoire associative floue (FAM)», car elle associe les ensembles flous d'entrée et de sortie. une règle floue peut être définie par : $R^{(k)} : \text{Si } x_1 \text{ est } A^k_1 \text{ et } x_2 \text{ est } A^k_2 \dots \text{ et } x_n \text{ est } A^k_n \text{ Alors } Y \text{ est } B^k$

A^k_i et B^k sont des ensembles flous dans $X_i \subset IR$ et $Y \subset IR$ respectivement.

$K=1,2,3,\dots,m$; $i=1,2,3,\dots,n$;

$x \in X = X_1 \times X_2 \times \dots \times X_n$ et $y \in Y$ sont des variables linguistiques.

I.4.7.1.3 Le moteur d'inférence flou :

Le principe de cette opération consiste à établir la liaison entre les grandeurs d'entrée du régulateur exprimées par des termes linguistiques et la variable de sortie sous sa forme floue à partir d'une parfaite connaissance et d'une expertise concernant le fonctionnement du processus à régler. Il y a plusieurs façons d'exprimer les inférences, à savoir par :

-Description linguistique.

On écrit les règles de façon explicite comme dans l'exemple suivant [BAL 99]:

Si (la température est élevée **ET** la vitesse est faible) **ALORS** la tention est grande positive.

-Description symbolique :

Les ensembles sont désignés de manière abrégée par leur symbole, on obtient alors une certaine simplification dans l'écriture, le tableau (I-3) présente quelques abréviations :

Symbole	Signification
NB	Negative Big
NM	Negative Medium
NS	Negative Small
ZE	Zéro
PS	Positive Small
PM	Positive Medium
PB	Positive Big

Tableau I- 3: désignation standard des ensembles flous

- Description par matrice d'inférences :

Une simplification ultérieure de description des inférences s'obtient à l'aide d'une représentation graphique, appelée matrice d'inférences, le tableau I-4 représente un exemple de matrice (matrice de MACVICAR-WHELN).

E dE	NB	NS	ZE	PS	PB
NB	NB	NB	NB	NS	ZE
NS	NB	NB	NS	ZE	PS
ZE	NB	NS	ZE	PS	PB
PS	NS	ZE	PS	PB	PB
PB	ZE	PS	PB	PB	PB

Tableau I-4 : matrice d’inférences pour deux variables linguistiques

I.4.7.1.4 La défuzzification :

C’est une interface «Linguistique-Numérique» permet d’obtenir une valeur précise à donner au processus (action du système) à partir des ensembles issus du bloc d’interface .Il existe plusieurs stratégies de défuzzification :

- **Méthode de maximum(MC):**

Cette méthode génère une commande qui représente l’abscisse de la valeur maximale de la fonction d’appartenance résultante issue de l’inférence floue. Cette méthode est simple rapide et facile, mais présente un certain inconvénient, lorsqu’ il existe plusieurs valeurs pour les quelles la fonction d’appartenance résultante est maximale, et ne tient pas compte de toutes les règles.

- **Méthode de moyenne des maximums(MOM):**

Cette méthode génère une commande qui représente la valeur moyenne de tous les maximums, dans le cas où il existe plusieurs valeurs pour lesquelles la fonction d’appartenance résultante est maximale.

- **Méthode de centre de gravité(COG):**

Cette méthode à prendre comme sortie le centre de gravité de l’ensemble flou de sortie. Pour un ensemble discret, l’expression souvent utilisée est :

$$y_0^{COG} = \frac{\sum_{i=1}^N y_i \mu_B (y_i)}{\sum_{i=1}^N \mu_B (y_i)} \dots\dots\dots (I-32)$$

N : Le nombre de niveaux de quantification de la sortie.

Y_i : Le centre de gravité de label de sortie B_i ,

Pour un ensemble continu:

$$y_0^{COG} = \frac{\int y \mu_B (y) dy}{\int \mu_B (y) dy} \dots\dots\dots (I-33)$$

La méthode de centre de gravité est la méthode la plus utilisée pour la plupart des cas.

I.5 Types de contrôleur flou :

Il existe plusieurs types de contrôleurs flous, parmi eux on citera ceux de **MAMDANI**, de **SUGENO**, de **LARSON**, de **TSUKAMOTO**. La différence entre ces contrôleurs flous se situe dans la conséquence de leurs règles floues, et ainsi leurs procédures d'agrégation et de défuzzification.

I.5.1 Contrôleurs flou de type Mamdani:

- Il a été proposé la première fois pour commander une combinaison de moteur et de chaudière de vapeur par une somme des règles linguistiques obtenues par des opérateurs humains expérimentés.
- Ce type de commande prend seulement des valeurs réelles (numériques) comme entrée, alors le défuzzicateur doit convertir un ensemble flou en une valeur numérique.

Les avantages du modèle flou de Mamdani sont résumés dans :

- 1) La simplicité dans la représentation des règles floues: les règles de la forme SI-ALORS prennent la forme d'ensemble flou, ce qui les rend facile à interpréter.
- 2) La flexibilité dans l'implémentation: vu la liberté dans le choix des fonctions d'implication floues.

Le principal inconvénient de ce type de modèle est le nombre élevé de règles floues SI-ALORS lorsque le système à contrôler est non linéaire et relativement complexe.

I.5.2 Contrôleurs flous de type Sugeno :

Il est semblable à la méthode de Mamdani à quelques différences près. En fait, les deux parties du système flou (base de règle, fuzzification) pour les entrées, sont exactement identiques.

En revanche, à la différence du système de Mamdani les fonctions d'appartenance de la variable de sorties sont uniquement linéaires ou constantes dans le cas du système de Sugeno.

I.6 Conclusion :

Nous avons vu dans ce chapitre l'intérêt de la logique floue dans le domaine du contrôle de processus. Cette approche permet de tenir compte à la fois des connaissances d'un expert humain et de l'incertitude et de l'imprécision des données traitées par le contrôleur.

En fait, le fonctionnement d'un contrôleur flou dépend d'un nombre très important de paramètres (fonctions d'appartenance, règles floues, règles d'inférence, défuzzification) qu'il faut établir lors de la conception.

Dans le chapitre II, nous décrivons le principe des algorithmes génétiques, qui permettent d'explorer de façon très efficace l'espace des solutions possibles d'un problème.

CHAPITRE II



Les algorithmes génétiques

II.1 Introduction

Les Algorithmes Génétiques (en abrégé AG) sont des méthodes d'exploration de l'ensemble des solutions d'un problème, en utilisant les mêmes mécanismes que ceux intervenant dans la sélection naturelle. Ils sont utilisés principalement dans les domaines de l'optimisation et de l'apprentissage. Le parallélisme implicite et l'exploration globale de l'espace des solutions sont les deux principaux avantages des algorithmes génétiques.

Tout d'abord, les algorithmes génétiques peuvent traiter des problèmes pour lesquels les solutions potentielles sont situées dans un espace de grande dimension, ce qui ne permet pas d'utiliser des méthodes standard reposant sur une exploration systématique. Ensuite, la recherche d'une solution optimale à un problème peut être réalisée par un algorithme génétique sans nécessiter de connaissance a priori sur la répartition des solutions dans l'espace.

Bien que les mécanismes exploités dans les algorithmes génétiques aient été découverts par Darwin et Mendel, le livre de John Holland publié en 1975 est considéré comme étant à l'origine des algorithmes génétiques. Dans ce livre, il a démontré le théorème des schémas qui constitue la justification théorique de cette approche. La même année, De Jong a soutenu une thèse qui sert encore de référence dans ce domaine, dans laquelle il a proposé une série de fonctions de test et réalisé de nombreuses expérimentations. En 1989, Goldberg [Gol 89] a publié un livre sur les AG qui est devenu une référence.

De nos jours, les AG ont trouvé des applications dans des domaines très variés, allant de la biologie jusqu'aux applications liées plus directement à l'informatique (traitement d'images, optimisation, placement de formes) [WeW98].

II.1.1 Evolution naturelle et algorithmes génétiques

Depuis que Charles Darwin a formulé l'idée simple d'une évolution naturelle des espèces basée sur un mécanisme de sélection et de reproduction, la théorie de l'évolution biologique est devenue un centre d'intérêt pour les scientifiques. L'étude des fossiles montre que la structure complexe de la vie peut évoluer sur des périodes de temps relativement courtes.

A l'heure actuelle, on sait que l'évolution d'une espèce a lieu lorsque les chromosomes, qui contiennent le code génétique de sa structure biologique, se trouvent modifiés. Bien que toutes les caractéristiques des évolutions naturelles ne soient pas connues, les spécialistes considèrent qu'il faut admettre au moins les règles suivantes :

- L'évolution n'agit pas directement sur les êtres vivants ; elle opère en réalité sur les chromosomes contenus dans leurs cellules.
- L'évolution est régie par deux composantes : la sélection et la reproduction. La sélection naturelle construit un lien entre les chromosomes et les individus. Elle permet d'aboutir à un meilleur taux de reproduction des individus qui ont de «bons» chromosomes.
- La reproduction est le moteur de l'évolution. Les modifications apparaissant lors de la reproduction peuvent produire de nouvelles générations dont les caractéristiques sont différentes.
- L'ensemble des chromosomes et le codage de ces chromosomes comprennent toutes les informations caractéristiques d'un individu. Ces individus peuvent cependant s'adapter à d'autres conditions de vie.

La plupart des espèces biologiques se reproduisent par sélection naturelle et reproduction sexuée. La sélection décide quel individu peut vivre et se reproduire alors que la reproduction sexuée assure le «mélange» du matériel génétique. Les espèces qui utilisent un mode de reproduction non sexué évoluent en général beaucoup moins vite, car la modification du matériel génétique n'intervient que lors de mutations.

Ces principales caractéristiques de l'évolution naturelle ont attiré John Holland et ses collègues de l'université du Michigan lors de recherches sur l'apprentissage des machines dans les années 60. Holland a trouvé que l'apprentissage se réalise non seulement par l'adaptation d'un individu mais également par l'évolution d'un groupe lors de générations successives. En 1975, il a publié son ouvrage «Adaptation in Natural and Artificial Systems» dans lequel il a proposé les premières versions d'un algorithme génétique, qui permet un apprentissage basé sur les règles de la sélection naturelle [WeW 98].

II.1.2 Terminologie

Avant de décrire le principe des algorithmes génétiques, il est nécessaire de présenter le vocabulaire que nous allons utiliser tout au long de ce travail.

	<i>Dans les systèmes naturels</i>	<i>Dans les algorithmes génétiques</i>
<i>Chromosome → chaîne, chromosome</i>	Les chromosomes sont les porteurs de l'information génétique nécessaire à la construction et au fonctionnement d'un organisme.	Les chaînes, ou chromosomes sont analogues aux chromosomes des systèmes biologiques. Ils sont les éléments à partir desquels sont élaborées les solutions
<i>Génotype → structure</i>	L'ensemble du matériel génétique est appelé le génotype	L'ensemble des chaînes est appelé structure
<i>Phénotype → ensemble de paramètres, solution, point</i>	L'organisme formé par l'interaction de l'ensemble du matériel génétique avec son environnement est appelé le phénotype	les structures décodées forment un ensemble de paramètres donné, ou une solution ou un point dans l'espace des solutions.
<i>Gène → trait, détecteur</i>	Les chromosomes sont constitués par les gènes	on dit que les chaînes se composent de traits ou détecteurs.
<i>Allèle → valeur de caractéristique</i>	L'allèle est une composante du gène. Les allèles sont les différentes valeurs que peuvent prendre les gènes.	L'allèle est également appelé valeur caractéristique
<i>Locus → position dans la chaîne</i>	Le locus est la position du gène dans le chromosome.	Ce terme est appelé également position dans la chaîne.

<p><i>Individu, organisme → individu, chromosome</i></p>	<p>Un organisme ou un individu biologique est une forme qui est le produit de l'activité des gènes.</p>	<p>l'individu est réduit à un chromosome, on l'appelle indifféremment individu ou chromosome.</p>
<p><i>Population → population, génération</i></p>	<p>La population est un groupe d'individus.</p>	<p>La population est l'ensemble des individus ou des chromosomes. Les populations sont également appelées des générations</p>

II.2 Principe de base d'un AG standard

Un AG standard nécessite en premier le codage de l'ensemble des paramètres du problème d'optimisation en une chaîne de longueur finie. Le principe d'un AG est simple, il s'agit de simuler l'évolution d'une population d'individus jusqu'à un critère d'arrêt. On commence par la génération d'une population initiale d'individus (solutions) de façon aléatoire. Puis, à chaque génération, des individus sont sélectionnés, cette sélection est effectuée à partir d'une fonction objectif appelée fonction d'adaptation. Puis, les opérateurs de croisement et de mutation sont appliqués et une nouvelle population est créée. Ce processus est itéré jusqu'à un critère d'arrêt. Le critère le plus couramment utilisé est le nombre maximal de générations que l'on désire effectuer. La **figure II.1** présente le principe de l'AG standard.

L'AG débute par la génération d'une population initiale et l'évaluation de la fonction d'adaptation de tous les individus qui composent cette première population. Puis, des individus sont sélectionnés aléatoirement pour la reproduction selon le principe de la survie

du plus adapté. Ensuite, des individus « enfants » (ou les descendants) sont générés en appliquant les deux opérateurs génétiques suivants : le croisement et la mutation. Ces enfants sont placés dans une nouvelle population $P(t)$ et vont se substituer, en tout ou en partie, à la population de la génération précédente. De nouvelles populations d'individus vont ensuite se succéder, d'une génération (t) à la génération $(t+1)$, chaque génération représentant une itération jusqu'à l'atteinte du critère d'arrêt. L'AG présenté ci-dessus est dit générationnel car tous les individus enfants générés sont placés dans une population et vont remplacer entièrement la population des individus parents **[DRD 05]**.

