

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITE MOHAMED SEDDIK BENYAHIA JIJEL

Faculté des sciences et de la technologie

Département d'Electronique

N° :/2023

MEMOIRE DE MASTER

DOMAINE : Sciences et Technologies

FILIERE: Electronique

Spécialité : Electroniques Des Systèmes Embarqués

Mémoire de master

Thème

**Application de l'algorithme de chauve-souris
pour le contrôle intelligent d'une lévitation
magnétique**

Présenté Par : Yousra Rida

Encadré par : Dr. N. Talbi

Chaima Kouihal

Date de soutenance : 26/06/2023

Jury de Soutenance

Président : Pr. A.Boukabou

Pr

Univ MSB Jijel

Encadrant : Dr. N. Talbi

MCB

Univ MSB Jijel

Examineur : Dr. S. Bouatmane

MCB

Univ MSB Jijel

Promotion : 2022 /2023

Remerciement

Nous remercions tout d'abord le tout puissant qui par sa grâce nous permettons d'arriver au bout de nos efforts en nous donnant la santé, la force et le courage.

Nous tenons à exprimer nos profondes reconnaissances à notre directrice de thèse madame **Talbi Nesrine**, qui grâce à sa disponibilité, son soutien, ses conseils et ses encouragements nous nous permettons de mener à bien ce travail.

Nos remerciements vont également à nos enseignants **Bouatmane Sabrina** et **Boukabou abdelkrim** d'avoir accepté de juger notre travail.

Nous remercions nos très chers parents qui ont toujours été là pour nous.

Enfin, nous remercions ainsi l'ensemble des collègues et amis de leurs support et soutien durant nos études.

Dédicace

Je dédie ce travail à :

A celui qui s'est changé la nuit en jour pour m'assurer les bonnes

Conditions à mon cher père Mouloud.

A celle qui a attendu avec patience les fruits de sa bonne éducation et de

Ses dévouements à ma chère mère Houria.

A mes sœurs Chahinez, Ferial et mon frère Mouhcine qui m'ont toujours

Soutenues.

A mon binôme Chaima et sa famille.

Dédicace

Je dédie ce travail à tous ceux qui m'ont aidé à réaliser ce travail.

A ma mère Walida

J'espère qu'elle le trouve récompensant de tous les sacrifices qu'elles consentirent pour moi.

A mon père AzeDinne

En témoignage de ma sincère gratitude pour ses sacrifices qu'il a consentis pour moi et pour toute la famille

A mes frères : Anis, Wail, Bassem et Nafaa.

A ma sœur : Rahil.

A mon binôme Yousra et sa famille.

A tous mes amis en témoignage de mon amitié.

ملخص

تقدم هذه المذكرة نهجًا جديدًا لتصميم قاعدة لمتحكم غامض باستخدام خوارزمية تحسين الخفافيش. تعد خوارزمية الخفافيش طريقة تحسينية واعدة تم اقتراحها مؤخرًا لحل مشاكل التحسين يعتمد على محاكاة سلوك تحديد الموقع بالصدى للخفافيش وتهدف إلى تحسين قاعدة المتحكم الغامض للأنظمة الديناميكية والمعقدة واللاخطية بشكل كبير مع تلبية معايير الأداء المطلوبة. لتوضيح فعالية النهج المقترح، تم اختيار نظام تعليق كرة مغناطيسية كمثال للتحكم. أظهرت نتائج المحاكاة أن النهج المقترح يمكن استخدامه كطريقة بسيطة وفعالة للحصول على تحديد أمثل للإعدادات قاعدة غامضة

Abstract

This work presents a new approach for designing a fuzzy controller using the Bat optimization algorithm. The Bat algorithm is a promising meta-heuristic recently proposed for solving optimization problems. It is based on simulating the echolocation behavior of bats and aims to optimize the fuzzy rule base parameters of the fuzzy controller for dynamic, complex, and highly nonlinear systems while meeting desired performance criteria. To illustrate the effectiveness of the suggested approach, a magnetic ball suspension system was chosen as a control example. Simulation results showed that the proposed approach can be used as a simple and efficient optimization method for obtaining optimal determination of fuzzy rule base parameters.

Résumé

Ce mémoire présente une nouvelle approche pour la conception d'une base de règles floues pour un contrôleur flou en utilisant l'algorithme d'optimisation de chauve-souris. L'algorithme de chauve-souris est une méta-heuristique récemment proposée pour la résolution de problèmes d'optimisation. Il se base sur la simulation du comportement d'écholocation des chauves-souris et vise à optimiser la base de règles floues du contrôleur flou pour les systèmes dynamiques, complexes et hautement non linéaires, tout en respectant les performances souhaitées. Pour illustrer l'efficacité de l'approche suggérée, un système de lévitation magnétique a été choisi comme exemple de contrôle. Les résultats de la simulation ont montré que l'approche proposée peut être utilisée comme une méthode d'optimisation simple et efficace pour obtenir une détermination optimale des paramètres de base de règles floues.

TABLES DES MATIERES

Introduction Générale.....	1
Chapitre I : La Commande Floue	
I.1 Introduction.....	4
I.2 Théorie de la logique floue	5
I.2.1 Définition.....	5
I.2.2 Historiques	5
I.2.3 Univers du discours	6
I.2.4 Variables linguistiques.....	6
I.2.5 les règles linguistiques	7
I.2.6 Caractéristiques des ensembles flous	7
I.2.6.1 Type	7
I.2.6.2 Support	7
I.2.6.3 Hauteur	8
I.2.6.4 Noyau	8
I.2.7 Fonction d'appartenance.....	8
I.2.7.1 Fonction triangulaire	9
I.2.7.2 Fonction trapézoïdale.....	9
I.2.7.3 Fonction gaussienne	10
I.2.7.4 Fonction sigmoïde.....	10
I.2.8 les opérations sur les ensembles flous	10
I.3 La commande floue	11
I.3.1 Le principe de la commande floue	11
I.3.1.1 Interface de fuzzification	12
I.3.1.2 Base de connaissance	12
I.3.1.3 Moteur d'inférence floue	13
I.3.1.4 Interface de défuzzification	14
I.3.2 les différents types de contrôleurs flous	16
I.3.2.1 Contrôleurs flous de type Mamdani	16
I.3.2.2 Contrôleurs flous de type Takagi-Sugeno	17
I.3.3 avantages et inconvénients de la logique floue.....	17
I.3.4 Domaines d'applications.....	18
I.4 Conclusion	19

Chapitre II : L'Optimisation par l'Algorithme de chauve-souris

II.1 Introduction.....	20
II.2 Méthodes d'optimisation	21
II.2.1 Objectif d'optimisation	21
II.2.2 les éléments d'optimisation.....	21
II.2.3 Classification.....	22
II.2.3.1 Méthodes exactes	22
II.2.3.2 Méthodes approchées	23
II.3 Les Méta-heuristiques	24
II.3.1. Définition	24
II.3.2. Différents types de méta-heuristiques.....	24
II.3.3.1 Les méta-heuristiques à population	24
II.3.3.2 Les méta-heuristiques à parcours	25
II.4 L'Algorithme de chauve-souris.....	25
II.4.1 Description.....	25
II.4.2 Echolocation des chauves-souris	25
II.4.3 L'algorithme standard de chauve-souris.....	26
II.4.4 Génération de la population initiale	28
II.4.5 Fonction objective.....	29
II.4.6 Critère d'arrêt.....	29
II.5 Domaine d'application de l'Algorithme de chauve-souris	29
II.6 Conclusion	30

Chapitre III : Simulation & Interprétations

III.1 Introduction	31
III.2 La commande floue du système de lévitation magnétique	32
III.2.1 Introduction	32
III.2.2 Le modèle mathématique de lévitation magnétique.....	32
III.2.3 Structure du contrôleur flou	34
III.2.4 Résultats de la simulation.....	35
III.2.4.1 Régulateur à 3 classes	35
III.2.4.2 Régulateur à 5 classes	37
III.2.4.3 Régulateurs à 7 classes.....	40
III.3 Optimisation de contrôleur flou à 3 classes par l'algorithme de chauve-souris	42

III.3.1 Stratégie d'optimisation du contrôleur flou.....	43
III.3.2 Application à la commande de la position de la balle magnétique	44
III.3.2.1 Résultats de la simulation	45
III.3.2.2 Interprétation des résultats	48
III.3.2.3 Test de robustesse.....	48
III.3.3 Comparaison de l'algorithme de chauve-souris et différentes méthodes d'optimisation	50
III.4 Conclusion	52
Conclusion Générale	53
Bibliographie	55

Liste des figures

Figure I.1 : Classification des ensembles flous	5
Figure I.2 : Variables linguistiques.....	6
Figure I.3 : Caractéristiques d'un sous ensemble flou A	8
Figure I.4 : fonction caractéristique.....	9
Figure I.5 : fonction d'appartenance.....	9
Figure I.6 : Formes usuelles des fonctions d'appartenance	9
Figure I.7 : Structure de base d'un contrôleur flou.....	11
Figure II.1 :Elements essentiels pour l'optimisation.....	21
Figure II.2 : Classification des méthodes d'optimisation.....	23
Figure II.3 : Principe de l'écholocation.....	26
Figure II.4 : Pseudo code de l'algorithme des chauves-souris	28
Figure III.1 : structure du procédé.....	32
Figure III.2 : structure du contrôleur position de la balle magnétique	34
Figure III.3 : formes des fonctions d'appartenance (3x3).....	35
Figure III.4 : surface caractéristique du régulateur flou 3 classes.....	36
Figure III.5 : Evolution de la position et la référence en fonction du temps.....	36
Figure III.6 : Evolution du signal de commande en fonction de temps.	37
Figure III.7 : Evolution de l'erreur et sa variation.	37
Figure III.8 : formes des fonctions d'appartenance (5x5).....	38
Figure III.9 : surface caractéristique du régulateur flou 5 classes.....	38
Figure III.10 : Evolution de la position et la référence en fonction du temps.....	39
Figure III.11 : Evolution du signal de commande en fonction de temps	39
Figure III.12 : Evolution de l'erreur et dérivé d'erreur.....	39
Figure III.13 : formes des fonctions d'appartenance (7x7).....	40
Figure III.14 : surface caractéristique du régulateur flou 7 classes	41
Figure III.15 : Evolution de la position et la référence en fonction du temps	41
Figure III.16 : Evolution du signal de commande en fonction de temps	41
Figure III.17 : Evolution de l'erreur et sa variation	42
Figure III.18 : Forme de fonction d'appartenance à optimiser.....	43
Figure III.19 : Structure d'optimisation des contrôleurs flous	44
Figure III.20 : L'évolution de la fonction d'évaluation	46
Figure III.21 : Disposition et formes des fonctions d'appartenance des prémisses et des conclusions du CF après optimisation	47
Figure III.22 : Evolution de la position et la commande optimisée.....	47
Figure III.23 : Evolution de l'erreur et sa variation après l'optimisation	48
Figure III.24 : Réponses aux changements des conditions initiales	49
Figure III.25 : Réponses aux changements de la masse de la balle m	49
Figure III.26 : Réponses aux changements de la résistance d'enroulement R.....	49
Figure III.27 : Réponses aux changements de la bobine d'enroulement L.....	50
Figure III.28 : La comparaison entre le BAT et des différentes méthodes de l'optimisation.	51