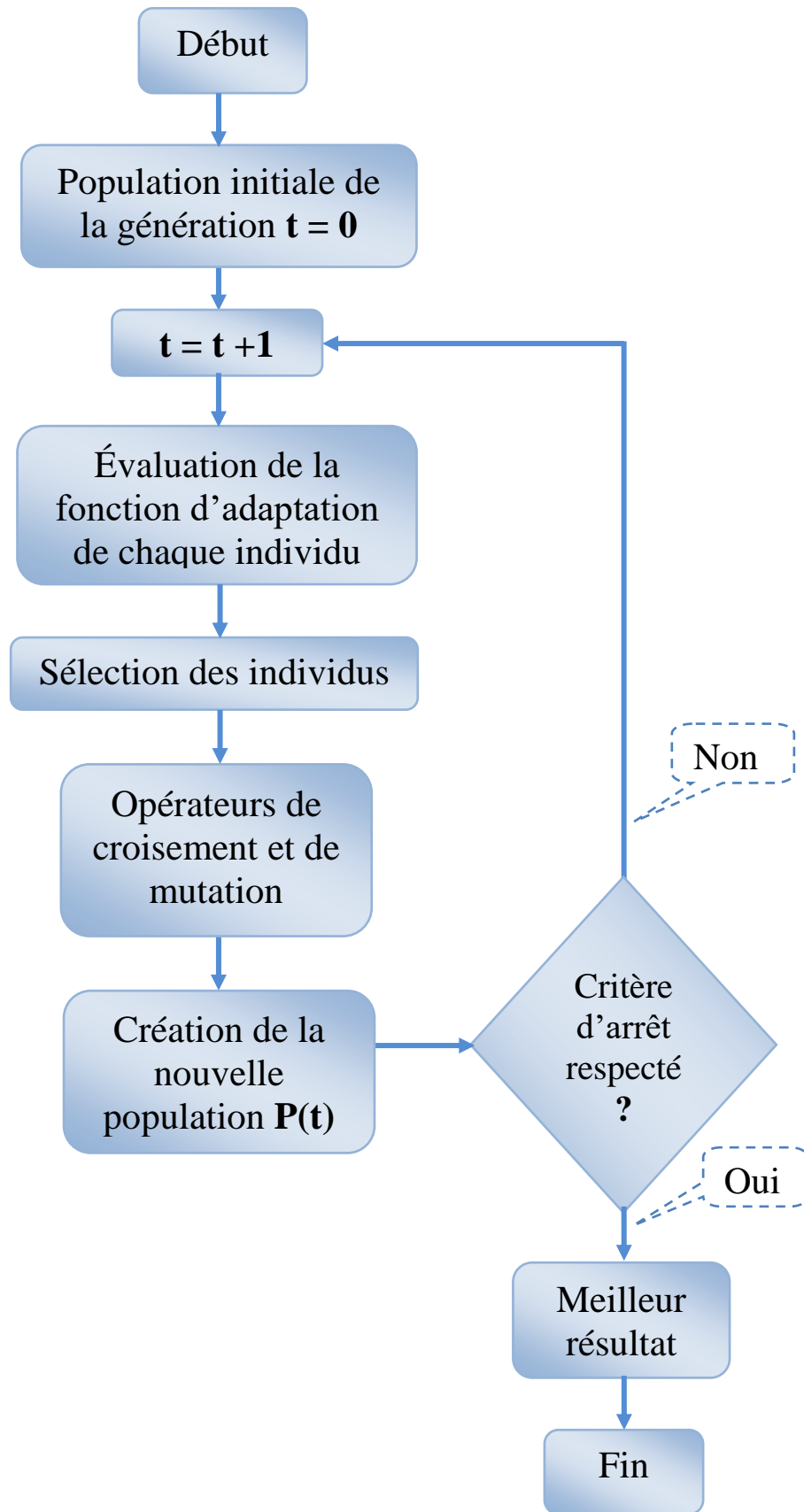


Figure II.1: Organigramme d'un AG standard.

II.3 Les opérateurs génétiques simples

Ils sont au nombre de trois : la sélection, le croisement et la mutation. Pour passer d'une génération à l'autre, on applique ces opérations à l'ensemble de points et dans l'ordre suivant sélection, croisement et mutation. [Mat 00].

II.3.1 La sélection

La sélection, comme son nom l'indique, permet de définir quels seront les individus de la population courante P qui vont être dupliqués dans la nouvelle population P' et vont servir de parents (application de l'opérateur de croisement).

Soit n le nombre d'individus de P , on doit en sélectionner $n/2$ (l'opérateur de croisement nous permet de repasser à n individus).

Cet opérateur est peut-être le plus important puisqu'il permet aux individus d'une population de survivre, de se reproduire ou de mourir. En règle générale, la probabilité de survie d'un individu sera directement reliée à son efficacité relative au sein de la population.

Il existe différents principes de sélection, dont on citera la plus connue et utilisée.

La « roulette wheel selection » (ou sélection par roulette de casino) : elle consiste à associer à chaque individu un segment dont la longueur est proportionnelle à sa fitness. Ces segments sont ensuite concaténés sur un axe gradué que l'on normalise entre 0 et 1. On tire alors un nombre aléatoire de distribution uniforme entre 0 et 1, puis on regarde quel est le segment sélectionné, et on reproduit l'individu correspondant. Avec cette technique, les bons individus seront plus souvent sélectionnés que les mauvais, et un même individu pourra avec cette méthode être sélectionné plusieurs fois. Néanmoins, sur des populations de petite taille, il est difficile d'obtenir exactement l'espérance mathématique de sélection à cause du faible nombre de tirages (le cas idéal d'application de cette méthode est bien évidemment celui où la population est de taille infinie). On aura donc un biais de sélection plus ou moins fort suivant la dimension de la population.

<i>individus</i>	<i>fitness</i>	<i>Fitness relative</i>
1	15	0.1
2	30	0.2
3	45	0.3
4	22.5	0.15
5	7.5	0.05
6	30	0.2

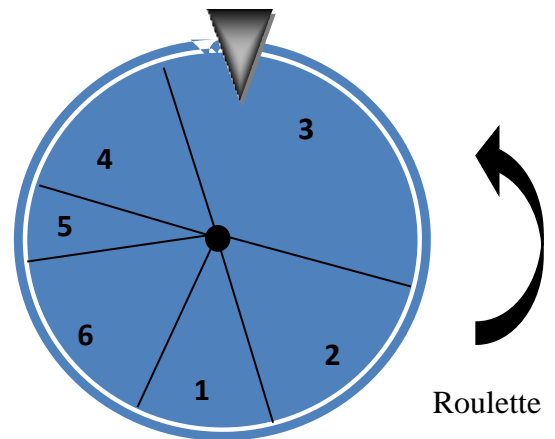


Figure II.2 - Exemple d'application de la roulette wheel selection

Et pour calculer les valeurs de la case Fitness relative, on utilise la relation suivante :

$$F(X_i) / \sum_j^6 F(X_j) \dots\dots\dots(\text{II.1})$$

II.3.2 Croisement

Après la sélection, les chaînes ou les chromosomes les mieux adaptées dans le processus de reproduction subiront l'opération de croisement. Cette dernière consiste à échanger du matériel génétique entre les deux chaînes reproductrices (parents) pour produire deux nouvelles chaînes (enfants). Le croisement est un processus aléatoire de probabilité P_c appliquée à un couple de parents arbitrairement choisis de la population.

Parmi les opérateurs de croisement les plus utilisés, on a:

- a. **Croisement à un point** : Le plus simple opérateur de croisement est à un point où premièrement l'emplacement du croisement est choisi avec une probabilité uniforme sur la longueur du chromosome ou de la chaîne. En d'autres mots, un nombre entier k , représentant une position ou un emplacement sur la chaîne, est choisi aléatoirement entre 1 et la longueur L de la chaîne moins 1 ($k \in [1, L - 1]$), ce nombre représentera donc la position où se produira le croisement, et enfin deux nouvelles chaînes seront créées en échangeant le matériel génétique (caractères) compris entre les positions $k + 1$ et L des chaînes reproductrices. Ce processus est illustré par la **figure II.3**

Parent 1	00110100 1110
Parent 2	01110110 0101
Enfant 1	00110100 0101
Enfant 2	01110110 1110

Figure II.3: Croisement à un point avec $K = 8$.

- b. **Croisement en deux points** : Cette technique est similaire à la précédente; mais, dans ce cas deux positions sont sélectionnées et les sous chaînes entre ces deux positions sont échangés. Cette méthode a les mêmes propriétés déjà décrites, mais elle peut combiner certains schémas que la version à un point ne peut pas.

Parent 1	00 101101 1110
Parent 2	01 110110 0101
Enfant 1	00 110110 1110
Enfant 2	01 101101 0101

Figure II.4: Croisement en deux points.

- c. **Croisement aléatoire** : Cet opérateur combine deux chromosomes selon une chaîne binaire aléatoire. A chaque position, les bits correspondants des parents sont échangés si la chaîne aléatoire contient un **1** à cette position. Si le bit aléatoire est **0**, il n'y a pas d'échange.

Parent 1:	001011011110
Parent 2:	011101100101
Chaîne aléatoire:	110011110010
Enfant 1:	011001101100
Enfant 2:	001111010111

Figure II.5: Croisement aléatoire.

Tous les opérateurs de croisement sont réalisés d'une manière qu'ils génèrent de nouveaux chromosomes différents de ceux appartenant déjà à la génération, c'est à dire, différents des deux parents, l'échange actuel d'information est aussi exigé. Si les nouveaux chromosomes créés par l'opérateur ne satisfont pas ces conditions, différentes positions de croisement sont essayées. S'ils échouent encore à créer des chromosomes différents, un nouveau accouplement aléatoire est sélectionné. Cette méthode génère des enfants qui n'ont pas de double, avec une probabilité élevée, mais si elle échoue encore la mutation peut modifier les doubles.

II.3.3 Mutation

La mutation maintient la diversité dans la population des chaînes en modifiant un gène d'un chromosome avec une probabilité donnée qui est généralement petite. Cet opérateur prend le complément d'un bit du chromosome avec une probabilité P_m . Il résulte en une marche aléatoire dans l'espace des paramètres et introduit de nouvelles informations dans le processus d'évolution qui auront pu être perdues avec une convergence prématurée de l'algorithme. Ce processus est exécuté bit par bit comme le montre la **figure II.6**,

Enfant	00110100 0 101
Enfant muté	00110100 1 101

Figure II.6: Mutation.

Pour ne pas altérer les performances de l'AG, Goldberg [Gol 89] conseille de prendre une fréquence de mutation tous les 1000 bits. Certains chercheurs préconisent de prendre la valeur suivante

$$P_m = \frac{1}{L} \dots\dots\dots (II.2)$$

Où L est la longueur du chromosome

II.4 le codage

Le **codage** est une partie très importante des algorithmes génétiques. Il permet de représenter l'individu sous la forme d'un chromosome. Ce chromosome est constitué de gènes qui prennent des valeurs dans un alphabet binaire ou non.

Plusieurs codes d'informations sont utilisés. Les plus fréquemment utilisés sont le code *binnaire naturel* et le code *binnaire de Gray*. Plus récemment le codage "*réel*" a fait son apparition. [SOP 04]

II.4.1 Le codage binnaire

Ce codage consiste, pour un individu donné, à concaténer toutes ses variables codées en binaire. La chaîne binaire **1000| 0110| 1101**, par exemple, correspond à un individu défini par 3 variables (**8, 6, 13**) en codage binaire naturel sur 4 bits chacune.

Ce codage binaire présente plusieurs avantages : alphabet minimum, facilité de mise en place d'opérateurs génétiques et existence de résultats théoriques. Néanmoins, ce codage présente trois inconvénients majeurs :

- Les performances de l'algorithme sont diminuées lorsque la longueur de la chaîne augmente.
- Deux nombres décimaux voisins (exemple : 7 et 8) peuvent être très éloignés dans le codage binaire naturel (1000 et 0111) : falaise de Hamming. Ce problème peut être réglé en remplaçant le code binaire naturel par le code binaire de Gray.
- La dissymétrie entre le 0 (plus fréquent) et le 1.

II.4.2 Le codage réel

Ce codage consiste simplement à la concaténation des variables x_i d'un individu x . Exemple un individu x (25, 31, 8) est codé 25| 31| 8. Ce codage présente des avantages majeurs. Il est plus précis que le codage binaire et l'espace de recherche est le même que l'espace du problème. Il a le mérite d'être simple à utiliser, l'évaluation de la fonction coût est plus rapide. Le codage réel évite de faire le transcodage du binaire naturel ou de Gray vers les réels à chaque évaluation. Néanmoins, il possède deux inconvénients, son alphabet est infini, et il a besoin d'opérateurs appropriés.

II.5 Conclusion

Les AGs sont des outils d'optimisation performants qui permettent de réaliser une exploration globale de l'espace des paramètres. Contrairement aux méthodes déterministes classiques, ils ne nécessitent aucun calcul des dérivées et peuvent être appliqués aussi bien à des fonctions continues d'une seule variable, qu'à des fonctions discontinues dépendant d'un grand nombre de paramètres. Les AGs sont capables d'obtenir des solutions quasi-optimales pour différents types de problèmes sans connaissance explicite du domaine de travail en manipulant simplement des chaînes de bits et en utilisant des opérateurs simples qui ne mettent en jeu que des procédures aussi peu complexes que la génération des nombres aléatoires, la copie des chaînes et les échanges des morceaux de chaînes.

CHAPITRE III



*Optimisation d'un contrôleur flou
par Algorithme Génétique*

III.1 Introduction :

Le fonctionnement d'un contrôleur flou dépend principalement des caractéristiques de trois sous-systèmes : la fuzzification, les règles floues et la défuzzification. La phase de fuzzification est parfaitement spécifiée lorsque sont définies les fonctions d'appartenance décrivant les entrées. Les règles floues sont issues d'une analyse des connaissances fournies par un expert humain et de certaines caractéristiques du système commandé et La défuzzification doit permettre de transformer une valeur de commande envoyée au système.

Les bases de règles floues sont utilisées depuis les années 70 pour réaliser de nombreuses applications industrielles. Depuis la fin des années 70 des méthodes d'apprentissage automatique des bases de règles floues sont apparues. [GuF 94].

Un algorithme génétique recherche le ou les extrema d'une fonction définie sur un espace de données. Pour l'utiliser, on doit disposer des quatre éléments suivants [SoR 04] : codage ; génération de la population initiale ; les opérateurs de croisement et mutation.

III.2 Structure du contrôleur flou :

Un contrôleur flou peut être vu comme un système expert fonctionnant à partir d'une représentation des connaissances basées sur la théorie des ensembles flous .Ce contrôleur fourni un algorithme de conversion d'une stratégie de contrôle linguistique basée sur l'expertise humaine en une stratégie de contrôle automatique. Après avoir énoncé les concepts de base et les termes linguistiques utilisés en logique floue dans le chapitre I, nous présentons la structure de notre contrôleur flou.

Dans ce qui suit, nous nous intéressons principalement au régulateur de deux entrées erreur de la position (E_P) (ΔD) et l'erreur de l'angle (E_G) ($\Delta E\theta$) et une sortie la variation de l'orientation ($\Delta\theta$).

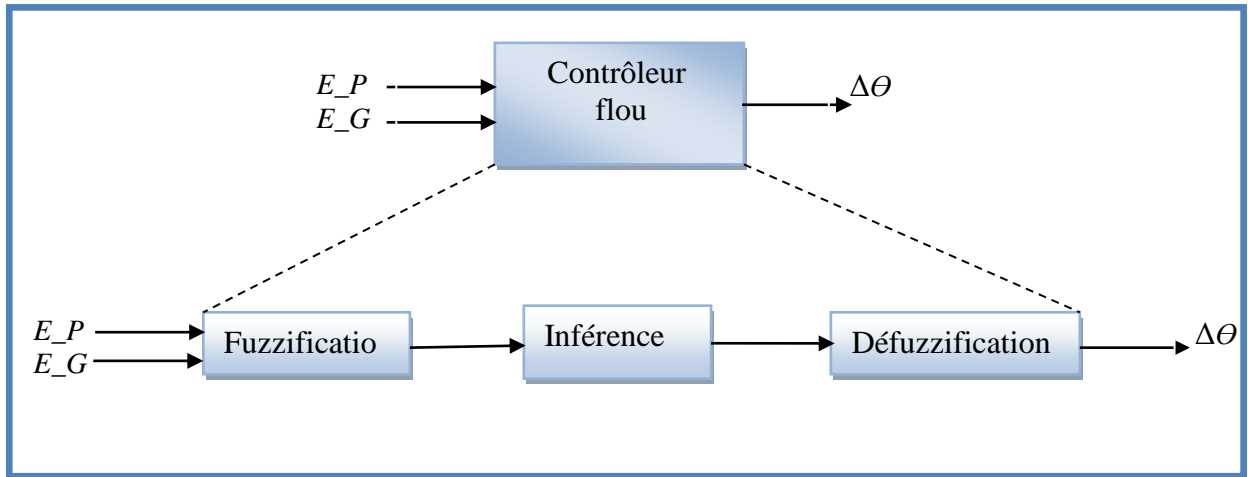


Figure III-1: structure d'un contrôleur flou

Les règles d'inférences permettent de déterminer le comportement du contrôleur flou. Il doit donc inclure des étapes intermédiaires qui lui permettent de passer des grandeurs réelles vers les grandeurs floues et vice versa ; ce sont les étapes de fuzzification et de defuzzification.