List des tableaux

Tableau III.1 : Paramètres du système.	33
Tableau III.2 : Table des règles floues du contrôleur flou (3x3).	36
Tableau III.3 : Table des règles floues du contrôleur flou (5x5).	38
Tableau III.4 : Table des règles floues du contrôleur flou (7x7).	40
Tableau III.5 : paramètres de l'algorithme de chauve-souris.	45
Tableau III.6 : Paramètres optimaux des centres de fonctions d'appartenance de $e(t)$	46
Tableau III.7 : Paramètres optimaux des centres de fonctions d'appartenance de $\Delta e(t)$	46
Tableau III.8 : Paramètres optimales de la base de règles floues.	46
Tableau III.9 : La comparaison entre l'algorithme de BAT et RS, RT.	51

Introduction générale

L'industrie, de nos jours fait face aux problèmes de contrôle des systèmes non linéaires de plus en plus complexes qui présente des modèles mathématiques inconnus ou difficiles à exploiter. Cependant des développements importants sur le plan pratique et théorique ont été effectuées dans le domaine de la commande des systèmes industriels .Plusieurs travaux furent concrétisés, pour sa part la recherche dans la dynamique et le contrôle des procédés non linéaires a connu également un développement remarquable [1].

Au cours des dernières années, des méthodes intelligentes telles que les réseaux de neurones et la logique floue et les méta-heuristiques ont connu un essor remarquable dans différents domaines [2]. La logique floue est une méthode intelligente de contrôle qui permet de tenir compte des variables imprécises ou incertaines dans les systèmes de contrôle. Elle repose sur la théorie des ensembles flous, qui permet de définir des ensembles de valeurs possibles pour une variable donnée avec une probabilité d'appartenance déterminée [3].

La conception d'un contrôleur flou est une tâche complexe, Surtout d'une part, le choix optimal des paramètres du contrôleur flou (les fonctions d'appartenance, les règles floues, les gains, etc....) représentent une tache très difficile, peut avoir un impact significatif sur la performance globale du système. C'est pourquoi les chercheurs ont développé plusieurs méthodes optimales et systématiques pour la conception des contrôleurs flous. Parmi ces méthodes, les méta-heuristiques sont particulièrement utiles car elles permettent d'explorer efficacement l'espace de recherche des paramètres du contrôleur flou. Ces techniques

d'optimisation sont basées sur des algorithmes évolutifs et des stratégies de recherche locales pour trouver les meilleures solutions possibles [4].

Les algorithmes méta-heuristiques deviennent maintenant des méthodes puissantes pour résoudre de nombreux problèmes d'optimisation difficiles. La grande majorité des algorithmes méta-heuristiques ont été dérivés du comportement des systèmes biologiques et/ou des systèmes physiques dans la nature [5]. Parmi ces méthodes, l'algorithme de chauve-souris, qui est la méthode utilisée dans ce projet, a été développé sur la base du comportement des chauves-souris lorsqu'elles cherchent de la nourriture dans l'obscurité. Les chauves-souris émettent des ultrasons pour localiser leur proie et ajustent leur comportement en fonction de la distance et de la qualité de la proie détectée [6].

Dans ce travail, nous nous concentrons sur l'application de l'algorithme de chauve-souris pour le contrôle intelligent d'une lévitation magnétique. Ce sujet a été choisi en raison de l'importance croissante des systèmes de lévitation magnétique dans divers domaines, tels que les transports, la médecine et l'industrie. L'objectif principal de ce mémoire est de l'optimisation de contrôleur flou de système lévitation magnétique par l'algorithme de chauves-souris.

Pour ce faire ce mémoire est organisé comme suit :

Le premier chapitre de ce mémoire se concentre sur la logique floue. Nous y décrivons les principes de base de cette méthode, en présentant notamment la théorie des éléments de base de la logique floue et les différentes opérations nécessaires à la commande floue. Nous expliquons également la structure d'un contrôleur flou, en détaillant les étapes de fuzzification, d'inférence et de défuzzification, ainsi que les caractéristiques des deux types de contrôleurs flous les plus utilisés (Mamdani et Takagi-Sugeno). Enfin, nous exposons les avantages et les inconvénients et les domaines d'applications de la logique floue.

Dans le deuxième chapitre, nous abordons l'optimisation et les différentes méthodes de résolution d'un problème d'optimisation. Nous présentons ensuite l'origine des méta-heuristiques, leur classification et leur formalisation. Nous expliquons en détail l'algorithme de chauve-souris, ses différentes caractéristiques et les domaines d'application dans lesquels il peut être utilisé. Enfin, nous exposons les avantages et les inconvénients et les domaines d'applications de l'algorithme de chauve-souris.

Dans le dernier chapitre, nous avons mis en œuvre la logique floue et l'optimisation pour la commande du système de lévitation magnétique. Dans un premier temps, nous avons

développé un contrôleur flou de type Mamdani pour réguler la position de la balle dans le système. Ensuite, nous avons optimisé les paramètres du contrôleur flou de type Mamdani à l'aide de l'algorithme de chauve-souris. Nous avons particulièrement optimisé les centres des fonctions d'appartenance des entrées du contrôleur flou (valeurs modales) ainsi que les conclusions des règles (les sorties). L'objectif était de minimiser un critère d'erreur entre la consigne et la sortie du système. Ce chapitre présente les résultats obtenus pour le système de lévitation magnétique, le test de robustesse du contrôleur flou obtenu ainsi qu'une comparaison entre l'algorithme de chauve-souris et d'autres différentes méthodes d'optimisation pour la commande du même système.

Le mémoire se termine par une conclusion générale sur le travail réalisé.

Chapitre I : La Commande Floue

I.1 Introduction:

Beaucoup de chercheurs, dans différents domaines scientifiques, utilisent la théorie de la logique floue formulée par le professeur Lotfi Zadeh ; cette théorie a connu un intérêt important dans la communauté scientifique pendant plusieurs années. L'une des raisons principales est l'énorme succès des équipements domestiques produits par l'industrie Japonaise, utilisant des régulateurs flous : ce marché atteignait 2 milliards de dollars en 1990. Cette théorie est très attractive, parce qu'elle est basée sur le raisonnement intuitif et prend en compte la subjectivité et l'imprécision mais ce n'est pas une théorie imprécise. C'est une théorie mathématique rigoureuse, adaptée au traitement de tout ce qui est subjectif et/ou incertain [7].

Les systèmes flous appartiennent à la classe des "systèmes à base de connaissance" ; leur but principal consiste à implémenter un savoir-faire humain, ou des règles linguistiques, sous forme d'un programme informatique.

Les algorithmes basés sur la logique floue sont considérés comme une solution très intéressante pour le réglage des systèmes non linéaires pour lesquels il n'existe pas de modèle mathématique [8].

L'objectif de ce chapitre est de présenter les notions générales de la logique floue, et le principe de conception de la commande floue. D'abord, quelques concepts de base de la logique floue ainsi que leurs applications pour la synthèse d'une loi de commande floue seront élaborés. Après, les éléments de bases d'un régulateur flou, les différents types de régulateurs flous, ainsi que leurs avantages et leurs inconvénients seront présentés.

I.2 La théorie de la logique floue

I.2.1 Définition

On peut dire que la logique floue est une extension de la logique booléenne. Dans la théorie des ensembles classiques, l'appartenance d'un élément à un sous-ensemble est définie par une valeur logique standard : 1 si l'élément appartient au sous-ensembles, 0 si non. Dans la théorie de la logique floue, un élément peut-appartenir en partie à un sous-ensemble : son degré d'appartenance est décrit par une valeur comprise entre 0 et 1. Cette théorie permet la modélisation des imperfections des données et se rapproche dans une certaine mesure de la flexibilité du raisonnement humain [9].

Un exemple simple d'ensembles flous est la classification des personnes selon leur âge en trois ensembles : jeune, moyen et vieux. La façon d'établir cette classification est présentée à la Fig.I.1

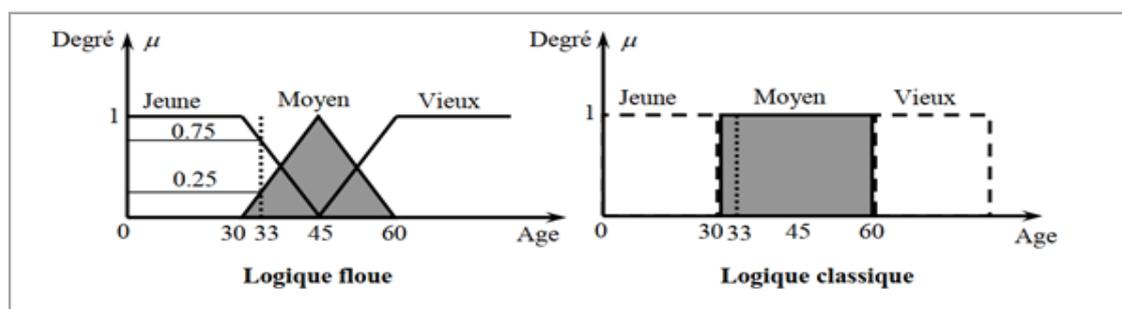


Fig.I.1 : Classification des ensembles flous [11].