La fuzzification :

Les ensembles flous des variables d'entrée et leurs fonctions d'appartenance sont à définir en premier lieu.

L'étape de fuzzification permet de fournir les degrés d'appartenance de la variable floue à ses ensembles flous en fonction de la valeur réelle de la variable d'entrée.

Nous avons opté pour des fonctions triangulaires et trapézoïdales pour les variables d'entrées (figure III-2). Elles permettent une implantation facile et l'étape de fuzzification ne requiert alors que peu de temps de calcul lors de son évaluation en temps réel.

Les deux entrées du contrôleur flou erreur de la position (E_P) et l'erreur de l'angle (E_G), sont évaluées par rapport aux sous ensembles flous à partir de cinq et sept fonction d'appartenance respectivement, sur des univers de discours normalisés, avec un chevauchement de 50% (**figure III-2**).

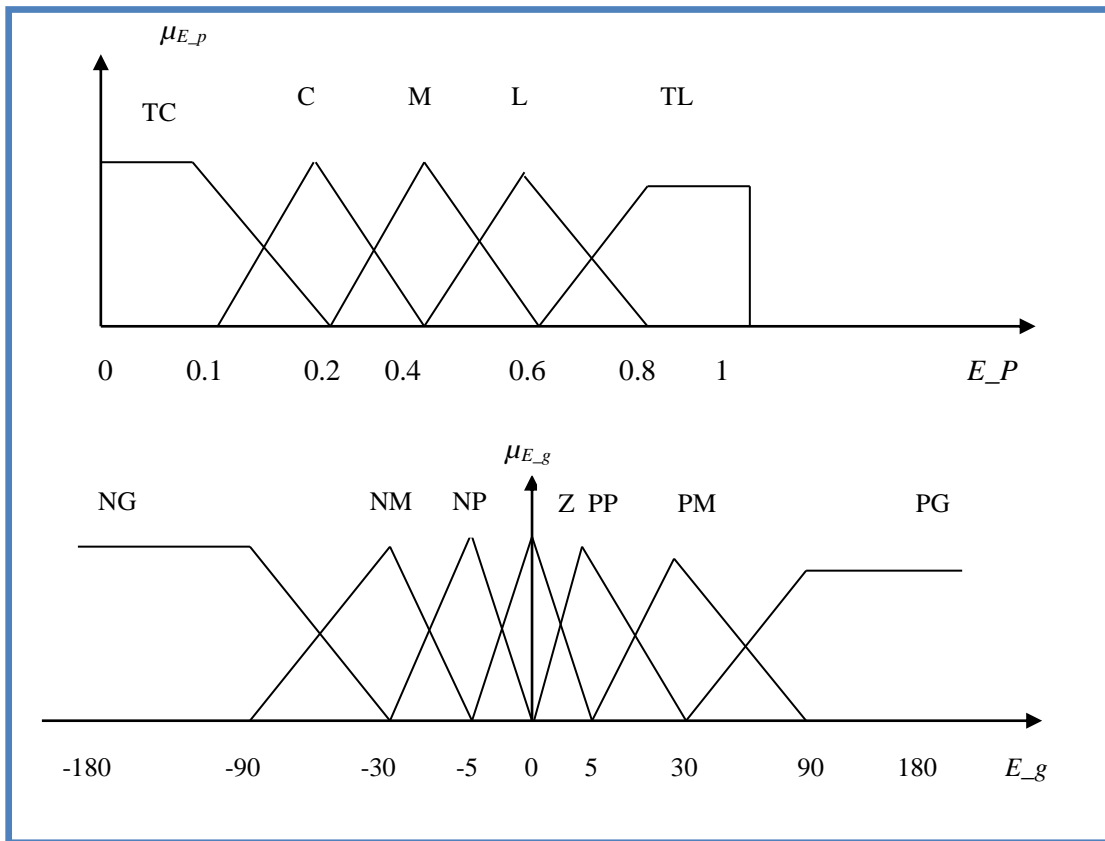


Figure III-2 : les fonctions d'appartenances des entrées de l'erreur de la position et l'erreur de l'angle

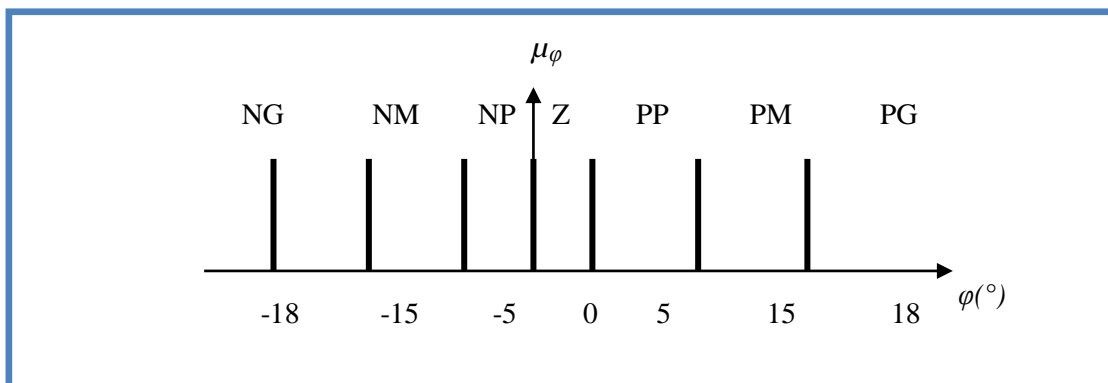


Figure III-3 : fonction d'appartenance pour la sortie $\varphi(^{\circ})$ l'orientation

Les variables linguistiques peuvent être décrites par rapport aux sous-ensembles L_{E_p} et L_{E_g} comme suit :

$L_{E_p} = \{TC, C, M, L, TL\}$.

$L_{E_g} = \{NG, NM, NP, Z, PP, PM, PG\}$.

C : Court, L : Large,

G : Grand, M : Moyenne,

P : Petite, Z : Zéro,

NM : Négative Moyenne,

NP : Négative Petite,

PP : Positive Petite,

PM : Positive Moyenne,

PG : Positive Grand,

TG : Très Grand,

L'Inférence :

Dans cette étape, nous allons baser sur une matrice d'inférence (tableau III-1). L'inférence se fait donc sur la base de matrice que l'on vient de décrire. On commence par utiliser un opérateur logique pour définir la description symbolique associée à la prémisse de la règle ; c'est-à-dire réaliser le « ET ». On passe ensuite à l'inférence proprement dite qui consiste à caractériser la variable floue de sortie pour chaque règle. C'est l'étape de la conclusion

« ALORS ».

E_g \ E_p	NG	NM	NP	Z	PP	PM	PG
TC	PM	PP	Z	Z	Z	NP	NM
C	PG	PG	PM	Z	NM	NG	NG
M	PM	PM	PP	Z	NM	NG	NG
L	PM	PP	PP	Z	NP	NP	NM
TL	PM	PM	PP	Z	NP	NM	NM

Tableau III-1 : la base de règles de contrôleur flou.

La défuzzification :

La dernière étape d'un système d'inférence flou est la défuzzification qui extrait une grandeur physique à la sortie du régulateur, nous avons choisi la méthode dite du « centre de gravité » qui consiste à calculer le centre de gravité des $\Delta\theta$ pondérés par leurs degrés d'appartenance.

Remarque :

D'un point de vue pratique, on peut résumer dans les points suivants les éléments qui ont peu d'importance sur le comportement global du régulateur flou [LUT 96] :

- La forme des fonctions d'appartenance, d'où le choix de formes triangulaires à cause de la simplicité de mise en œuvre.
- Le choix des fonctions pour réaliser les opérateurs et le mécanisme d'inférence (agrégation des règles :SI- Alors).
- Le choix de la méthode de défuzzification

III.3 Optimisation par les algorithmes génétiques :

Le but des algorithmes génétiques n'est pas de trouver, le plus rapidement possible, la meilleure solution. Leur objectif est de transformer un ensemble de solutions éventuelles, appelées individus, formant une population, en une nouvelle population (génération) « meilleure ». Pour résoudre un problème, il est donc nécessaire de coder les individus sous la forme d'une « chaîne de symboles ». Cette dernière est similaire à un génome constitué d'une chaîne de gènes. Un algorithme génétique générique a la forme suivante :

Algorithme_ génétique (n=taille_ population, k=nb_ gène, nb_ génération, PC,PM)

Début

 Initialisation de n individus de k gènes;

 Pour g=1 à nb_ génération

 Evaluation de la population ;

 Reproduction par sélection ;

 Croisement suivant PC ;

 Mutation suivant PM ;

 Mise à jour de la population courante ;

 Fin Pour

Fin

Pour la première génération : la population initiale, les individus sont généralement créés au hasard. Une fois les individus générés, il est nécessaire d'évaluer chaque individu. Une fonction d'évaluation (FE ou fitness function) doit donc être définie.

Ainsi, plus la valeur de la FE d'un individu est élevée, plus ses allèles (valeurs de ses gènes) sont pertinents. Cette évaluation se fait le plus souvent à l'aide de plusieurs critères. Dans un deuxième temps, les couples d'individus (parents) qui pourront se reproduire sont sélectionnés.

Les principes généraux de croisement et de mutation sont les suivants : pour le croisement, deux individus sont sélectionnés suivant la probabilité PC. Ces deux parents seront donc combinés pour générer deux enfants. Ils peuvent être combinés par des croisement mono ou multipoints. Dans tous les cas, les enfants posséderont une partie des gènes d'un parent et une partie de l'autre. Pour la mutation, un individu est sélectionné, un ou plusieurs de ses gènes est transformé suivant la probabilité PM .

Pour notre travail on a choisi l'optimisation des règles floues par un Algorithme Génétique.

III.3.1 Optimisation des règles floues par un Algorithme Génétique :

Les règles floues permettent d'interpréter les variables linguistiques qui caractérisent les entrées du contrôleur. Plusieurs auteurs se sont penchés sur le problème de l'optimisation des règles d'inférence utilisées dans un contrôleur flou.

Les règles floues du notre contrôleur possèdent deux prémisses et une seule conclusion :

R_{i_1, i_2} : **Si** (X1 est A_{1i_1}) **et** (X2 est A_{2i_2}) ; **alors** (Y est B_j)

$i_1 = 1, \dots, N_{te1}$, $i_2 = 1, \dots, N_{te2}$, $j = 1, \dots, N_{ts}$

Où N_{te1} est le nombre des termes linguistiques pour l'entrée 1.

N_{te2} est le nombre des termes linguistiques pour l'entrée 2.

N_{ts} est le nombre des termes linguistiques pour la sortie.

où $A_{11}, \dots, A_{1N_{te}}$ sont les termes linguistiques qualifiant la première entrée X_1 , $A_{21}, \dots, A_{2N_{te}}$ sont les termes linguistiques pour la deuxième entrée X_2 et $B_1, \dots, B_{N_{ts}}$ sont les termes linguistiques qui qualifient la sortie Y.

On peut définir au maximum $N_{te1} \times N_{te2}$ règles floues pour ce contrôleur. Parmi toutes les règles possibles, certaines ne sont pas utiles. Afin de représenter les règles non utilisées, on choisit de les marquer en considérant que la sortie n'est pas qualifiée. Cela correspond à ajouter un $N_{ts} + 1$ ème terme linguistique pour qualifier la sortie, auquel on affecte de façon conventionnelle un indice zéro. La conclusion d'une règle repérée par les indices i_1 et i_2 est alors représentée par un nombre $R_{i_1 i_2}$ défini selon :

$$R_{i_1 i_2} = \begin{cases} 0 & \text{si la règle } (i_1, i_2) \text{ est inutile} \\ B_j & \text{si la conclusion de la règle } (i_1, i_2) \text{ est } B_j \end{cases}$$

A. Codage

Le codage est le premier choix à faire. Afin de permettre un traitement par un algorithme génétique, la représentation des règles (base de règles) décrite précédemment doit être codée pour être insérée dans un chromosome.

Le codage le plus simple consiste à représenter les règles par une chaîne binaire, chaque élément de la représentation étant codé sur 8 bits.

La population initiale est initialisée de manière aléatoire, on définit simplement la fitness comme étant la valeur d'individu, plus la valeur de l'individu sera élevée plus la fitness sera grande.

Exemple :

Individu	Code binaire	fitness
Indiv 1	00010101	21
Indiv 2	00100010	34
Indiv 3	01001010	74

B. sélection :

Le rôle de la sélection est de choisir parmi tous les individus d'une population les parents qui assureront la reproduction. Ce choix est réalisé par tirage au sort parmi les individus, en tenant compte d'une probabilité de sélection affectée à chacun d'eux. Un individu a d'autant plus de chances d'être sélectionné que sa fonction d'évaluation prend une valeur importante. Pratiquement, la probabilité p_i d'évolution d'un individu c_i est définie par :

$$P_i = \frac{F(c_i)}{\sum_{j=1}^N f(c_j)} \dots\dots\dots(III-1)$$

Où $f(c_i)$ est la fonction d'évaluation (fitness) de l'individu c_i et $\sum_{j=1}^N f(c_j)$ est la somme de toutes les fonctions d'évaluation des individus.

La méthode souvent utilisée pour sélectionner les individus assurant la reproduction est la roulette de casino. Chaque individu occupe un secteur de la roulette dont l'angle est

proportionnel à sa probabilité de sélection(P_i). On constate aisément que les individus les plus performants ont plus de chances d'être sélectionnés puisqu'ils occupent une surface plus importante de la roulette.

Cela peut être simulé par l'algorithme suivant :

- On calcule la somme $S1$ de toutes les fonctions d'évaluation d'une population.
- On génère un nombre ' r ' entre 0 et $S1$.
- On calcule ensuite une somme $S2$ des évaluations en s'arrêtant dès que r est dépassé.
- Le dernier chromosome dont la fonction d'évaluation vient d'être ajoutée est sélectionné.

entier roulette de casino (valu : tableau de réel)

Début

$S1 =$ Somme de toutes valeurs d'adaptation du population

$r =$ réel tiré au hasard entre 0 et $S1$. // on tire une valeur au hasard

$i = 1$

$S2 = \text{valu} [i]$ // $S2$ est somme partielle

tant que ($r > S2$)

$i = i+1$

$S2 = S2 + \text{valu} [i]$

fin tant que

retourne i

Fin.

c. croisement :

L'étape de sélection a permis de choisir parmi une population les individus les plus aptes à se reproduire. Ensuite, ces individus sont regroupés par paires qui constitueront les parents de la génération suivante. Durant l'étape de reproduction, les chaînes binaires qui caractérisent deux parents sont utilisées pour créer les chaînes binaires qui caractérisent deux enfants.

On applique l'opérateur de croisement en utilisant un seul point de crossover. Normalement chaque couple donne deux enfants qu'on ajoute à notre nouvelle population P' .

Dans un algorithme génétique, on définit la probabilité de croisement pc . Le croisement de deux chromosomes est réalisé de la façon suivante :

$$\begin{aligned} \text{Fils 1} &= pc (\text{parent1}) + (1-pc) \text{parent2} \dots\dots\dots \\ \text{(III-2)} \\ \text{Fils 2} &= pc (\text{parent2}) + (1-pc) \text{parent1} \end{aligned}$$

D. Mutation :

Une mutation correspond à une inversion de la valeur d'un bit de la chaîne binaire. Elle permet de modifier de façon tout à fait aléatoire le chromosome d'un individu. Toutefois, pour ne pas trop perturber l'évolution globale de la population, la probabilité Pm d'apparition d'une mutation doit rester très faible.

Dans notre exemple, si on utilise une probabilité de mutation Pm , on doit s'attendre à obtenir une inversion d'environ k bits à chaque génération. Puisque la population comporte N individus caractérisés par 8 bits, on tire au sort un nombre compris entre 0 et 1, et on inverse la valeur du bit si ce nombre est inférieur à Pm .

$$K = Pm \times \text{nombre des bits pour chaque individu} \times \text{nombre des individus pour chaque génération.}$$

Où K est le nombre des bits a mutés pour chaque génération.

Après avoir la structure de le contrôleur flou (fuzzification , fonction d'appartenances , base de règles , défuzzification) et l'optimisation de base de règles par les algorithmes génétiques (codage binaire ,....) on va appliquer ces méthodes sur un robot mobile tricycle.

But: on cherche à expérimenter un apprentissage par algorithme génétique de l'ensemble des données liées au problème du suivi de trajectoire par un robot mobile.

Donc pour trouver la trajectoire il faut calculer la position et l'orientation du robot alors de commander soit les vitesses ou l'angle d'orientation ou les deux. Pour la commande du robot de tricycle on va contrôler l'angle ou la variation de l'angle, nous supposons que la vitesse linéaire est constante donc on va jouer sur l'angle.

Les entrées et sorties de notre contrôleur flou qu'on a défini précédemment sont calculées par les équations de mouvement d'un robot mobile de type tricycle avec deux roues arrière motrices indépendantes et une roue folle à l'avant.

III.4 Modèle géométrique du robot tricycle:

Deux roues motrices, dont la direction reste fixe par rapport au robot, sont placées à l'arrière, une roue d'orientation (roue folle) totalement libre est située à l'avant du robot pour assurer la stabilisation du robot.

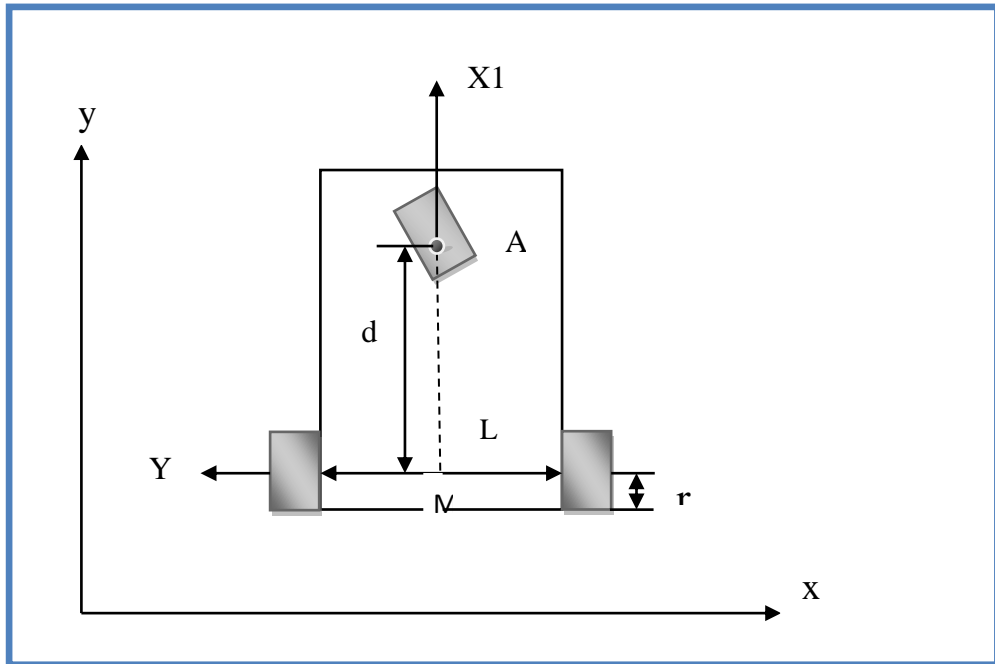


Figure III-4 : Modèle géométrique d'un robot mobile tricycle

r : est le rayon des roues motrices.

L : la longueur de l'essieu.

d : la distance entre la roue avant et le centre de l'essieu des roues arrière.

Les équations de mouvement du robot par rapport au centre d'inertie le milieu de l'essieu des roues arrière sont :

$$\Delta E\theta_n = \frac{\Delta q_d - \Delta q_g}{L} \dots\dots\dots (III-3)$$

$$\Delta D = \frac{\Delta q_d + \Delta q_g}{2} \dots\dots\dots (III-4)$$

$$\theta_{n+1} = \theta_n + \Delta\theta_n$$

$$x_{n+1} = x_n + \Delta D \cos(\theta_n + \Delta\theta_n) \dots\dots\dots (III-5)$$

$$y_{n+1} = y_n + \Delta D \sin(\theta_n + \Delta\theta_n)$$

x_n, y_n, θ_n : les coordonnées de la position et de l'orientation de point M à l'instant t_n dans le repère absolu (OXY).

$\Delta\theta_n, \Delta D$: la variation de l'orientation et de la distance du robot parcourue par le point M entre les instants t_n et t_{n+1} .

$\Delta q_d, \Delta q_g$: les distances parcourues par le centre de chacune des roues droit et gauche entre les instants t_n et t_{n+1} .

L : la distance entre les roues.

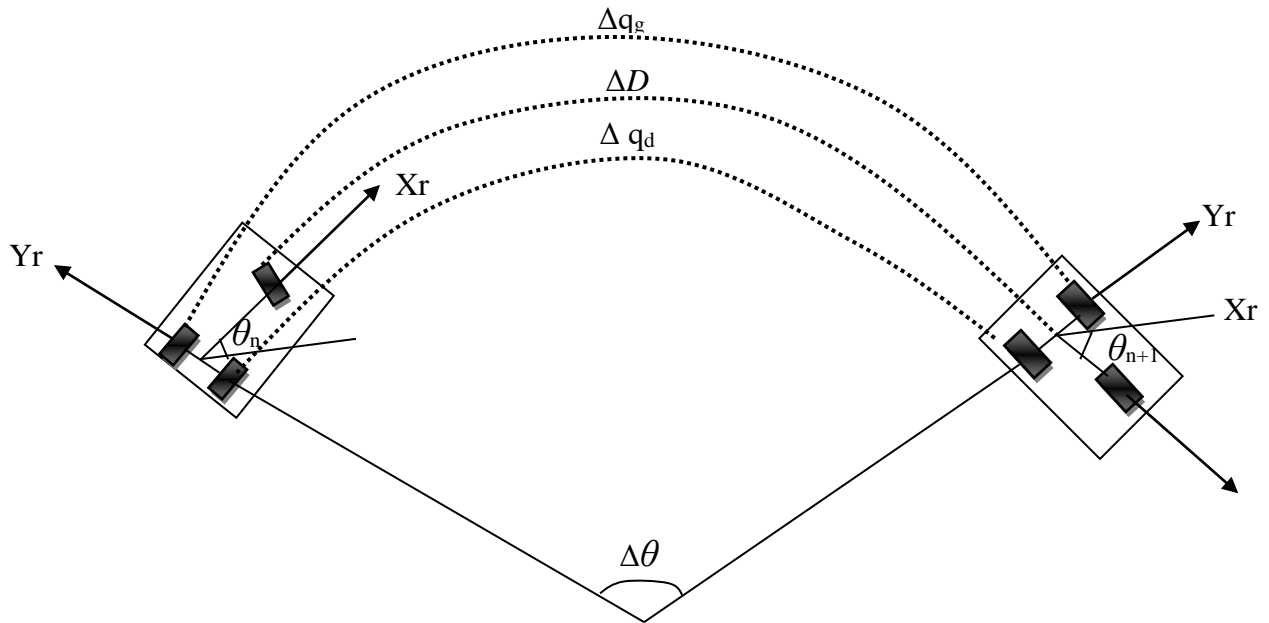


Figure III-5 : principe de Les équations de mouvement

On appliquant cette commande sur le robot mobile de type tricycle ses paramètres sont :

$$R=0.1,$$

$$L=0.2$$

$$d=0.15,$$

III.5 conclusion :

Dans ce chapitre, nous avons présenté le contrôleur flou et l'optimisation de leur base de règles par un algorithme génétique. On a appliqué ce contrôleur sur un robot mobile tricycle pour tester le fonctionnement de ce contrôleur.

Dans le chapitre suivant nous présentons l'application développée et les différentes composantes de l'application, ainsi que les diverses techniques implémentées.

CHAPITRE IV

Présentation de simulation



IV.1 Introduction :

Le but de notre projet est de réaliser un contrôleur flou optimisé par les algorithmes génétiques. Nous présentons dans ce chapitre la description détaillée de l'application (les différentes composantes de l'application, ainsi que les diverses techniques implémentées).

IV.2 Environnement du travail

Ce logiciel a été développé sur une machine de type Intel Pentium IV, dans l'environnement WINDOWS XP, avec l'utilisation de *Delphi 7* comme un langage de programmation, car en plus de sa facilité de programmation, il offre une interface conviviale à notre logiciel.

IV.3 Présentation du logiciel :

Dans la figure suivante (figure IV-1), on trouve la fenêtre principale de notre simulateur.

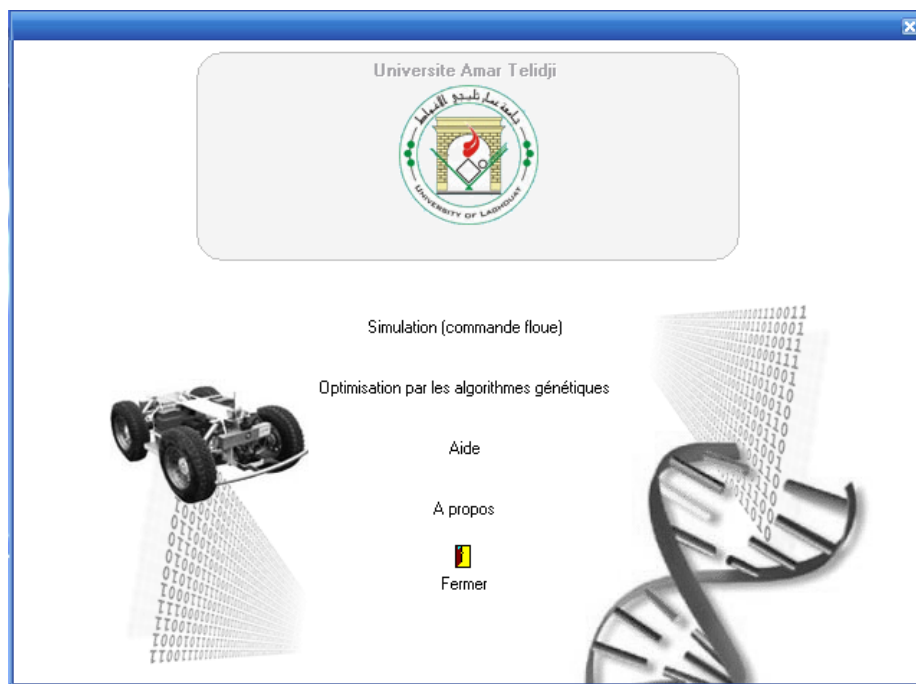


Figure IV -1 : la fenêtre principale du logiciel

IV.3.1 Menus du logiciel

Le système de menus déroulant donnant un accès simple à toutes les fonctions du système.

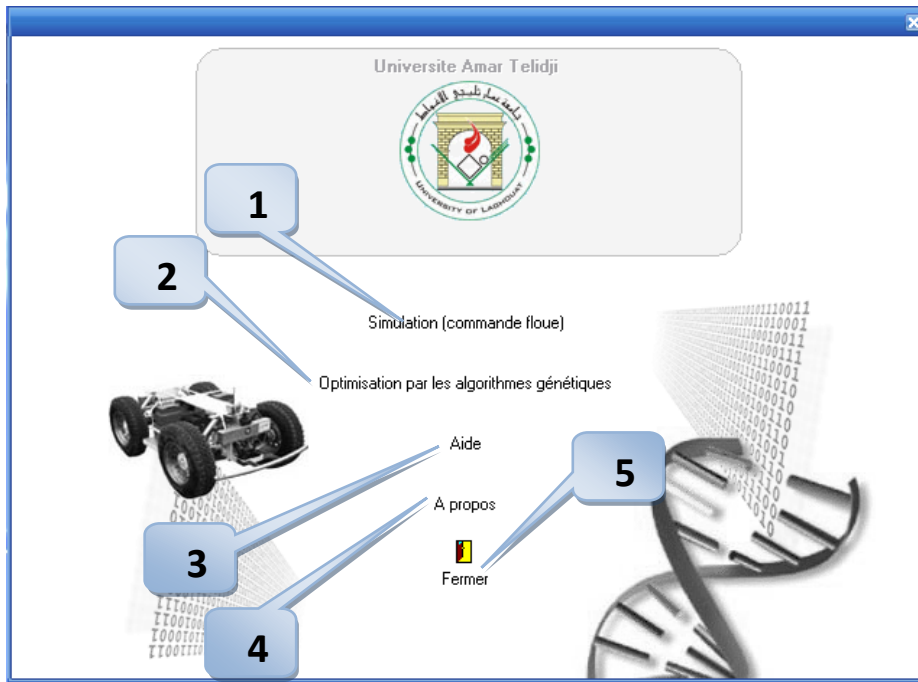


Figure IV-2 : Menus du logiciel

- 1** : Simulation par commande floue ;
- 2** : Optimisation par les algorithmes génétiques (application d'algorithme génétique sur le contrôleur flou) ;
- 3** : C'est l'aide (aide sur notre logiciel) ;
- 4** : À propos (quelques détails sur notre logiciel) ;
- 5** : Fermer (fermer la fenêtre) ;

Après la définition des différents composants du système, on va détailler les fonctionnalités de chaque composant :

IV.3.1.1 le menu simulation par la commande floue :

Pour l'initialisation :

Ce menu est pour initialiser le comportement de la simulation de notre contrôleur flou :

Après qu'on a cliqué sur le bouton numéro (1) dans la figure IV-2 on obtient la figure suivante : (figure IV-3)

La figure suivante présente l'initialisation de la commande floue :

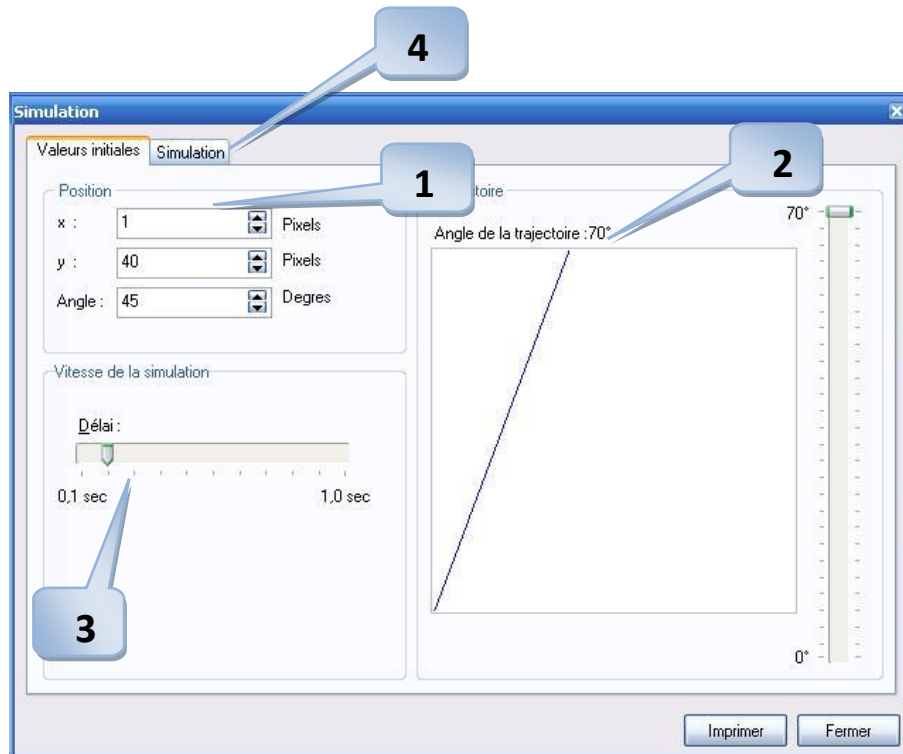


Figure IV-3 : l'initialisation de la commande floue

Cette figure est divisée en trois parties principales :

- 1** : Configuration initiale (x, y, Angle) pour le robot mobile (l'utilisateur a le droit de modifier les coordonnées initiaux du robot) ;
- 2** : Pour définir la trajectoire à suivre ;
- 3** : La vitesse de la simulation : 0.1 sec c- à-d plus vite ; 1 sec c.-à-d. plus faible ;
- 4** : La simulation.