I.2.2 Historique [8] :

Les prémisses de la logique floue sont apparues en 1940 lorsque des chercheurs américains ont commencé à se pencher sur le concept d'incertitude.

En 1965, Lofti Zadeh introduit la théorie des ensembles flous, (fuzzy set theory) [12]. En 1973 le professeur Zadeh publie un article qui mentionne pour la première fois le terme de valeurs linguistiques dont la valeur est un mot du langage naturel.

En 1974, Mamdani établit la première application industrielle de la logique floue : la régulation floue d'une chaudière à vapeur. Il introduit ainsi la commande floue dans la régulation industrielle.

Grâce au chercheur japonais M. Sugeno, la logique floue a été introduite au Japon en 1985. Les sociétés japonaises comprennent l'avantage technique et commercial de cette logique à savoir :

- facilité d'implantation
- solution des problèmes multi-variables complexes
- robustesse vis à vis des incertitudes

I.2.3 Univers du discours [11]

L'univers de discours est l'ensemble de référentiel qui contient tous les éléments qui sont en relation avec le contexte de données, il est considéré comme le domaine de fonctionnement du processus car il couvre l'ensemble des valeurs prises par cette variable. C'est l'une des étapes principales pour la conception d'une application floue.

I.2.4 Variables linguistiques [8]

Les variables linguistiques sont des variables dont les valeurs ne sont pas des nombres, mais des mots du langage naturel. La variable linguistique est caractérisée par un triplé $(v ; T(v) ; V)$ avec :

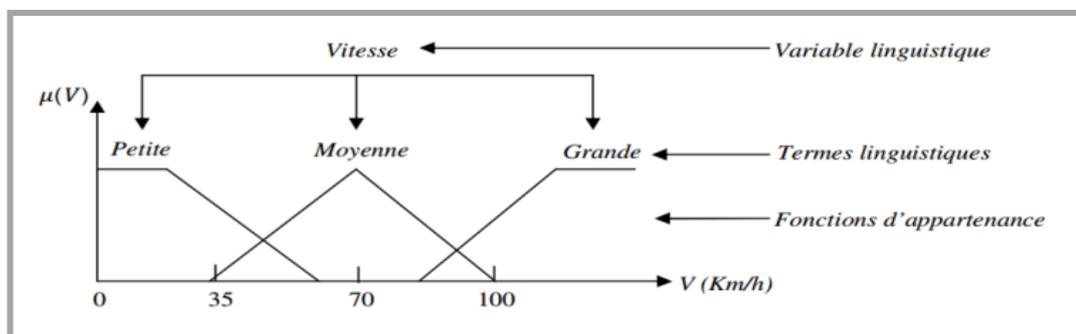


Fig.I.2: Variables linguistiques [12].

- v est le nom de la variable linguistique (vitesse)
- V est l'univers de discours $V = [0, 120\text{km/h}]$
- $T(v)$ est un ensemble de termes ou valeurs linguistiques (petite, moyenne, grande)

I.2.5 Les règles linguistiques

L'idée principale des systèmes basés sur la logique floue, est d'exprimer la connaissance humaine sous forme des règles linguistiques de forme **si... Alors...** chaque règle a deux parties:

- Partie antécédente (prémisse ou condition), exprimée par **si...**
- Partie conséquence (conclusion) exprimée par **Alors.**

La partie antécédente est la description de l'état du système. La partie conséquence exprime l'action que l'opérateur doit exécuter. Chaque règle floue est basée sur l'implication floue.

Il y a plusieurs formes de règles floues, la forme générale est :

Si (un ensemble des conditions est satisfait) **Alors** (un ensemble des conséquences peut être exécuté).

I.2.6 Caractéristiques des ensembles flous [13]

Il existe 4 caractéristiques essentielles caractérisant l'ensemble flou de l'ensemble booléen: Le type, Le noyau, la hauteur et le support.

I.2.6.1 Le type

Les ensembles flous peuvent être présentés par des fonctions d'appartenance triangulaire, trapézoïdale, gaussienne, sigmoïde,.....etc.

I.2.6.2 Le support

C'est l'ensemble des éléments de X qui appartiennent au moins un peu à A . Il est défini par :

$$S(A) = \{x \in X / \mu_A(x) > 0\} \quad (\text{I.1})$$

- L'élément x de X tel que $\mu_A(x) = 0.5$ est appelé « **point de commutation** ».
- Un ensemble flou dont le support est un singleton dans X avec $\mu_A(x) = 1$ est appelé « **Singleton flou** » [14].

I.2.6.3 La hauteur

C'est la valeur maximale de la fonction d'appartenance, généralement, égale à 1, Elle est définie par :

$$h(A) = \max_{x \in X} \{\mu_A(x)\} \quad (\text{I.2})$$

I.2.6.4 Le noyau

$$\text{Défini par : } \eta(A) = \{x \in X / \mu_A(x) = 1\} \quad (\text{I.3})$$

C'est l'ensemble booléen de tous les éléments appartenant de façon absolue à l'ensemble flou A. Quand $\eta(A) = v \in]0,1]$, celui-ci est appelée 'valeur modale' de A. Pour la fonction d'appartenance triangulaire, la valeur modale correspond à la valeur du sommet.

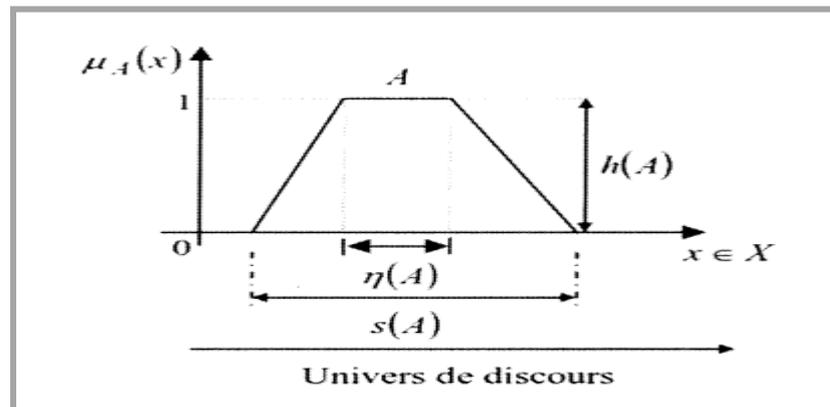


Fig.I.3 : Caractéristiques d'un sous ensemble flou A.

I.2.7 Fonction d'appartenance [15]

Un ensemble flou est défini par sa « **fonction d'appartenance** », qui correspond à la notion de « **fonction caractéristique** » en logique classique. Supposons que nous voulons définir l'ensemble des personnes de « **taille moyenne** ». En logique classique, nous conviendrons par exemple que les personnes de taille moyenne sont celles dont la taille est comprise entre 1,60 m et 1,80 m. La fonction caractéristique de l'ensemble représentée dans la (fig.I.4) donne « 0 » pour les tailles hors de l'intervalle [1,60 m ; 1,80 m] et « 1 » dans cet intervalle. L'ensemble flou des personnes de « taille moyenne » sera défini par une « fonction d'appartenance » qui diffère d'une fonction caractéristique par le fait qu'elle peut prendre

n'importe quelle valeur dans l'intervalle [0, 1]. A chaque taille possible correspondra un « degré d'appartenance » à l'ensemble flou des « tailles moyennes » (fig.I.5), compris entre 0 et 1.

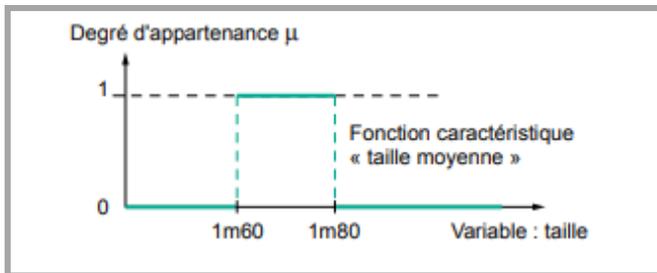


Fig.I.4 : fonction caractéristique

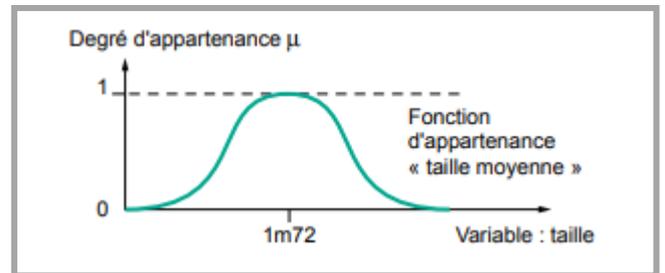


Fig.I.5 : fonction d'appartenance

Parmi les fonctions d'appartenance les plus utilisé on cite [16] :

I.2.7.1 La fonction triangulaire

$$\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \quad (I.4)$$

I.2.7.2 La fonction trapézoïdale

$$\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{c-x}{c-b}\right), 0\right) \quad (I.5)$$

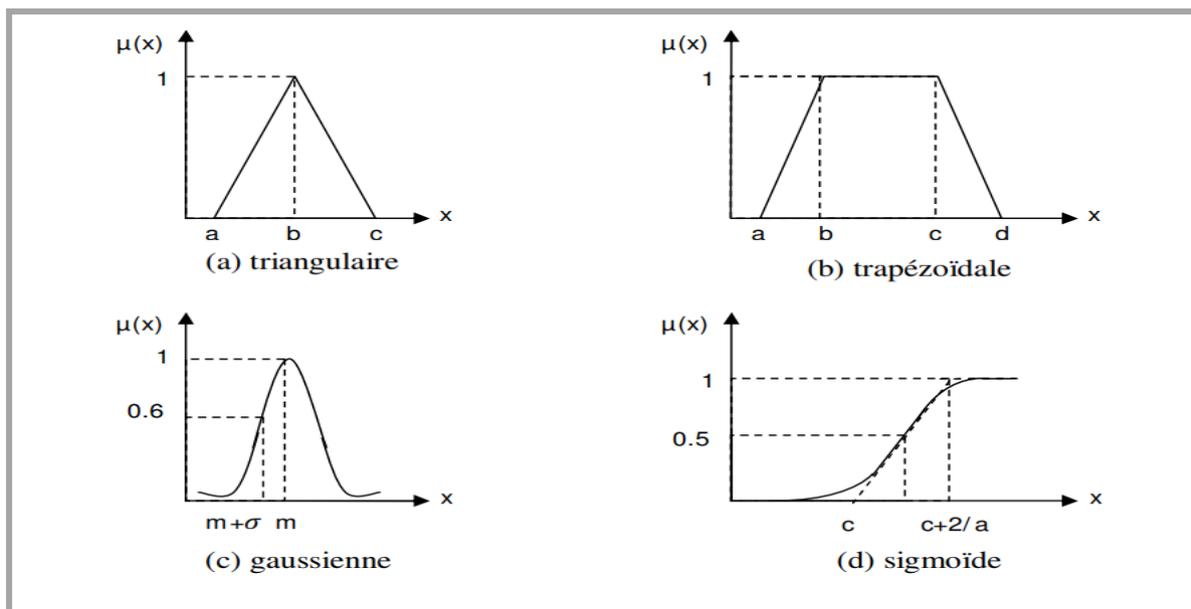


Fig.I.6 : Formes usuelles des fonctions d'appartenance.