On commence par l'initialisation de la position du robot mobile ; on saisie sa position initiale (x, y, angle) dans la partie position (numéro 1) , en suite, on glisse la barre de l'angle (numéro 2) pour définie le trajectoire à suivre. On peut modifier la vitesse de la

simulation : on glisse la barre de la vitesse (numéro 3) 0.1 seconde c à d plus vite et 1 seconde c à d plus lente.

Après qu'on a défini la position initiale et la trajectoire à suivre, on commence la simulation par la commande floue, on clique sur le bouton numéro (4) et on obtient la figure suivante : (figure IV- 4).

Pour la simulation :

Cette figure présente la fenêtre résultat de la simulation :

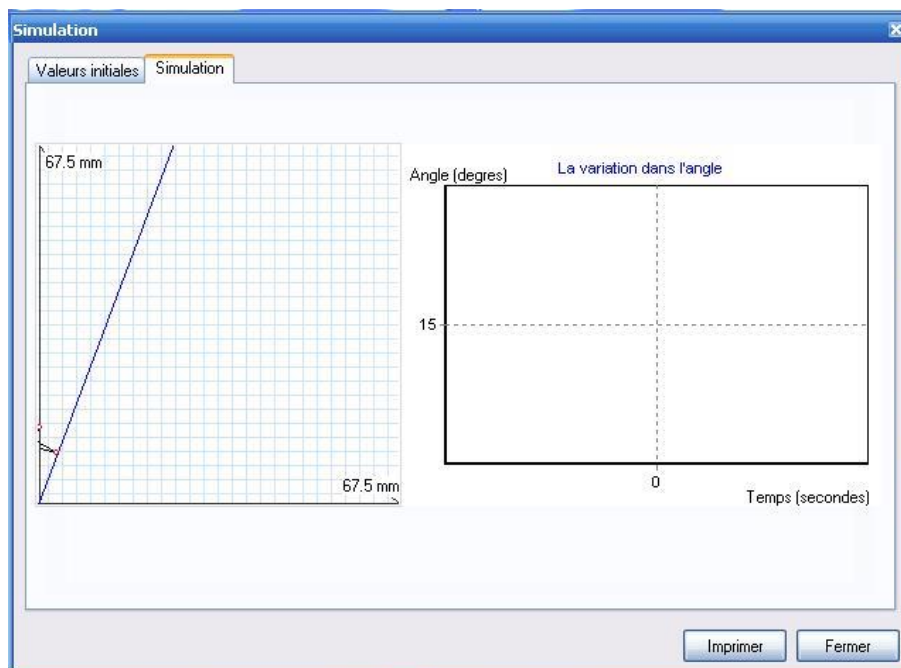


Figure IV-4 : la simulation par la commande floue

Lorsqu'on commence la simulation, le robot est initialement positionné par les coordonnées (x, y, angle). Dans le plan, on trouve la trajectoire définie et les traces du mouvement du robot mobile ; le graphe sous le nom « la variation dans l'angle » est pour présenter la variation de sortie de la commande floue selon le temps.

IV.3.1.2 le menu optimisation pour les algorithmes génétiques: (application d'algorithme génétique sur le contrôleur flou) :

Pour commence la simulation de l'optimisation par les algorithmes génétiques on doit initialiser le comportement de la simulation. Après qu'on a cliqué sur le bouton numéro (2) dans la figure IV-2 on obtient la figure suivante : (figure IV-5)

La figure suivante présente fenêtre d'initialisation de l'optimisation par les algorithmes génétiques :

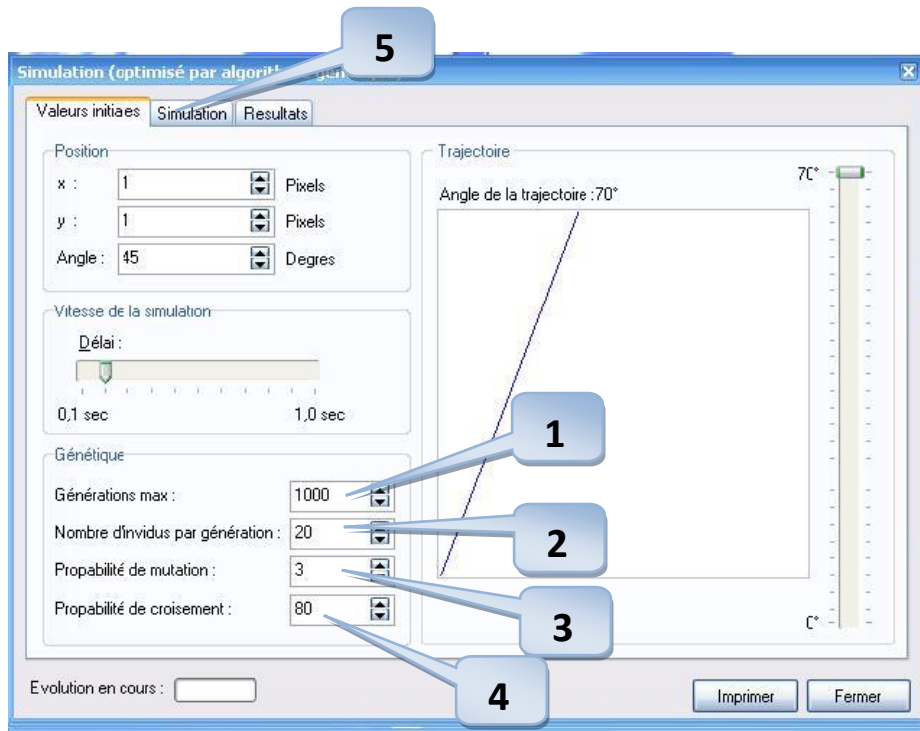


Figure IV-5 : l'initialisation de la simulation par les algorithmes génétiques

1 : Définition du nombre de générations pour la simulation par les algorithmes génétiques (teste d'arrêt pour l'optimisation).

2 : Pour définir le nombre d'individus pour chaque génération

3 : Probabilité de mutation.

4 : Probabilité du croisement.

On commence par l'initialisation du nombre maximal pour les générations, on saisie ce nombre dans la partie générations max (numéro 1); en suite, on saisie le nombre d'individus par chaque génération (numéro 2), la probabilité de mutation (numéro 3) avec une valeur appartenant à l'intervalle [1, 10], et la probabilité de croisement (numéro 4) avec une valeur appartenant à l'intervalle [10, 100].

Après qu'on a défini le nombre des générations max, le nombre d'individus, et les probabilités pour la mutation et le croisement, on commence la simulation par les algorithmes génétiques. On clique sur le bouton numéro (5) et on obtient la figure suivante : (figure IV- 6).

La fenêtre de la simulation:

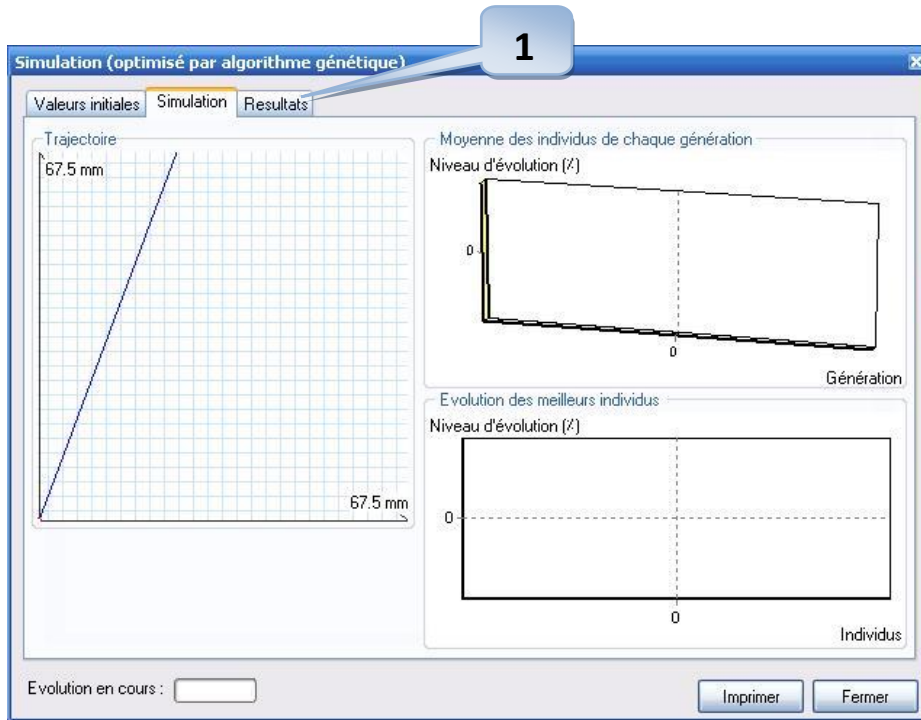


Figure IV-6 : la simulation par les algorithmes génétiques

Le graphe « moyenne des individus » représente les moyennes des individus par génération, chaque génération est évaluée selon les moyennes de leurs individus. En revanche, le graphe « évaluation des meilleurs individus » représente la valeur du meilleur individu trouvé.

On clique sur le bouton numéro (1) et on obtient la figure suivante : (figure IV- 7).

Résultat de la simulation par les algorithmes génétiques :

La fenêtre résultat de l'optimisation par les algorithmes génétiques est présentée dans la figure suivante :

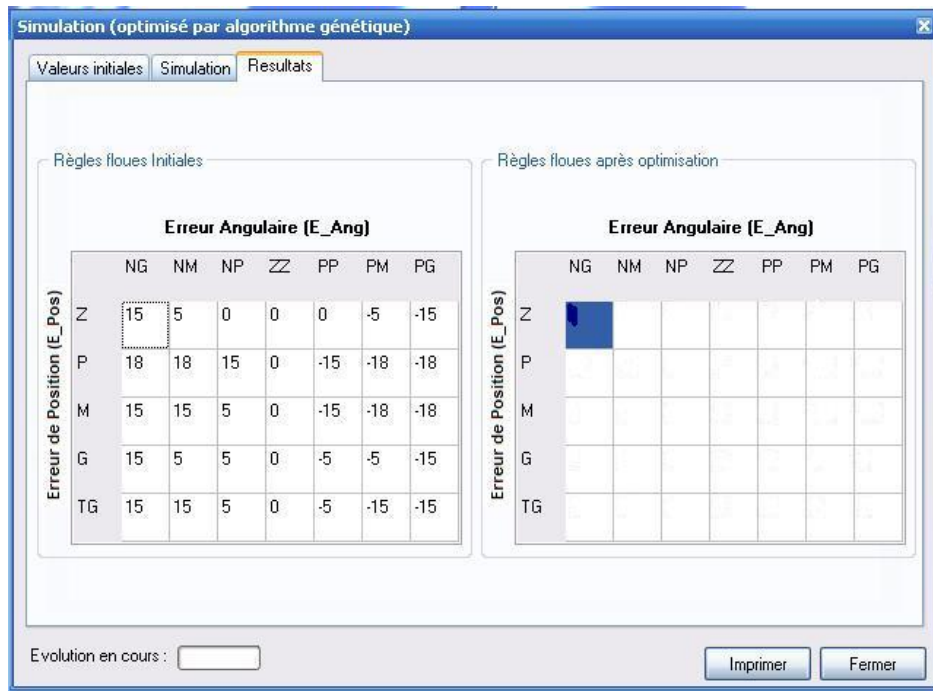


Figure IV-7 : le résultat de la simulation par les algorithmes génétiques

Cette figure présente deux matrices : la première est la matrice de la base des règles initiales ; la deuxième est pour afficher le résultat (base de règles finales optimisées).

IV.3.2 exemple d'application :

a) La simulation par la commande floue :

initialisation : si on prend $x=10$, $y=70$, $\text{angle}=45$, et l'angle pour la trajectoire à suivre est 30° (voir Figure IV-8), on obtient les résultats ci-dessous (voir Figure IV-9).

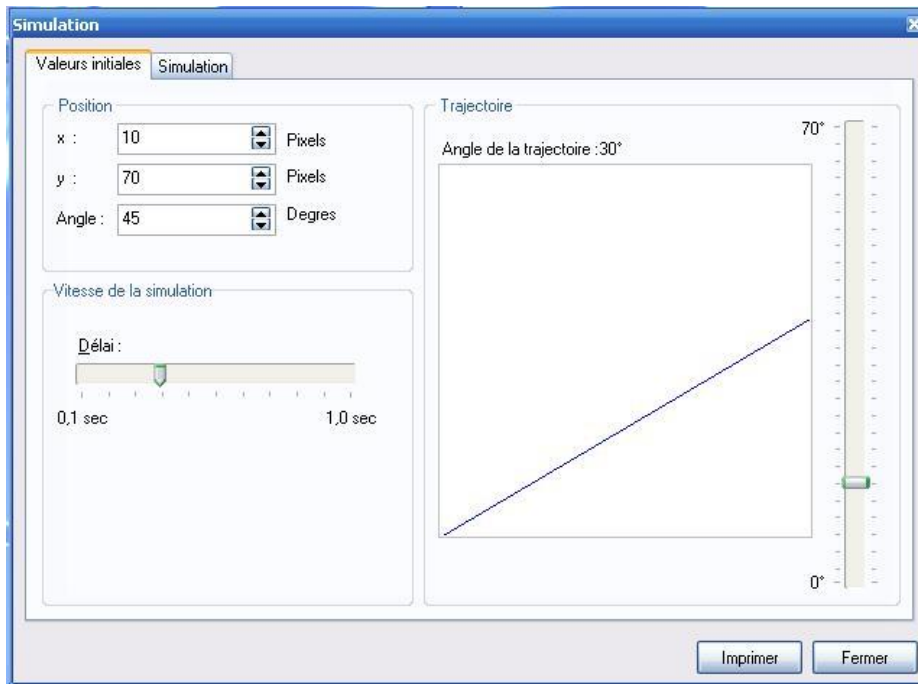


Figure IV-8 : exemple de l'initialisation de la commande floue

La figure suivante représente les résultats de la simulation ; on peut constater comment le robot se déplace selon la variation de l'angle qu'est le résultat du contrôleur flou.

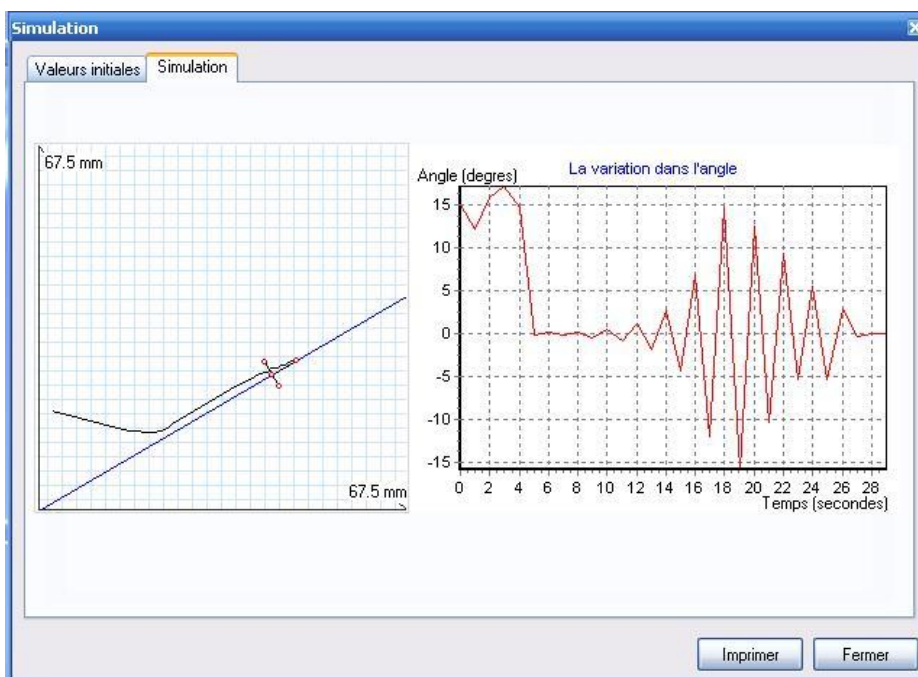


Figure IV-9 : exemple de la simulation par la commande floue

Le graphe « la variation dans l'angle » montre la variation de l'angle selon le temps. A chaque instant, le robot modifie son angle de déplacement en fonction de la variation de l'angle (comme précisé dans le graphe). L'angle dans cet exemple varie entre 15° et -15° ; ces valeurs sont changeables selon les valeurs initiales de l'initialisation du contrôleur.

b) La simulation par les algorithmes génétiques :

initialisation pour les algorithmes génétiques :

On prend le point initiale du robot $x=10$; $y=70$; $\text{angle}=45^\circ$, et pour l'angle de la trajectoire $=30^\circ$. Pour les algorithmes génétiques, on prend pour le nombre maximum de génération= 1000 ; le nombre des individus pour chaque génération= 20 ; la probabilité de la mutation $=3$; (c-à-d $3 / 1000$ parce que la probabilité de la mutation appartient à l'intervalle $[0.001 ; 0.01]$), la probabilité du croisement $=80$ (c-à-d $80/100$ parce que la probabilité du croisement appartient à l'intervalle $[0.1 ; 1]$).

Comme l'indique la figure suivante :

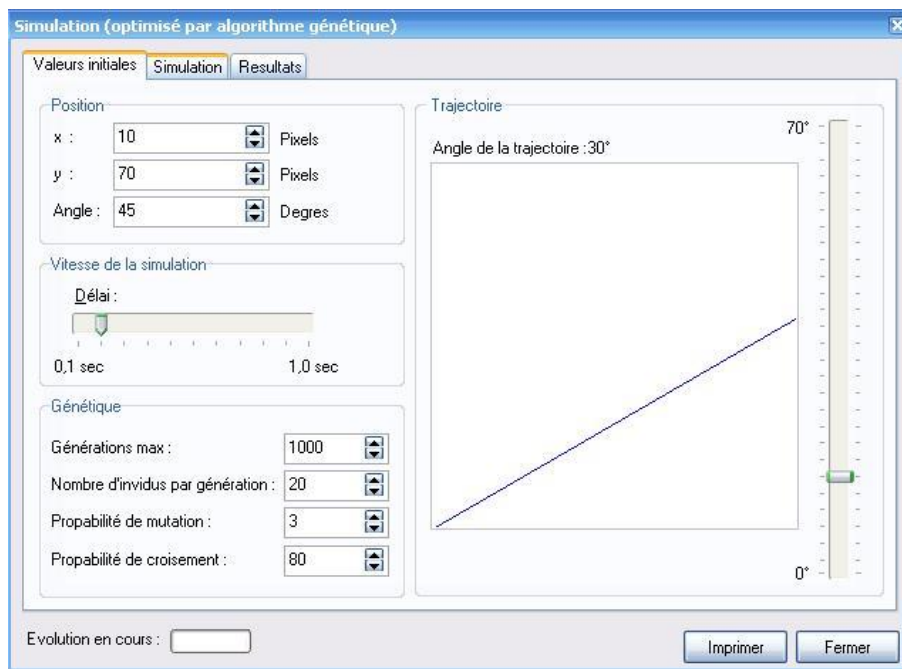


Figure IV-10 : exemple d'initialisation pour les algorithmes génétiques

Résultat de la simulation :

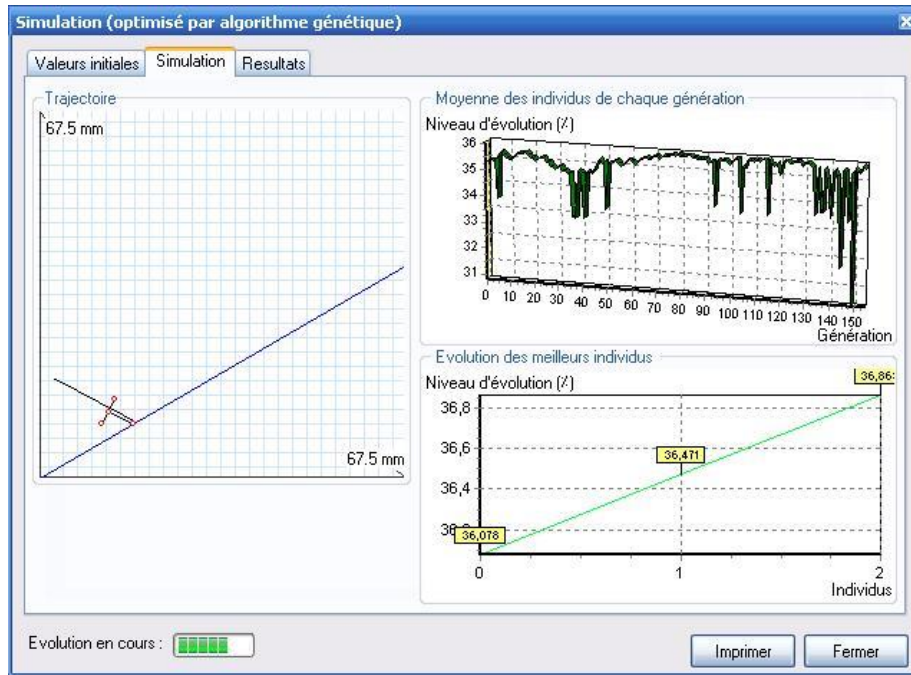


Figure IV-11 : exemple de la simulation par les algorithmes génétiques

On trouve pour le graphe de « moyenne des individus de chaque génération » que les moyennes des individus varient entre 31% comme valeur minimale et 36% comme valeur maximale. La génération qui contient la valeur maximale (36%) est la meilleure génération.

Pour le deuxième graphe « évolution des meilleurs individus », chaque fois qu'on trouve un meilleur individu, on affiche ses valeurs. Par exemple dans la Figure IV-11, on voit que l'individu avec une valeur de 36,86% est le meilleur individu de sa génération.

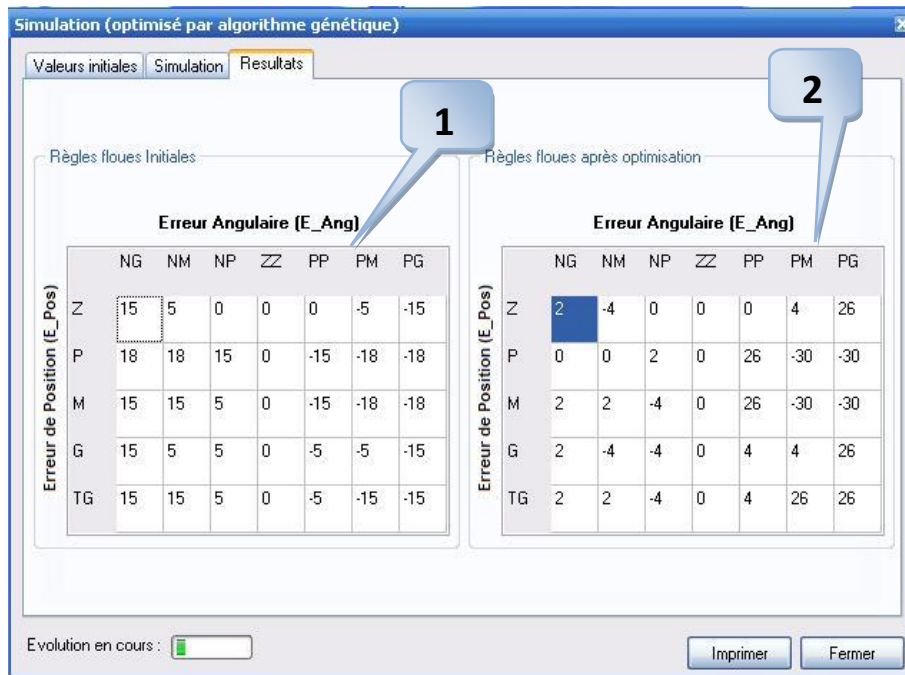


Figure IV-12 : exemple le résultat de la simulation par les algorithmes génétiques.

Lorsqu'on termine la simulation de l'optimisation par les algorithmes génétiques, on obtient le résultat sous forme d'une matrice de base de règles optimisées. Par contre, les règles inutiles seront remplacées par des valeurs nulles comme le montre la figure IV-12.

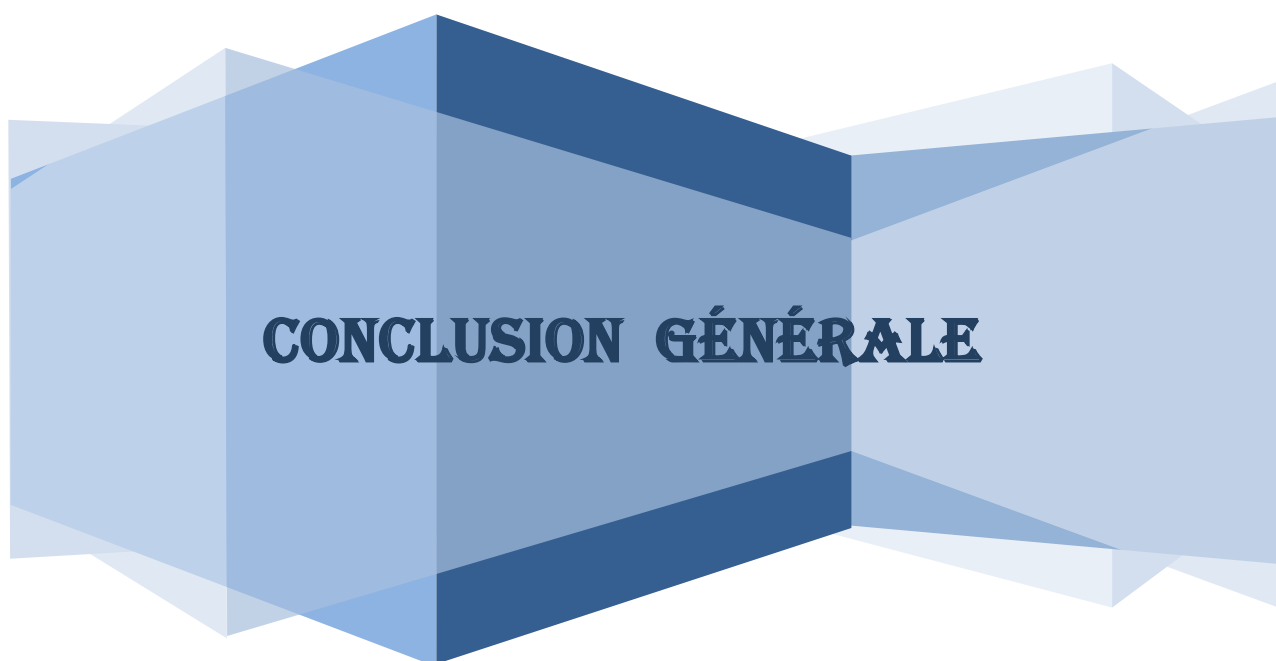
1 : La base des règles initiale.

2 : La base de règles finale (après l'optimisation).

IV.4 Conclusion :

Dans ce chapitre on a présenté le simulateur qu'on a réalisé. Notre logiciel est simple et flexible pour l'opérateur. Il comporte les méthodes et les fondements que nous avons étudiés et présentés dans ce mémoire, dont principalement: la logique floue, les algorithmes génétiques, et sans oublier aussi le modèle et les équations de mouvement pour le robot mobile tricycle.

En outre, l'objectif principal de notre travail est de réaliser un logiciel présentant un contrôleur flou optimisé par les algorithmes génétiques. Cet objectif a été réalisé avec beaucoup de succès et nous croyons avoir même dépassé cet objectif vers une étape plus avancée qu'illustre l'optimisation et la génération d'une base de règles.



CONCLUSION GÉNÉRALE

L'objectif principal de notre travail était de réaliser un logiciel qui fait l'optimisation d'un contrôleur flou en utilisant les algorithmes génétiques qui a été réalisé avec succès, et nous avons choisi comme une application le suivi d'une trajectoire robot mobile tricycle.

À cet effet, nous avons commencé par étudier la logique floue, la théorie des ensembles flous et les algorithmes génétiques qui sont des outils efficaces pour une classe de problèmes très large.

Ensuite, nous avons présenté en détails les paramètres du contrôleur flou et comment faire l'étape d'optimisation de la base de règles de cet contrôleur par les algorithmes génétiques : le codage, les operateurs génétiques comme la sélection, croisement et la mutation, et nous avons appliqué ce contrôleur sur un robot mobile tricycle pour tester leur fonctionnement.

Nous avons décomposé notre application en deux parties : la première, nous avons utilisé la commande floue sans appliquer les algorithmes génétiques a pour but de comprendre les différentes étapes de la commande floue, les résultats sont bons .Mais quand nous avons appliqué les algorithmes génétiques dans la deuxième partie en remarquant que les résultats sont mieux par apport les autres (le temps, la base de regèles).

Notre outil est caractérisé par une interface graphique claire, et simple à utiliser, et qui permet aux utilisateurs un grand degré d'interaction à l'aide des animations et des messages explicatifs.