I.2.7.3 La fonction gaussienne

$$\mu(x) = e^{\left(-\frac{(x-m)^2}{2\sigma^2}\right)} \quad (\text{I. 6})$$

I.2.7.4 La fonction sigmoïde

$$\mu(x) = \frac{1}{1+e^{(-a(x-c))}} \quad (\text{I. 7})$$

I.2.8 Les opérations sur les ensembles flous [16]

Soient A et B deux ensembles flous dans U ayant respectivement μ_A, μ_B comme fonctions d'appartenance. L'union, l'intersection et la complémentation des ensembles flous sont définis à l'aide de leurs fonctions d'appartenance.

- **L'Union** : L'union de deux sous-ensembles flous A et B de E est un sous-ensemble flou $A \cup B$ qui est défini par le plus grand sous-ensemble flou qui contient A et qui contient B . Sa fonction d'appartenance est donnée par :

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)]; \forall x \in E \quad (\text{I. 8})$$

- **L'Intersection** : L'intersection de deux sous-ensembles flous A et B de E est un sous-ensemble flou $A \cap B$ qui est défini par le plus petit sous-ensemble contenu à la fois dans A et dans B . Sa fonction d'appartenance est donnée par :

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]; \forall x \in E \quad (\text{I.9})$$

- **Le Complément** : Le complément d'un sous-ensemble flou A de B est un ensemble flou dénoté par \bar{A} dont la fonction d'appartenance est donnée par :

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x); \forall x \in E \quad (\text{I. 10})$$

- **L'Inclusion** : Soit les deux sous-ensembles flous A et B de l'ensemble E , on dira que A est inclus dans B si :

$$\forall x \in E : \mu_A \leq \mu_B(x) \quad (\text{I. 11})$$

Et on notera alors : $A \subset B$

I.3 La commande floue [17]

La commande floue est un domaine d'application de la théorie des ensembles flous qui a été proposé par L. Zadeh et développé après par plusieurs chercheurs comme (Mamdani, Assilian, etc.). Son but est de traiter des problèmes de commande de processus (machines-outils, procédés industriels, véhicules sans pilote...etc.), c'est-à-dire de gérer un processus en fonction d'une consigne donnée par action sur les variables qui décrivent le processus, mais son approche est différente de l'automatique. Elle se sert des connaissances des experts ou d'opérateurs qualifiés travaillant sur le processus.

I.3.1 Le principe de la commande floue

La commande floue a pour but de traiter des problèmes de commande de processus à partir de connaissances du comportement du procédé formulées sous forme linguistique (floue) [18].

Exemple :

- ✓ Commande de véhicule autonome,
- ✓ Commande de température d'une serre,
- ✓ Régulation de niveau complexe,
- ✓ Commande avec capteur imprécis (Camera...).

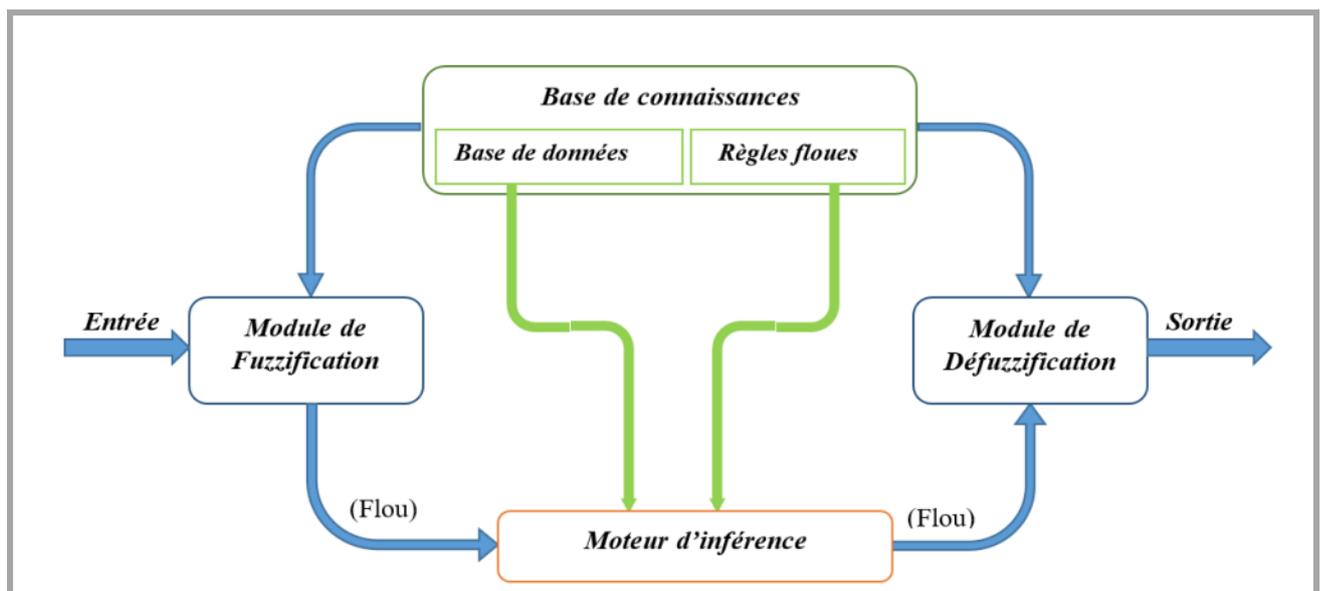


Fig.I.7: Structure de base d'un contrôleur flou [11].

La configuration de base d'un contrôleur flou comprend quatre parties :

1. Interface de fuzzification.
2. Base de connaissances.
3. Moteur d'inférence floue.
4. Interface de défuzzification.

I.3.1.1 Interface de fuzzification [19]

La fuzzification est le premier traitement qui entre en compte dans la structure d'un contrôleur flou. Elle consiste à donner un degré d'appartenance à une valeur réelle d'entrée en fonction des ensembles flous. Dans un contrôleur flou, il faut rendre floues (fuzzifier) les entrées et les sorties du système. La fuzzification des variables est une étape importante du processus de mise en œuvre. Les caractéristiques de cette étape sont habituellement déterminées par des experts ou des opérateurs qualifiés travaillant sur le processus et recourant le plus souvent à leurs connaissances. De plus, les performances du contrôleur flou seront influencées par la fuzzification. Les étapes de la fuzzification consistent à :

- ✓ Etablir les variables linguistiques,
- ✓ Etablir les quantificateurs flous (nombre de valeurs linguistiques),
- ✓ Attribuer une signification numérique à chaque quantificateur flou : Fonction d'appartenance.

I.3.1.2 Base de connaissances [18]

La conception d'une base de connaissances représente la phase de conception des systèmes experts. Elle comprend la base de données et la base des règles floues.

➤ La base de données

Contient la définition des ensembles flous, les facteurs d'échelle pour la normalisation des ensembles de référence et la partition de l'espace flou d'entrée et de sortie.

➤ La base des règles floues

Elle rassemble l'ensemble des règles floues de type « **Si-Alors** » décrivant en termes linguistiques basés sur la connaissance d'un expert le comportement dynamique du système :

$$R_l : \text{si } x_1 \text{ est } A_1^l \text{ et } \dots \text{ et } x_n \text{ est } A_n^l \text{ Alors } u_l \text{ est } B^l \quad (\text{I. 12})$$

Avec : $[x_1, \dots, x_n]$: les entrées du régulateur.

u_l : La sortie du régulateur.

Chaque régulateur activé donne un sous-ensemble flou de sortie.

D'une manière générale, la base de règles d'un contrôleur flou doit respecter certaines conditions afin d'assurer le bon fonctionnement de ce dernier. Parmi celles-ci citons:

- La complétude : une base de règles d'un contrôleur flou est dite complète si, pour chaque vecteur d'entrée, il existe au moins une règle floue activée. Afin d'assurer cette propriété, les fonctions d'appartenance doivent couvrir toutes les plages possibles des variables d'entrée. L'utilisation de fonctions d'appartenance triangulaires régulièrement réparties respecte la propriété de complétude.
- La consistance : une base de règles d'un contrôleur flou est dite inconsistante, s'il existe deux règles floues ayant la même prémisse mais des conclusions différentes. La propriété de consistance permet d'éviter les contradictions dans une base de règles.

I.3.1.3 Moteur d'inférences floues [16]

C'est un mécanisme de décision ; il permet à partir d'un fait observé de la base des règles floues d'élaborer une décision en exploitant le raisonnement approximatif. Dans les inférences de régulateur par logique floue, ils interviennent les opérateurs **ET** et **OU**. L'opérateur **ET** s'applique aux variables à l'intérieur d'une règle tandis que l'opérateur **OU** lie les différentes règles.

Pour le réglage par logique floue, on utilise en générale une des méthodes suivantes:

- Méthode d'inférence max-min (Mamdani)
- Méthode d'inférence max-produit (Larsen)
- Méthode d'inférence somme-produit (Sugeno).

A. Méthode d'inférence max-min (méthode de Mamdani)

La méthode d'inférence max-min est réalisée, au niveau de la condition l'opérateur «**ET**» par la formulation du minimum. La conclusion dans chaque règle, introduite par «**ALORS**», lie le facteur d'appartenance de la prémisse avec la fonction d'appartenance de la variable de sortie est réalisé par la formation du minimum. Enfin l'opérateur «**OU**» qui lie les différentes règles est réalisé par la formation du maximum.

B. Méthode d'inférence max-produit (méthode de Larsen)

La méthode d'inférence max-produit est réalisée, au niveau de la condition, l'opérateur «**ET**» par la formation du produit. La condition dans chaque règle, introduite par «**ALORS**» est réalisée par la formation du produit. L'opérateur «**OU**», qui lie les différentes règles, est réalisé par la formation du maximum.

C. Méthode d'inférence somme-produit (Sugeno)

L'opérateur «**ET**» est réalisé par la formation du minimum, la conclusion de chaque règle floue a une forme polynomiale. La sortie est égale à la moyenne pondérée de la sortie de chaque règle floue [9].

I.3.1.4 Interface de défuzzification [20]

La défuzzification est un processus mathématique utilisé pour convertir un ensemble flou en nombre réel. Les ensembles flous générés par l'inférence floue dans les règles floues doivent être mathématiquement combinés pour obtenir un seul nombre en sortie d'un contrôleur flou. Pour un système flou avec plus d'une variable de sortie, la défuzzification est calculée pour chacune d'entre elles séparément mais de manière très similaire.

Il existe différents types de défuzzification. Les trois techniques les plus utilisées pour le processus de défuzzification sont :

- Méthode de centre de gravité.
- Méthode du maximum.
- Méthode de la moyenne des maxima.

A. Méthode de centre de gravité

La commande résultante représente le centre de gravité de l'ensemble flou inféré :

Dans le cas discret :

$$u_r = \frac{\sum_{i=1}^n u_i \mu_{res}(u_i)}{\sum_{i=1}^n \mu_{res}(u_i)} \quad (\text{I. 13})$$

n : Le nombre d'ensembles flous de la sortie du contrôleur flou.

Dans le cas continu:

$$u_r = \frac{\int u \cdot \mu_{res}(u) du}{\int \mu_{res}(u) du} \quad (\text{I. 14})$$

B. Méthode du maximum

Cette méthode, s'applique uniquement dans le cas où la fonction d'appartenance associée à l'ensemble de sortie n'admet qu'un seul maximum. On choisit comme sortie l'abscisse u_r correspondant à ce maximum :

$$\mu_A(u_r) = \max(\mu_A(u)) \quad (\text{I. 15})$$

C. Méthode de la moyenne des maxima

Dans cette méthode, la valeur de sortie est estimée par l'abscisse du point correspondant au centre de l'intervalle M pour lequel la fonction d'appartenance est maximale. Cette valeur est fournie par l'expression :

$$u_r = (\inf(M) + \sup(M)) / 2 \quad (\text{I. 16})$$

Où $\inf(M)$ et $\sup(M)$ sont respectivement les bornes inférieure et supérieure de l'intervalle M [16].

I.3.2 Les Différents types de contrôleurs flous [21]

Il existe deux types de contrôleurs flous, les contrôleurs flous de Mamdani et de Takagi-Sugeno. Les différences entre les types de contrôleurs flous résident principalement dans l'implémentation du moteur d'inférence et du défuzzificateur. La conception générale d'un contrôleur flou implique les aspects suivants :

- ✓ Désignation de l'univers du discours.
- ✓ Conception des ensembles flous et des fonctions d'appartenance pour les espaces d'entrée et de sortie.
- ✓ Définition des règles linguistiques qui décrivent le comportement dynamique avec l'aide d'un expert humain.
- ✓ Le concepteur doit décider comment le fuzzifier, le moteur d'inférence et le défuzzifier doivent être implémentés.

I.3.2.1 Contrôleur flou de type Mamdani

Un contrôleur flou de Mamdani est basé sur une collection de règles du type :

Si x est A_i et y est B_j Alors Z est C_k

Où A_i, B_j, C_k représentent des termes linguistiques auxquels est attribuée une signification floue que l'on suppose normalisée pour des entrées numériques précise x_0 et y_0 , un sous-ensemble flou de sortie, noté F , est généré selon la relation

$$\mu(z) = \max_{(i,j,k) \in I} \min (\mu_{A_i}(x_0), \mu_{B_j}(y_0), \mu_{C_k}(z)) \quad (\text{I. 17})$$

L'opération correspond à l'application de l'opérateur max sur l'ensemble des règles définies. La valeur numérique de sortie délivrée par le contrôleur flou est obtenue par défuzzification, classiquement implantée par la méthode du centre de gravité.

$$Z = \frac{\int z \cdot \mu_F(z) dz}{\int \mu_F(z) dz} \quad (\text{I. 18})$$

I.3.2.2 Contrôleur flou de type Takagi-Sugeno

Il est construit à partir d'une base de règles "Si ... alors ...". Les prémisses sont toujours exprimées linguistiquement mais les conclusions sont par de nature numérique. Elles peuvent prendre différentes formes : équations linéaires ou polynômiales, statiques ou dynamiques.

Par exemple le contrôleur flou de Takagi-Sugeno d'ordre zéro à des conclusions constantes, les règles étant donc formulées de manière suivante :

$$R_{(i,j,k)}: \text{Si } x \text{ est } A_i \text{ et } y \text{ est } B_j \text{ alors } Z = C_k$$

Où C_k représente une valeur réelle quelconque. Pour les valeurs d'entrée précise x_0 , et y_0 , la sortie z est évaluée selon le mécanisme de calcul suivant :

$$Z = \text{SUG}(x_0, y_0) = \frac{\sum_{(i,j,k) \in I} w_{i,j} \cdot c_k}{\sum_{(i,j,k) \in I} w_{i,j}} \quad (\text{I. 19})$$

$$\text{avec } w_{i,j} = \mu_{A_i}(x_0) \cdot \mu_{B_j}(y_0)$$

I.4 Avantages et Inconvénients de la commande floue [22]

Le réglage par logique floue réunit un certain nombre d'avantages et inconvénients essentiels.

Les avantages sont :

- ✓ Robustesse de la commande floue vis-à-vis des incertitudes.
- ✓ La possibilité d'implémenter des connaissances (linguistique) de l'opérateur de processus.
- ✓ La maîtrise du système à régler avec un comportement complexe (fortement non linéaire et difficile à modéliser).
- ✓ L'emploi possible aussi pour les processus rapides (grâce à des processeurs dédiés).

- ✓ La disponibilité de systèmes de développement efficaces soit pour microprocesseurs ou PC (solution logicielle), soit pour circuit intégrés (processeur dédiés, solution matérielle).

Par contre, les inconvénients sont :

- ✓ Le manque de directive précise pour la conception d'un contrôleur flou (choix des fonctions d'appartenances d'E/S du contrôleur flou, la base de règles floues, mécanisme d'inférences et la méthode de défuzzification)
- ✓ L'approche artisanale et non systématique (implémentation des connaissances de l'opérateur souvent difficile).
- ✓ L'impossibilité de la démonstration de la stabilité du circuit de réglage en toute généralité (en l'absence d'un modèle valable).
- ✓ La possibilité d'apparition de cycles limites à cause du fonctionnement non linéaire.

I.5 Domaines d'applications

Parmi les nombreuses applications de la logique floue, la commande floue s'avère être le champ d'application le plus actif à travers le monde, Nous donnons quelques exemples d'application de la commande floue :

- ✓ Commande de tubes broyeurs pour la fabrication du ciment (première réalisation en 1979 au Danemark).
- ✓ Commande de métros avec un fonctionnement plus confortable et économique et une précision d'arrêt augmentée (1987 à Sendai, Japon).
- ✓ Production du fer et de l'acier, purification, de l'eau, chaînes et robots de fabrication.
- ✓ Opérateurs, unités arithmétique, micro-ordinateurs.
- ✓ Traitements d'images.
- ✓ Systèmes de transport (train, métro, ascenseur, ... etc.).
- ✓ Systèmes de décision, diagnostic (médical, assurance).

I.6 Conclusion

La conception d'un contrôleur flou implique la détermination de sa structure et de ses paramètres. Dans beaucoup de cas, la structure est déterminée, empiriquement, en choisissant à priori le type de raisonnement approximatif linguistique ou relationnel, le nombre des sous-ensembles flous pour chaque variable d'entrée et en prenant toutes les combinaisons possibles pour construire la base de règles. Le réglage par essais successifs des nombreux paramètres est assez long et fastidieux. Diverses techniques d'auto-réglage, d'optimisation et d'apprentissage des systèmes flous ont été développées. Dans le prochain chapitre, on va détailler l'algorithme de chauve-souris qui est l'une des méta-heuristiques les plus récentes dans le domaine d'optimisation et qui sera appliqué dans le troisième chapitre pour optimiser un contrôleur flou de type Mamdani pour la commande d'un système de lévitation magnétique.

Chapitre II : L'Algorithme de Chauve-souris

II.1 Introduction

L'optimisation combinatoire est une branche de l'optimisation mathématique qui se concentre sur la résolution de problèmes impliquant des choix discrets. Elle implique la recherche de la meilleure solution possible parmi un grand nombre de possibilités. Les problèmes d'optimisation combinatoire peuvent être trouvés dans de nombreux domaines, tels que la planification de la production, la recherche opérationnelle, la logistique, la finance, l'ingénierie et les télécommunications [23].

Les problèmes d'optimisation combinatoire impliquent souvent la recherche d'une combinaison optimale des variables pour atteindre un objectif spécifique, tout en respectant des contraintes. Les méthodes d'optimisation combinatoire utilisent des algorithmes sophistiqués pour explorer toutes les solutions possibles et trouver la meilleure solution.

Parmi les méthodes d'optimisation existantes, on trouve les méta-heuristiques. Ces méthodes sont apparues dans les années 80 et forment une famille d'algorithmes d'optimisation visant à résoudre des problèmes d'optimisation difficile, pour lesquels on ne connaît pas de méthode classique plus efficace.