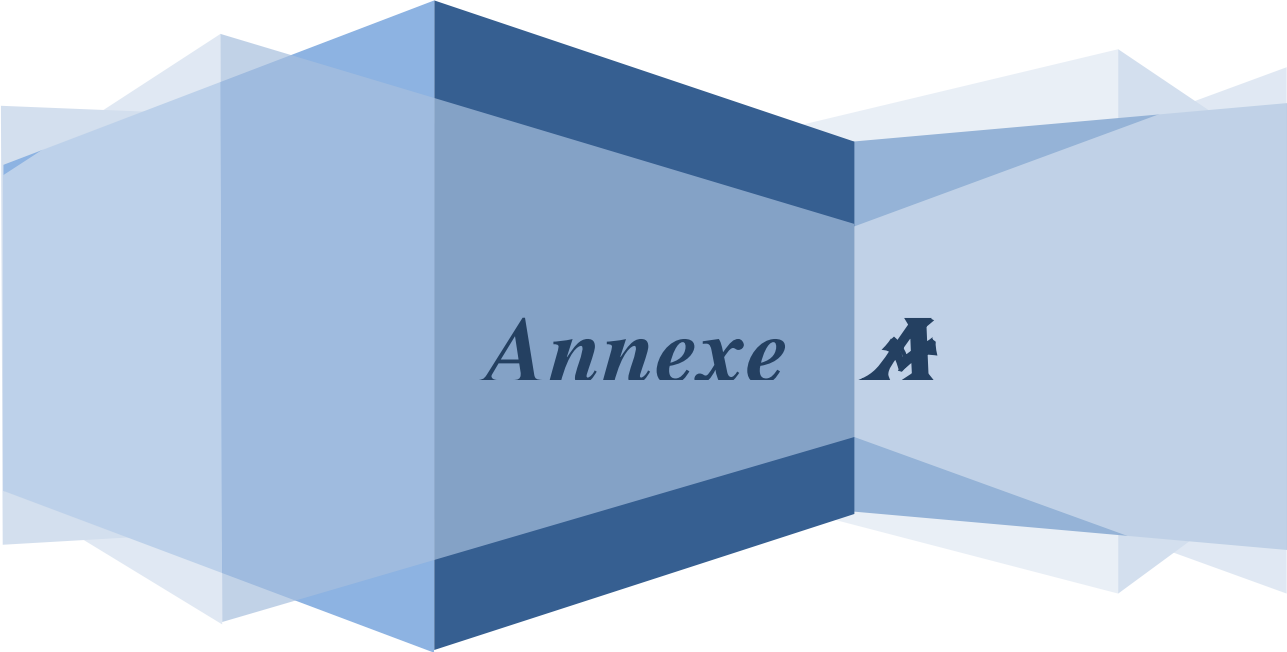
Ce travail nous a permis de :

- ✓ De bien comprendre la logique floue et les algorithmes génétiques.
- ✓ Bien comprendre le problème d'optimisation de la base de règles d'un contrôleur flou.

- ✓ Avoir une bonne maîtrise de logiciel de programmation (Borland Delphi 7).
- ✓ D'avoir une bonne expérience dans le domaine de recherche.

À ce point, nous proposons les extensions futures suivantes :

- ✓ Proposer des améliorations pour le logiciel à fin de traiter d'autre cas d'optimisation comme
 - Les fonctions d'appartenance des termes linguistiques permettant la qualification des entrées du contrôleur.
 - Les fonctions d'appartenance des termes linguistiques décrivant l'état des sorties.
 - Les paramètres définissant la méthode de défuzzification.



ANNEXE A

Notions fondamentales du schéma

Holland (1975) a introduit la notion de schémas qui considère qu'un algorithme génétique manipule implicitement des similarités entre codes. Dans le modèle de représentation binaire, un schéma (ou un motif de similarité) est un outil d'identification de similarités entre des chaînes de longueurs finies et définies sur un alphabet fini. Dans ce qui suit, on se limite à des chaînes définies sur un alphabet binaire $\{0,1\}$, ce qui ne nuit pas à la généralisation de la discussion. Puisque chaque bit dans le codage binaire d'une solution possible capture certaines informations sur le problème, les chromosomes avec des performances élevées sont similaires dans le sens qu'ils contiennent des 0 et des 1 dans des positions particulières qui codent les caractéristiques des performances élevées. Les schémas décrivent ces similarités dans les chaînes binaires à l'aide d'un alphabet ternaire $\{0,1,*\}$. Les nombres 0 ou 1 à une position donnée dénotent qu'il y a une valeur binaire fixe dans la chaîne. Le symbole * peut prendre indifféremment la valeur 0 ou 1 à la position où il se trouve.

Chaîne 1:	10110101
Chaîne 2:	01111000
Schéma:	**11**0*

Figure A.1: Exemple d'un schéma contenu dans deux chaînes binaires.

Deux propriétés importantes du schéma sont utilisées pour décrire et classer les similarités des chaînes. L'ordre d'un schéma S , dénoté par $o(S)$, est simplement le nombre de positions fixées dans la chaîne. Pour le schéma de l'exemple de la **figure A.1** est $o(**11**0*) = 3$. Une autre propriété du schéma est sa longueur utile. Pour un schéma S , la longueur utile $\delta(S)$ est la distance entre la première et la dernière position spécifiée dans le motif considéré. Pour le schéma de l'exemple de la **figure A.1** est $\delta(**11**0*) = 7 - 3 = 4$.

La performance d'un chromosome est mesurée par son aptitude (sa fonction d'évaluation). De la même manière, un schéma particulier est caractérisé par l'aptitude moyenne des chromosomes qui le contiennent. Comment la recherche se développe dépend des effets des opérateurs de reproduction contenus dans une population de chromosomes.

Effet de la sélection

L'importance du rôle de la sélection peut être mesurée par son effet sur la propagation d'un schéma S dans une population de N individus entre deux générations. On suppose que la sélection est du type proportionnel (c'est à dire une sélection aléatoire uniforme où les chromosomes sont sélectionnés avec une probabilité proportionnelle à leur aptitude) et que le nombre d'exemplaires du schéma S est égal à $m(S, t)$ dans la génération t . A la génération $t + 1$, le nombre attendu de représentants de S est estimé comme suit:

$$m(S, t+1) = m(S, t) \frac{f(S)}{f_{moy}} \dots\dots\dots (A.1)$$

où $f(S)$ et f_{moy} sont l'aptitude moyenne des individus représentant le schéma S à la génération t et l'aptitude moyenne de la population respectivement, et

$$f_{moy} = \sum_{j=1}^N f_j / N, \quad f_j \text{ est la fonction d'aptitude de la chaîne } j. \text{ L'équation (A.1)}$$

implique que les schémas ayant des aptitudes moyennes supérieures à celle de la population seront davantage copiés à la génération suivante et ceux avec des aptitudes moyennes inférieures à la moyenne le seront moins. Supposons qu'un schéma quelconque S reste au-dessus de la moyenne d'une quantité égale à cf_{moy} , où c est une constante, c'est à dire, $f(S) = f_{moy} + cf_{moy}$.

L'équation (A.1) qui traduit l'évolution des schémas peut alors être écrite comme

$$m(S, t+1) = (1+c) m(S, t) \dots\dots\dots (A.2)$$

Si l'aptitude d'un schéma particulier reste au-dessus ou au-dessous de la moyenne de la population par une certaine quantité (c a une valeur connue fixe), le taux de survie ou de décadence du schéma est exponentiel, et l'équation (A.2) devient

$$m(S, t+1) = (1+c)^t m(S, t) \dots\dots\dots (A.3)$$

Effet de croisement

Les effets de l'opérateur de croisement à un point sur un schéma particulier dépendent de sa distance utile. Un schéma survit si l'emplacement ou le point du croisement tombe à l'extérieur de sa longueur utile; mais, le schéma ne sera pas perturbé par l'opérateur du croisement, puisqu'il est possible que les deux parents puissent contenir ce schéma. Tous les emplacements de croisement possible peuvent être choisis avec la même probabilité; ainsi, plus la longueur utile est courte, plus la probabilité de survie du schéma est élevée. [Zak 07]

La probabilité de décadence ou de destruction d'un schéma S de longueur utile $\delta(S)$ lors d'un croisement à un point est donnée par

$$P_d(S) = \frac{\delta(S)}{L-1} \dots\dots\dots (A.4)$$

Où L est la longueur du chromosome.

Si le croisement a lieu avec une probabilité P_c , la probabilité de survie est alors

$$P_s(S) \geq 1 - P_c \frac{\delta(S)}{L-1} \dots\dots\dots (A.5)$$

Si on considère le fait que la sélection et le croisement soient des opérations indépendantes, l'effet combiné des deux opérations est obtenu en multipliant le nombre attendu du schéma lors d'une sélection seule par la probabilité P_s de survie lors d'un croisement

$$m(S, t + 1) \geq m(S, t) \frac{f(S)}{f_{moy}} \left[1 - P_c \frac{\delta(S)}{L-1} \right] \dots\dots\dots (A.6)$$

Plus la longueur de définition d'un schéma est longue, plus il a de chance d'être cassé par un croisement.

Effet de la mutation

De même, l'effet de la mutation est lié à ce que l'on définit comme l'ordre du schéma $o(S)$, c'est-à-dire le nombre de bits fixés du schéma. Pour qu'un schéma survive à l'opérateur de mutation, il faut que tous les gènes fixés demeurent inaltérés [Zak 07]. Si $1-p_m$ est la probabilité de survie d'un gène, la probabilité qu'un représentant d'un schéma d'ordre $o(S)$ survive est :

$$P_s(S) = (1 - P_m)^{o(S)} \dots\dots\dots (A.7)$$

Puisque, une approximation à l'expression (A.7) peut être donnée par

$$P_s(S) = 1 - o(S)P_m \dots\dots\dots (A.8)$$

Théorème fondamental des AGs

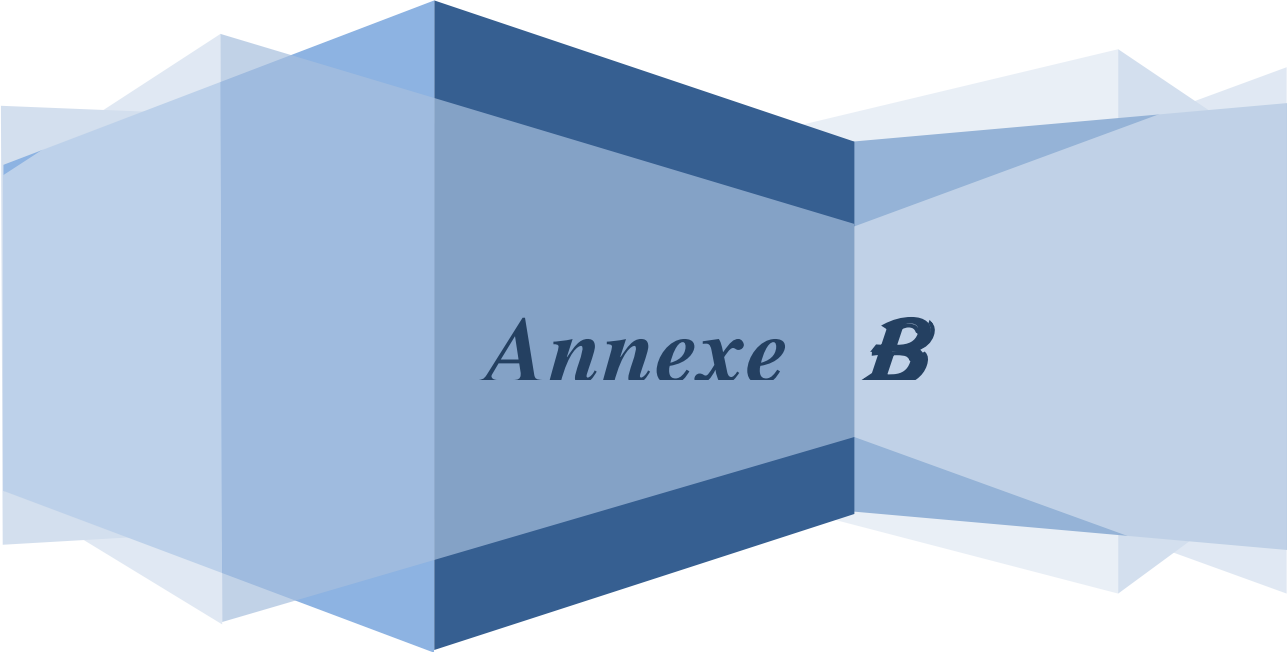
Si on considère les opérations de la sélection, du croisement, et la mutation comme des événements indépendants, alors le nombre de chaînes représentant un schéma S dans la prochaine génération peut être exprimé par

$$m(S, t + 1) \geq m(S, t) \frac{f(S)}{f_{moy}} \left[1 - P_c \frac{\delta(S)}{L - 1} \right] \left[1 - o(S)P_m \right] \quad \dots(A.9)$$

Le terme $P_c P_m o(S) \delta(S) / (L - 1)$ est tellement petit par rapport aux autres termes de l'expression (B.9) qu'il est négligé et cette dernière devient alors

$$m(S, t + 1) \geq m(S, t) \frac{f(S)}{f_{moy}} \left[1 - o(S)P_m - P_c \frac{\delta(S)}{L - 1} \right] \quad \dots\dots\dots(A.10)$$

Donc, selon l'équation (A.9) ou (A.10), on peut conclure que les schémas courts, de petits ordres, et d'aptitudes au-dessus de la moyenne auront un nombre exponentiellement croissant de représentants dans les générations ultérieures, cette conclusion est le théorème fondamental des AGs ou le théorème des schémas. Il a une importance spéciale dans les applications des AGs.



Annexe **B**

ANNEXE B

L'origine du mot « robot » :

Le mot robot a été utilisé pour la première fois dans la pièce R.U.R (Rossum's Universal Robots) de Karel Capek. Le mot robot a été inventé par Josef, son frère de Capek. Il est dérivé du pronom tchèque "robota" - "le travail"-, alors le robot est "quelqu'un qui exerce du travail". Ce mot est devenu connu dans le monde entier et il est aujourd'hui utilisé très fréquemment [Zun 98].

Pourquoi les robots ?

L'avantage d'un robot sur l'homme est sa régularité : il peut exécuter le même mouvement des milliers de fois d'affilée sans ressentir la moindre lassitude ou fatigue. De plus, les robots peuvent être construits de manière à pouvoir résister à des conditions qui seraient dangereuses, voire mortelles pour l'homme (gaz nocifs, fortes chaleurs, froid intense, radiations, ...). Par exemple : l'utilisation de robots pour peindre les châssis de voiture avec des revêtements contenant du plomb.

Définition d'un robot :

C'est une machine pouvant manipuler des objets et réalisant des mouvements variés dictés par un programme aisément modifiable.

Programmer un robot consiste, dans un premier temps, à lui spécifier la séquence des mouvements qu'il devra réaliser. Certains robots sont dotés de "sens" ; c'est-à-dire d'un ensemble plus ou moins important d'instruments de mesure et d'appréciation (caméra, thermomètre, télémètre, ...) permettant au programme du robot de décider du mouvement le mieux adapté aux conditions extérieures.

Par exemple: si un robot mobile muni d'une caméra est amené à se déplacer dans un local inconnu, on peut le programmer pour qu'il contourne tout obstacle qui entraverait sa route.

On essaie également de doter des robots d'un dispositif d'intelligence artificielle afin qu'ils puissent faire face à des situations imprévues et nouvelles (le robot pourrait acquérir une certaine "expérience").

En général, on peut définir un robot mobile comme étant une machine équipée de capacité de perception, de décision et d'action qui lui permettent d'agir de manière autonome dans son environnement en fonction de la perception qu'il en a.

Le type de locomotion définit deux type de constraints :

- Les contraintes cinématiques, qui portent sur la géométrie des déplacements possibles du robot.
- Les contraintes dynamiques, liées aux effets du mouvement (accélérations bornées, vitesses bornées, présence de forces d'inertie ou de friction). **[BKH 06]**

Les robots mobiles à roues :

Les robots mobiles à roues sont les plus répons actuellement pour des raisons de simplicité de conception et de commande. Les mécanismes sont conçus avec des roues ayant plusieurs fonctions : roues motrices, roues directrices, roues motrices-directrices, roues libres et roues folles. Les roues sont le moyen de locomotion le plus répandu en matière de robotique de mobile.