Dans ce chapitre, nous allons présenter l'une des méta-heuristiques, à savoir l'algorithme de chauve-souris (*Bat Algorithm*), qui est basé sur le comportement d'écholocation des chauves-souris. La capacité d'écholocation des microchiroptères est fascinante car ces chauves-souris peuvent trouver leurs proies et distinguer différents types d'insectes même dans l'obscurité totale. On va utiliser l'algorithme de chauve-souris dans notre projet en vue d'optimisation du contrôleur flou au niveau du chapitre 3.

II.2 Méthodes d'optimisation

II.2.1. Objectif d'optimisation

Un problème d'optimisation est défini par un ensemble de variable, un ensemble de contraintes sur ces variables et une fonction objective que l'on cherche à minimiser ou à maximiser. Chaque problème d'optimisation possède un espace de recherche qui représente l'ensemble de solutions possibles de problèmes. La résolution d'un problème d'optimisation, revient à chercher la ou les meilleures solutions, tous en satisfaisant l'ensemble de contraintes dans un temps d'exécution raisonnable [24].

II.2.2 les éléments d'optimisation [25]

L'optimisation est une des mathématiques consacrée à l'étude des minimums/maximums d'une fonction à une ou plusieurs variables sur un certain domaine de définition, de l'étude de leur existence à leur détermination, en général par la mise en œuvre d'un algorithme et par suite d'un programme.

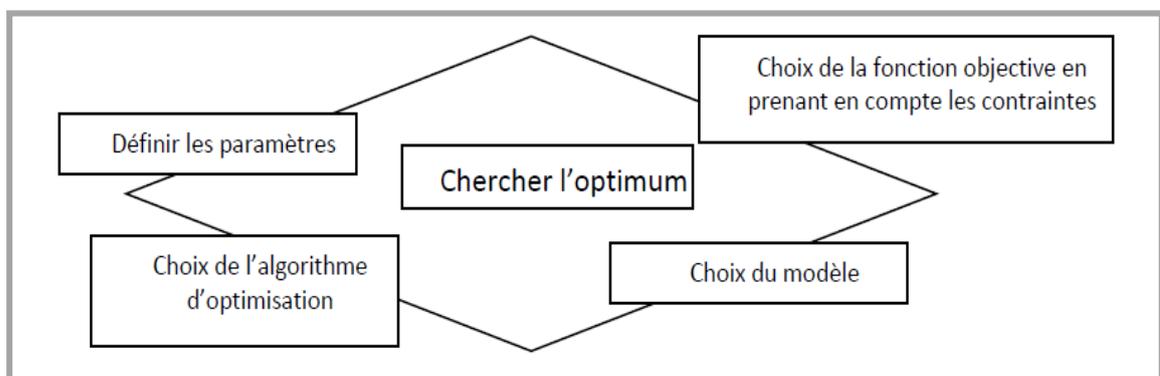


Fig.II.1 : Eléments essentiels pour l'optimisation.

Pour bien mener une opération, des éléments sont indispensables et conditionnent la solution trouvée. La figure II.1 présente les quatre éléments essentiels à la résolution d'un problème d'optimisation.

En général, beaucoup de paramètres sont indispensables. Certains paramètres ont une influence sur la fonction choisie. Les paramètres les plus influents sont :

- **Une fonction objective** : Définie l'objectif à atteindre. Le problème est formulé en un problème d'optimisation par l'intermédiaire de la fonction objectif. C'est cette fonction qui est au centre de l'optimisation.
- **Un modèle du système étudié** : Doit être précis, robuste et malléable du système. Ce modèle doit être utilisable sur un domaine d'étude le plus largement possible.
- **Un algorithme d'optimisation** : Permet de trouver la solution. Différentes méthodes d'optimisation existent.

II.2.3. Classification [26]

Etant donné l'importance de problèmes d'optimisation combinatoire, de nombreuses méthodes de résolution ont été développées. Ces méthodes peuvent être classées en deux grandes catégories : les méthodes exactes et les méthodes approchées.

II.2.2.1 Les méthodes exactes

Les méthodes exactes sont une autre branche de l'optimisation combinatoire qui vise à trouver la solution optimale d'un problème en explorant toutes les solutions possibles. Ces méthodes peuvent garantir la solution optimale, mais peuvent être très coûteuses en temps de calcul.

Les méthodes exactes comprennent des algorithmes tels que la programmation linéaire en nombres entiers, la programmation dynamique et la méthode la recherche arborescente (branch & bound). Ces méthodes sont souvent utilisées dans des problèmes de taille relativement petite ou lorsque la solution optimale est absolument nécessaire, sauf la méthode de branch & bound est particulièrement utile pour les problèmes de grande taille.

II.2.2.2 Les méthodes approchées [27]

Les algorithmes approximatifs (heuristiques) permettent de trouver une solution dont le coût est proche du coût de la solution optimale. Ils ont l'avantage de permettre en un temps raisonnable de trouver une solution. Ils constituent donc une alternative très intéressante pour traiter les problèmes d'optimisation de grande taille si l'optimalité n'est pas primordiale. Les méthodes approchées sont fondées principalement sur diverses heuristiques, souvent spécifiques à un type de problème.

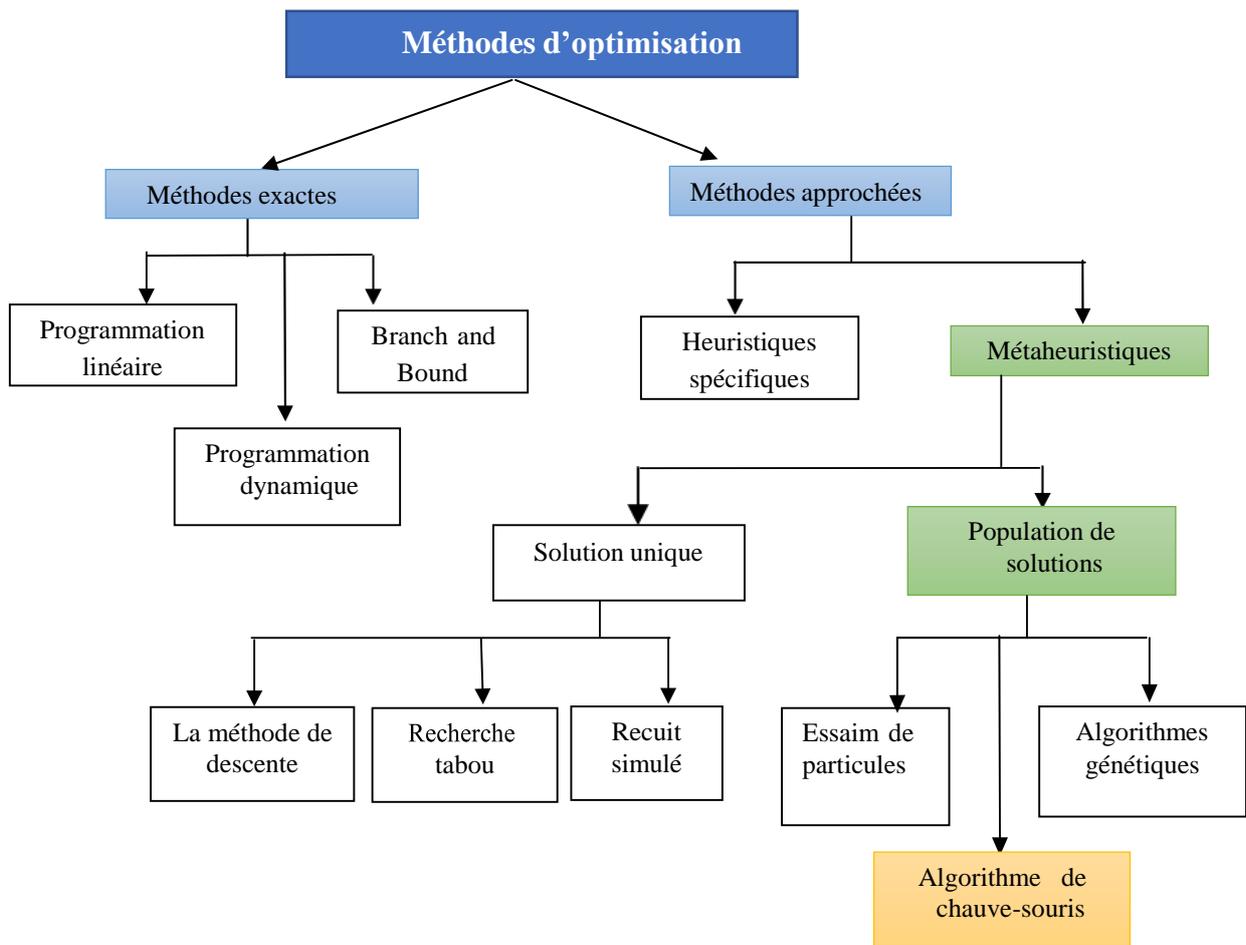


Fig.II.2 : Classification des méthodes d'optimisation [27].

Une heuristique est un algorithme qui a pour but de trouver une solution réalisable, sans garantie d'optimalité, contrairement aux méthodes exactes qui garantissent des solutions exactes. Comme les algorithmes de résolution exacte sont de complexité exponentielle pour les problèmes difficiles, il peut être plus judicieux de faire appel aux heuristiques pour calculer une

solution approchée d'un problème ou aussi pour accélérer le processus de résolution exacte. Des progrès importants ont été réalisés permettant l'apparition d'une nouvelle génération des algorithmes approchés puissants et généraux, souvent appelés méta-heuristiques.

II.3 Les Méta-heuristiques [28]

II.3.1 Définition

Le mot méta-heuristique est dérivé de la composition de deux mots grecs :

- Heuristique qui vient du verbe *heuriskein* et qui signifie 'trouver'
- Meta qui est un suffixe signifiant '*au-delà*', dans un niveau supérieur

Les méta-heuristiques forment un ensemble de méthodes utilisées en recherche opérationnelle et en intelligence artificielle pour résoudre des problèmes d'optimisation réputés difficiles. Résoudre un problème d'optimisation combinatoire, c'est trouver l'optimum d'une fonction, parmi un nombre fini de choix, souvent très grand.

Une méta-heuristique est une méthode algorithmique stochastique itérative capable de guider et d'orienter le processus de recherche dans un espace de solution, souvent très grand à des régions riches en solutions optimales. Le fait de rendre cette méthode abstraite et plus générique conduit à une vaste utilisation pour des champs d'applications différents.

II.3.2 Différents types de méta-heuristiques

Il existe un grand nombre de méta-heuristiques différentes, allant de la simple recherche locale à des algorithmes complexes de recherche globale. Ces méthodes peuvent être adaptées à une large gamme de problèmes différents. Elles peuvent être regroupées en deux grandes catégories ; les méta-heuristiques à population, et les méta-heuristiques à parcours.

II.3.2.1 Les méta-heuristiques à population

Le principe des méta-heuristiques à population ou encore dite évolutives est de faire évoluer un ensemble de solutions vers l'optimum cherché. Cette évolution se fait à partir des transformations et des coopérations entre les individus qui représentent individuellement une

solution de l'espace total du problème. Parmi ces méthodes, nous distinguons essentiellement les algorithmes génétiques, et les colonies de fourmis, l'algorithme de chauve-souris, etc.

II.3.2.2 Les méta-heuristiques à parcours

Les méta-heuristiques à parcours ne travaillent qu'avec une seule solution, cette dernière est modifiée à chaque itération de l'algorithme. L'espace des solutions est visité par la solution, et celle-ci est modifiée pour être améliorée au cours de ce parcours. Les méta-heuristiques à parcours les plus classiques sont : le recuit simulé, la recherche Tabou.

II.4 L'algorithme de chauve-souris

II.4.1 Description

L'algorithme des chauves-souris (*Bat Algorithm en anglais*) est l'une des méta-heuristiques les plus récentes proposé par Xin-She Yang en 2010 pour la résolution de problèmes d'optimisation [29]. Cet algorithme est basé sur le comportement de l'écholocation de microchiroptères. L'écholocation est un sonar biologique (à cause de manque de vision) qui permet détecter la distance, et ils ont aussi la capacité de faire la différence entre la nourriture/proie et les obstacles. La méthode s'appuyé sur une technique de régulation de fréquence pour augmenter la diversité des solutions dans la population, ainsi il tente d'équilibrer l'exploration et l'exploitation pendant le processus de recherche, en imitant les variations des taux d'émission d'impulsions et la vitesse des chauves-souris lors de la recherche de proies [30].

L'algorithme standard de chauve-souris présente de nombreux avantages; L'un d'entre eux est qu'il peut obtenir une convergence rapide aux étapes initiales en passant de l'exploration à l'exploitation. Cela en fait un algorithme efficace lorsqu'une solution rapide est nécessaire. Afin d'améliorer les performances, de nombreuses modifications ont été ajoutées pour augmenter la diversité de la solution et pour améliorer les performances de l'algorithme de chauve-souris standard comme mentionné précédemment [31].

II.4.2 Echolocation des chauves-souris

L'écholocation est un processus utilisé par les chauves-souris pour naviguer et trouver de la nourriture dans l'obscurité. Les chauves-souris émettent des sons à haute fréquence qui rebondissent sur les objets environnants et sont ensuite captés par les oreilles de la chauve-

souris. En utilisant ces informations, la chauve-souris peut déterminer la distance et la position des objets dans son environnement. Et ils ont aussi la capacité de faire la différence entre la nourriture/proie et les obstacles [27].

L'algorithme de chauve-souris s'inspire de ce processus naturel pour résoudre des problèmes d'optimisation combinatoire. Les solutions sont représentées comme des positions dans un espace de recherche et les chauves-souris virtuelles émettent des sons pour explorer cet espace. Les sons émis par chaque chauve-souris représentent une solution potentielle et sont modifiés au fil du temps pour trouver des solutions de meilleure qualité.

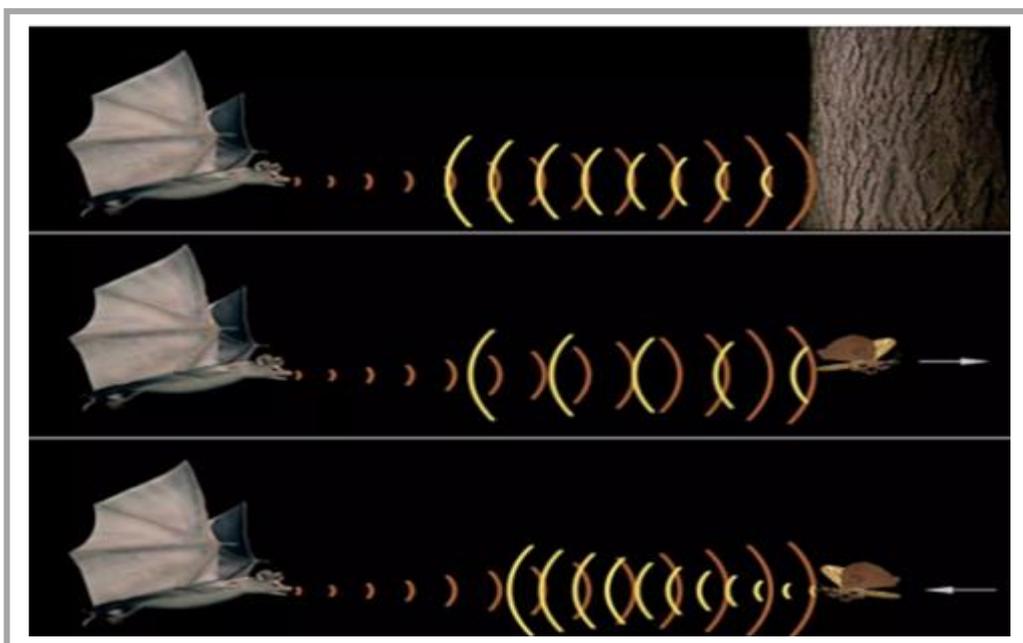


Fig. II.3 : Principe de l'écholocation [27].

II.4.3 L'algorithme standard de chauve-souris [30]

L'algorithme des chauves-souris tente de modéliser le comportement d'écholocation. Il repose essentiellement sur le scénario suivant: des chauves-souris volent aléatoirement en vue de trouver de la nourriture, celles-ci n'ont aucune connaissance sur l'endroit où se trouvent leurs proies, cependant elles peuvent calculer la distance qui sépare chacune d'elles de la cible au moyen de l'écholocation. Leur objectif commun est non seulement d'atteindre leurs proies

mais aussi de trouver la meilleure stratégie de chasse. La stratégie la plus efficace consiste à suivre la chauve-souris la plus proche de la volée.

L'algorithme des chauves-souris est une méthode d'optimisation où toutes les chauves-souris collaborent en population dans le but d'atteindre un but collectif. Ce dernier est formulé par une fonction dite fonction objective.

L'algorithme des chauves-souris, est essentiellement composé des trois étapes suivantes, et qui seront répétées jusqu'à atteindre un certain nombre d'itérations :

1. Evaluer chaque chauve-souris en calculant sa fonction objective.
2. Mettre à jour les meilleures solutions locales et globales.
3. Mettre à jour les positions, les vitesses et les fréquences de chacune des chauves-souris.

Les deux premières étapes sont assez simples. L'évaluation de chacune des chauves-souris s'effectue à l'aide de la fonction objective. Tandis que la mise à jour des meilleures solutions locales et globales s'effectue en comparant la fonction objective courante de chacune des chauves-souris avec la fonction objective de la meilleure solution qu'elle a rencontré jusqu'ici, puis de choisir la meilleure d'entre elles. Quant à la recherche de la solution globale, il suffira de comparer toutes les meilleures solutions locales et garder la meilleure.

La troisième étape est la plus critique, en effet, le mouvement des chauves-souris est responsable de l'efficacité de l'algorithme. Les règles définissant la nouvelle solution et mettant à jour la position et la vitesse de chacune des chauves-souris sont les suivantes :

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (\text{II.1})$$

$$v_i(t) = v_i(t - 1) + (x_i(t) - x^*)f_i \quad (\text{II.2})$$

$$x_i(t) = x_i(t - 1) + v_i(t) \quad (\text{II.3})$$

Avec : f_i est la fréquence d'émission de la chauve-souris x_i et qui appartient à la gamme $[f_{max}, f_{min}]$ correspondant à la gamme de longueurs d'ondes $[\lambda_{max}, \lambda_{min}]$. $\beta \in [0,1]$ est un vecteur aléatoire issu d'une distribution uniforme. x^* Est la meilleure solution globale courante qui est déterminée en comparant toutes les solutions parmi tous les n chauves-souris.

Pour la recherche locale, une fois une solution est sélectionnée parmi les meilleures solutions courantes, une nouvelle solution pour chaque chauve-souris est générée localement en utilisant le parcours aléatoire.

$$x_{new} = x_{old} + \varepsilon A(t) \quad (\text{II.4})$$

Avec $\varepsilon \in [-1, 1]$ est un nombre aléatoire, $A(t)$ est la moyenne d'intensité de toutes les chauves-souris à l'instant t [28].

En se basant sur ces règles et ces explications, les étapes de base de l'algorithme de chauve-souris peuvent être résumées comme indiqué dans la figure ci-dessous (Fig.II.4).

1. Définir la Fonction objective : $f(x)$, $x = (x_1, \dots, x_d)$
2. Initialiser la position de chauve-souris x_i et la vitesse v_i , $i = 1, 2, \dots, n$
3. Définir la fréquence f_i de chaque position x_i
4. Initialiser le taux de pulsation r_i et l'intensité A_i
5. Tant que ($t < \text{nombre maximum d'itérations}$)
 - Générer de nouvelles solutions en ajustant la fréquence et en actualisant les vitesses et les positions / solutions. (Équations II.1, II.3).
 - Si ($\text{rand} > r_i$)
Sélectionnez une solution parmi les meilleures solutions Générer une solution locale autour de la meilleure solution sélectionnée x^*
 - Fin si
 - Générer une nouvelle solution en volant aléatoirement
 - Si ($\text{rand} < A_i$ et $f(x_i) < f(x^*)$)
Accepter de nouvelles solutions, Incrémenter r_i et réduire A_i
 - Fin si
 - Classer les chauves-souris selon sa fonction d'objective et trouver la meilleure solution x^*
6. Fin tant que
7. Afficher les résultats donnés par la meilleure solution x^*

Fig.II.4 : Pseudo code de l'algorithme des chauves-souris [29].