Il existe plusieurs types de robots à roues :

- Robot mobiles à une roue.
- Robot mobiles à deux roues.
- Robot mobiles à trois roues.
- Robot mobiles à quatre roues.
- Robot mobiles à six roues.

Robots mobiles tricycles :

Un grand nombre de conception de formes géométriques des robots existe soit au niveau de l'industrie ou de recherche. La forme est habituellement soit circulaire, soit rectangulaire. Trois roues équipent généralement les plates formes mais le nombre des roues motrices, le nombre des roues directrices ainsi que l'emplacement de celles-ci sur le robot différent complètement d'un robot à un autre.

Selon sa cinématique, un robot est dit :

-Holonome : s'il se déplacer instantanément dans toutes les directions.

-Non-holonome : si ses déplacements autorisés sont des courbes dont la courbure est bornée

Etude cinématique du robot :

Le modèle cinématique du robot mobile à roues considéré dans notre projet est du type tricycle, avec deux roues arrières motrices indépendantes et une roue folle à l'avant (figure 1). Le mouvement d'un robot mobile à roues sur un plan peut être décomposé en un mouvement de translation et un mouvement de rotation par rapport à un point de référence. Le point considéré dans notre cas est le point A, le centre de rotation naturel (CR) du véhicule lorsque les vitesses des roues motrices sont d'amplitudes égales et opposées. Ce point est généralement différent du centre instantané de rotation (CIR) qui se trouve à l'intersection de l'axe des roues motrices avec l'axe de la roue folle.

Soient ω_g et ω_d les vitesses de rotation des roues gauche et droite respectivement.

La vitesse linéaire des roues est donc :

$$\begin{aligned} V_g &= r \cdot \omega_g \\ V_d &= r \cdot \omega_d \end{aligned} \dots\dots\dots(B-1)$$

Le mouvement d'un solide vérifie le théorème suivant :

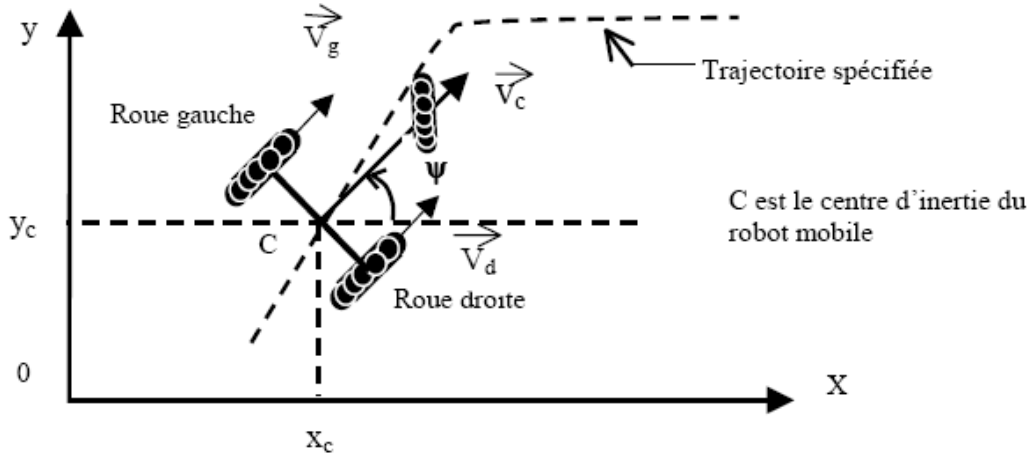


Figure 1 : Robot mobile considéré

A chaque instant 't' il existe un vecteur Ω tel que pour tout point P du solide on a :

$$\vec{V}_P = \vec{V}_A + \vec{\Omega} \wedge \vec{AP} \dots\dots\dots(B-2)$$

Où A est un point quelconque du solide (robot mobile). Les vitesses des roues motrices V_g et V_d sont mesurées par odométrie. Ainsi la vitesse V_c du centre de rotation est donnée par :

$$\vec{V}_C = \vec{V}_g + \vec{\Omega} \wedge A\vec{I}_g \quad \dots\dots\dots (B-3)$$

$$\vec{V}_C = \vec{V}_d + \vec{\Omega} \wedge A\vec{I}_d \quad \dots\dots\dots (B-4)$$

En sommant les équations (4) et (5) on obtient :

$$2.\vec{V}_C = \vec{V}_d + \vec{V}_g + \vec{\Omega} \wedge (A\vec{I}_D + A\vec{I}_G) \quad \dots\dots\dots (B-5)$$

Les vecteurs $A\vec{I}_D$ et $A\vec{I}_G$ étant opposés, on déduit donc :

$$\vec{V}_C = \frac{\vec{V}_g + \vec{V}_d}{2} \quad \dots\dots\dots (B-6)$$

Du point de vue directionnel, le vecteur V_c a la même direction que les vecteurs V_g et V_d . De même, en soustrayant (A-3) de (A- 4), on obtient :

$$\vec{V}_d - \vec{V}_g + \vec{\Omega} \wedge (A\vec{I}_D - A\vec{I}_G) = \vec{V}_d - \vec{V}_g + \theta.\vec{k}' \wedge (-2h\vec{j}') = 0 \quad \dots\dots\dots (B-7)$$

L'expression de Ω est donc :

$$\vec{\omega} = \theta.\vec{k}' = \frac{V_d - V_g}{2.h}.\vec{k}' \quad \dots\dots\dots (B-8)$$

On constate que le vecteur ω est de direction constante et orthogonale au plan du mouvement. Le théorème précédent (équation 3) peut être utilisé pour déterminer le centre instantané de rotation (CIR) du robot mobile. Celui-ci se trouve dans le prolongement de l'axe commun des roues motrices et il est caractérisé par une vitesse V_p nulle (figure 2).

$$\vec{V} + \vec{\Omega} \wedge A.CIR = \vec{V} + \dot{\theta} \vec{k} \wedge \rho . \vec{j}' = 0 \dots\dots\dots(B-9)$$

D'où : $\rho = \frac{V}{\dot{\theta}}$ (B- 10)

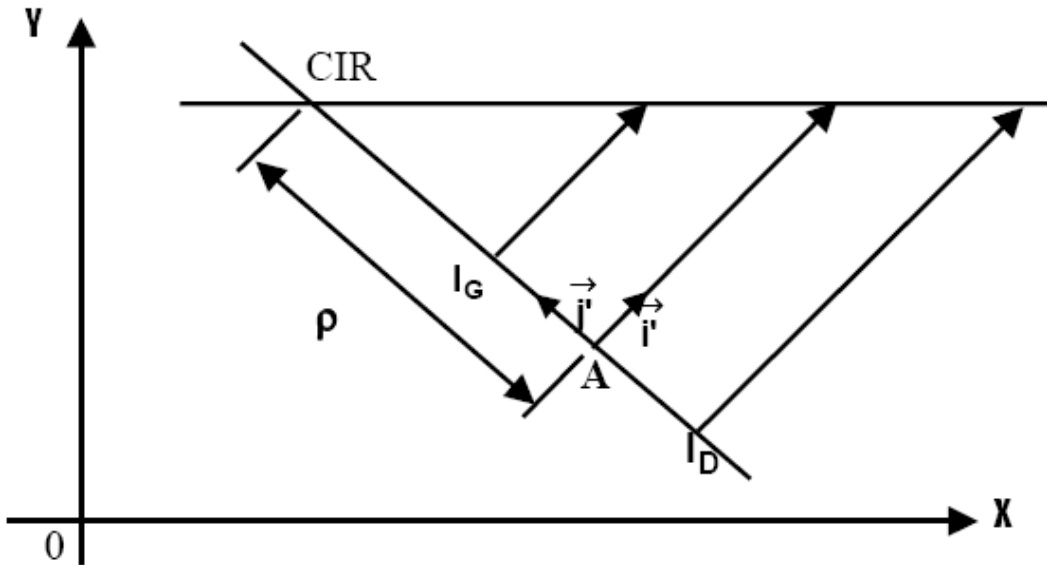


Figure B-2 : détermination du CIR

A partir de l'équation (A-5) on peut déduire les trois équations définissant les cinématiques du robot dans un repère inertiel :

$$\begin{aligned} \dot{x} &= \frac{Vd + Vg}{2} \cos \theta \\ \dot{y} &= \frac{Vd + Vg}{2} \sin \theta \dots\dots\dots(B-11) \\ \dot{\theta} &= \frac{Vd - Vg}{2L} \end{aligned}$$

Equation de la non-holonomie :

Pour ce type du mobile ,on suppose que la base mobile roule sans glissement, alors la vitesse du centre de masse C est perpendiculaire à l'axe des roues. Ses composantes sont données par les relations suivantes :

$$\dot{x} = Vc \cos \theta \quad \dots\dots\dots(B-12)$$

$$\dot{y} = Vc \sin \theta \quad \dots\dots\dots(B-13)$$

On peut exprimé en fonction des coordonnées (x_A, y_A) d'un point A situé à l'avant de la base mobile on écrit :

$$x_A = x + d \cos \theta \quad \dots\dots\dots(B-14)$$

$$y_A = y + d \sin \theta$$

Localisation des robots mobiles :

La localisation et la navigation ont toujours suscité un intérêt particulier dans le domaine de la robotique mobile. Un robot mobile doit, par définition, évoluer dans un environnement, celui-ci peut être assez complexe et dans certain cas hostile.

Une localisation s'avère ainsi indispensable, car certain choix et décisions ne peuvent être faits sans la connaissance de celle-ci. [Aba 98]

Un robot mobile ne peut atteindre un but défini par ses coordonnées que s'il connaît sa propre position et son orientation par rapport à l'environnement. Pour être autonome, un robot mobile doit être capable de se localiser mais également pouvoir élaborer des stratégies de déplacement. C'est le rôle de la navigation. En effet, durant l'exécution d'une tâche, le robot est amené soit à poursuivre une trajectoire prédéfinie soit à éviter les obstacles imprévus sur son chemin soit suivre un mure ...[Ben 00]

Donc on peut dire que la localisation est une fonction importante d'un robot mobile.

Elle permet de connaître à chaque instant les coordonnées (x, y, θ) représentant la position et l'orientation du robot par rapport à un référentiel lié à l'environnement. Ces trois coordonnées sont nécessaires pour maintenir le robot sur sa trajectoire et le diriger vers sa destination.

L'odométrie :

Est la méthode de localisation la plus couramment employée pour les robots disposant d'une structure de locomotion à roues. Son principe consiste à déduire une position, de façon incrémentale, à partir de la vitesse et de la géométrie des roues. La mise en œuvre de cette méthode est plus simple et ne nécessite qu'une puissance de calcul très limitée.

[MAL 01]

L'odométrie permet d'estimer le déplacement à partir de la mesure de rotation des roues (ou du déplacement des pattes). La mesure de rotation est en général effectuée par un codeur optique disposé sur l'axe de la roue, ou sur le système de transmission (par exemple sur la sortie de la boîte de vitesse). Le problème majeur de cette mesure est que l'estimation du déplacement fournie dépend très fortement de la qualité du contact entre la roue et le sol. Elle peut être relativement correcte pour une plate-forme à deux roues motrices sur un sol plan de qualité uniforme, mais est en général quasiment inutilisable seule pour un robot à chenille par exemple. Notons cependant que l'erreur de ces méthodes se retrouve en général principalement sur l'estimation de la direction du robot, tandis que la mesure de la distance parcourue est souvent de meilleure qualité.

Le principe de calcul de la position et de l'orientation du robot dans cette méthode, par rapport à un repère absolu. Elle suppose de petits déplacements du robot entre deux instants successifs t_n et t_{n+1} .

Les équations calculées par l'odométrie sont :

$$\Delta E \theta_n = \frac{\Delta q_d - \Delta q_g}{L}$$

$$\Delta D = \frac{\Delta q_d + \Delta q_g}{2}$$

$$\theta_{n+1} = \theta_n + \Delta \theta_n$$

$$x_{n+1} = x_n + \Delta D \cos(\theta_n + \Delta \theta_n)$$

$$y_{n+1} = y_n + \Delta D \sin(\theta_n + \Delta \theta_n)$$

BIBLIOGRAPHIE

- [Aba 98] : F.Ababsa, « commande d'un robot mobile par IB-PC » thèse de magister institut d'électronique , université de BATNA, 1998.
- [BAL 99] : Lotfi BAGHLI, « Contribution à la commande de la machine asynchrone, utilisation de la logique floue, des réseaux de neurones et des algorithmes génétiques », Thèse de doctorat , université Henri Poincaré, Nancy-I ; 14 janvier 1999.
- [Ben 00] : A.Bendifallah « localisation par balises d'un robot mobile », thèse De Magister, institut d'électronique université de Batna 2000.
- [BMS 01] : BenMansour Nabil, « Analyse et synthèse d'un contrôleur flou de type PID ».Mémoire de fin d'études ; spécialité : électronique, option : contrôle ;Université de BATNA ; septembre 2001.
- [DRD 05] : Mme DRDI Leila, « Les algorithmes génétiques »,Le présent chapitre est extrait de la thèse de doctorat ; INRS-ETE, 2005.
- [Glo 99] : Glorennec Pierre Yves, « algorithmes d'apprentissages pour système d'inférence floue », Département d'informatique INSA de Rennes / IRISA ;1999
- [GuF 94] : Guely François , « Apprentissage de bases de règles floues: contributio à une étude systématique de l'approche de l'optimisation ». Thèse de doctorat ; Ecole centrale des arts et manufactures, Chatenay-Malabry, France ; 1994.

- [Ham 07] : Hamdouche Sonia, « Contribution a l'amélioration des structures de régulation des courants des machines à courant alternatif ». Thèse de Magister en Electronique de Puissance ; Université de BATNA 2007.
- [Kou 06]: Hanane Kouara, « Application d'un filtre actif série au contrôle de la Tension d'un réseau basse tension». Thèse de Magister en Electronique de Puissance ; Université de BATNA 2006.
- [Lab 07] : Lavoie benoit, « logique classique et logique floue », Programme de Doctorat en Informatique Cognitive Université du Québec à Montréal, 8 février 2007.
- [LUT 96] : Lutaud-Brunet, « Identification et contrôle de processus par réseaux neuro-flous » . Thèse de doctorat ; Université de d'Evry Val d'Essonne ,1996.
- [MAL 01]: Mallet Anthony, « Localisation d'un robot mobile autonome en Environnements naturels Méthodes et intégration » . Thèse préparée au Laboratoire d'Analyse et d'Architecture des Systèmes en vue de l'obtention du titre de Docteur de l'Institut National Polytechnique de Toulouse Systèmes Automatiques ; 2 juillet 2001.
- [MAT 00] : Mathieu Renversade, « Optimisation d'un dispositif hyper-sustentateur par Algorithmes Génétiques » ;N° 4029 Octobre 2000.

- [SOP 04]: Sophie Voisin, « Application des Algorithmes Génétiques à l'estimation de mouvement par modélisation markovienne » *Rapport de stage*, Stage DEA - Mars - Juin 2004.
- [SoR 04] : Souquet Amédée ,Radet Francois-Gérard, « Algorithmes Génétiques ». TE de fin d'année ; 21 juin 2004.
- [WeW98] : Wei WU, « Synthèse d'un contrôleur flou par Algorithme Génétique : Application au réglage dynamique des paramètres d'un système ». Thèse pour obtenir le grade de Docteur de l'Université de Lille 1 ; 18 Décembre 1998.