II.4.4 Génération de la population initiale [16]

Le choix de la population initiale d'individus conditionne fortement la rapidité de convergence de l'algorithme. Si la position de l'optimum dans l'espace d'état est totalement

inconnue, il est naturel de générer aléatoirement des individus en faisant des tirages uniformes dans chacun des domaines associés aux composantes de l'espace d'état en veillant à ce que les individus produits respectent les contraintes. Si des informations a priori sur le problème sont disponibles, il est naturel de générer les individus dans un sous domaine particulier afin d'accélérer la convergence.

II.4.5 Fonction objective [16]

La fonction objective (*coût*) est généralement l'étape dans laquelle on mesure la performance de chaque individu. Pour pouvoir juger la qualité d'un individu et ainsi le comparer aux autres, il faut établir une mesure commune d'évaluation. Aucune règle n'existe pour définir cette fonction, son calcul peut ainsi être quelconque, que ce soit une simple équation ou une fonction affine. La manière la plus simple est de poser la fonction d'adaptation comme la formalisation du critère d'optimisation. Dans ce travail, on a choisis l'erreur quadratique moyenne comme fonction d'évaluation.

II.4.6 Critère d'arrêt [16]

Le cycle de génération et de sélection de population est répété jusqu'à ce qu'un critère d'arrêt soit satisfait ; ce critère peut être notamment un nombre maximum d'itérations, un temps maximal de calcul, une valeur de fitness minimale, ou/et une convergence vers une solution satisfaisante.

II.5 Domaine d'application de l'algorithme de chauve-souris

Les algorithmes de chauve-souris ont été appliqués dans presque tous les domaines, tels que [32] :

- ✓ l'optimisation de réseaux de télécommunications
- ✓ la planification de tournées de livraison
- ✓ la conception de circuits électroniques
- ✓ la gestion de stocks, traitement d'image
- ✓ la conception de réseaux neuronaux
- ✓ l'optimisation de systèmes d'énergie renouvelable, etc.

II.6 Conclusion

En conclusion, L'algorithme de chauve-souris est une méthode d'optimisation prometteuse qui a montré des résultats encourageants dans diverses applications. Bien que cette méthode présente des avantages par rapport à d'autres méthodes d'optimisation combinatoire, elle présente également certaines limites et des améliorations peuvent être apportées pour améliorer son efficacité. Cependant, l'algorithme de chauve-souris a un grand potentiel pour résoudre des problèmes encore plus complexes dans le futur grâce à son adaptabilité et à sa capacité à trouver des solutions optimales en peu de temps. De plus, l'utilisation de l'algorithme de chauve-souris pour résoudre des problèmes d'optimisation combinatoire peut contribuer à améliorer les performances des systèmes dans de nombreux domaines et à relever les défis les plus complexes de notre monde en constante évolution.

Dans ce mémoire, l'algorithme de chauve-souris est utilisé pour l'optimisation des paramètres des fonctions d'appartenances des entrées et de sortie ainsi que la base de règles floues de contrôleur flou afin de garantir une bonne stabilisation d'un système de lévitation magnétique.

Chapitre III : Simulation & Interprétations

III.1 Introduction

Le contrôle de mouvement précis peut être compris au mieux par une sorte de dispositif de suspension sans contact et sans usure connu sous le nom de systèmes de lévitation magnétique est une technologie de pointe qui permet de faire léviter des véhicules sans contact avec une surface en utilisant des aimants. Cette technologie est utilisée dans diverses applications, notamment les trains à sustentation magnétique, les dispositifs de stockage de données et les expériences scientifiques [33].

Le problème abordé dans la classe des systèmes de suspension magnétique est de contrôler précisément la hauteur de la boule magnétique au-dessus du sol en la faisant léviter contre la force de gravité à l'aide d'électro-aimants.

Afin de mettre en évidence l'ensemble des caractéristiques des contrôleurs flous, nous avons procédé, dans la première partie de ce chapitre, à l'élaboration d'un régulateur flou de type Mamdani pour commander la position de la balle magnétique du système de lévitation magnétique.

Nous proposons, dans la deuxième partie, une stratégie d'optimisation basée sur l'algorithme des chauves-souris (BAT). Cette optimisation concerne les paramètres des prémisses et des conclusions de règles floues. En combinant les deux techniques d'intelligence artificielle à savoir la commande floue et l'algorithme BAT, il est possible de créer des systèmes de lévitation magnétique très précis et fiables qui peuvent être utilisés dans une variété d'applications industrielles et commerciales.

III.2 La commande floue du système de lévitation magnétique

III.2.1 Introduction

La commande floue est une méthode de contrôle qui permet de gérer des systèmes complexes en utilisant des règles linguistiques plutôt que des équations mathématiques. Elle est particulièrement adaptée aux systèmes non linéaires et incertains, tels que les systèmes de lévitation magnétique. Ces systèmes reposent sur la stabilisation d'un objet en suspension dans l'air grâce à des champs magnétiques. La commande floue permet d'optimiser la stabilité et la précision du système en ajustant les paramètres en fonction des conditions environnementales et des perturbations. Dans cette partie, nous allons explorer les principes de base de la commande floue appliquée aux systèmes de lévitation magnétique.

III.2.2 Le modèle mathématique de lévitation magnétique [34]

La structure du procédé de contrôle de la position de la balle magnétique de système de lévitation magnétique est donnée par la figure suivante :

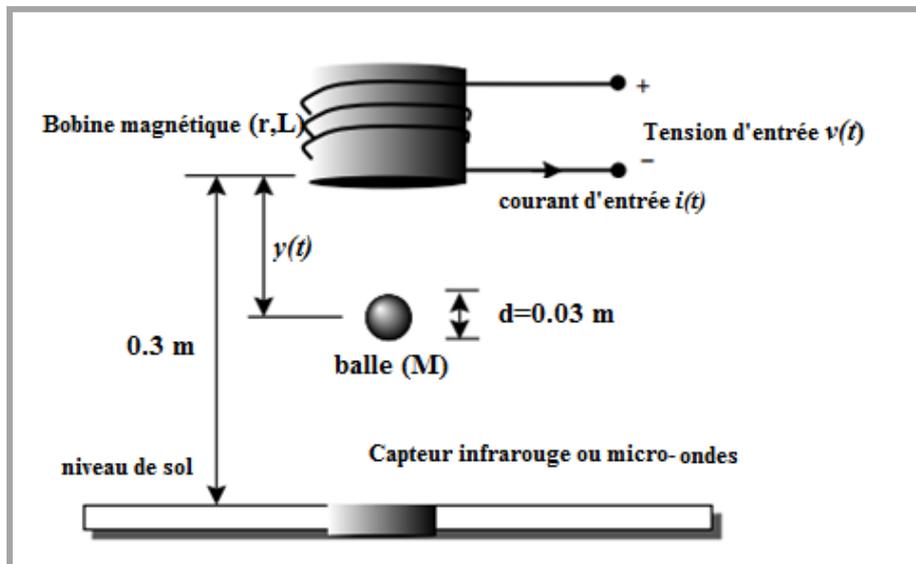


Fig.III.1: structure du procédé.

Le modèle du système de suspension de balle magnétique représenté sur la figure III.1 est défini par :

$$m \frac{d^2 y(t)}{dt^2} = mg - \frac{i^2(t)}{y(t)} \quad (\text{III.1})$$

$$v(t) = ri(t) + L \frac{di(t)}{dt} \quad (\text{III.2})$$

Avec :

$y(t)$: La position de la balle.

m : La masse de la balle.

g : L'accélération gravitationnelle.

r : La résistance d'enroulement.

L : L'enroulement inductance.

$v(t)$: La tension d'entrée.

$i(t)$: Le courant d'enroulement.

Les paramètres du système sont donnés par la table suivante :

Coefficient	valeur
m	0.1 kg
g	9.8 m/s ²
r	50 Ω
L	0.5 H

Tableau III.1: Paramètres du système.

La position de la balle est détectée par un capteur de position (capteur infrarouge) et elle est supposée être totalement détectable sur toute la gamme entre la bobine magnétique et le niveau du sol. Nous supposons que la balle reste entre la bobine et le grand niveau. Dans l'espace d'états, l'équation devient [34]:

$$\left\{ \begin{array}{l} \frac{dx_1(t)}{dt} = x_2(t) \\ \frac{dx_2(t)}{dt} = g - \frac{x_3^2(t)}{mx_1(t)} \\ \frac{dx_3(t)}{dt} = -\frac{r}{L}x_3(t) + \frac{1}{L}v(t) \end{array} \right. \quad (\text{III.3})$$

Où $[x_1(t), x_2(t), x_3(t)]^T = [y(t), \frac{dy(t)}{dt}, i(t)]^T$. Notons que les non-linéarités sont induites a $x_3^2(t)$ et $\frac{1}{x_1(t)}$ et le terme $\frac{dx_2(t)}{dt}$ dans l'équation (III.3), en linéarisant le modèle de

l'usine en équation (III.3), en supposant que la balle est initialement située à $x_1(0) = y(0)$, On peut trouver un système linéaire en calculant la matrice jacobéenne en $y(0)$.

La forme linéaire de l'espace d'état du système de suspension de balle magnétique est donnée comme suit [34] :

$$\left[\begin{array}{l} \frac{dx_1(t)}{dt} = x_2(t) \\ \frac{dx_2(t)}{dt} = \frac{g}{y(0)} x_1(t) - 2 \sqrt{\frac{g}{my(0)}} x_3(t) \\ \frac{dx_3(t)}{dt} = -\frac{r}{L} x_3(t) + \frac{1}{L} v(t) \end{array} \right. \quad (\text{III.4})$$

III.2.3 Structure du contrôleur flou

Le but de ce travail est de commander la position de la balle du système de lévitation magnétique. Le contrôleur flou utilisé est de type Mamdani où la sortie du contrôleur flou est la commande du système de lévitation magnétique (fig.III.2):

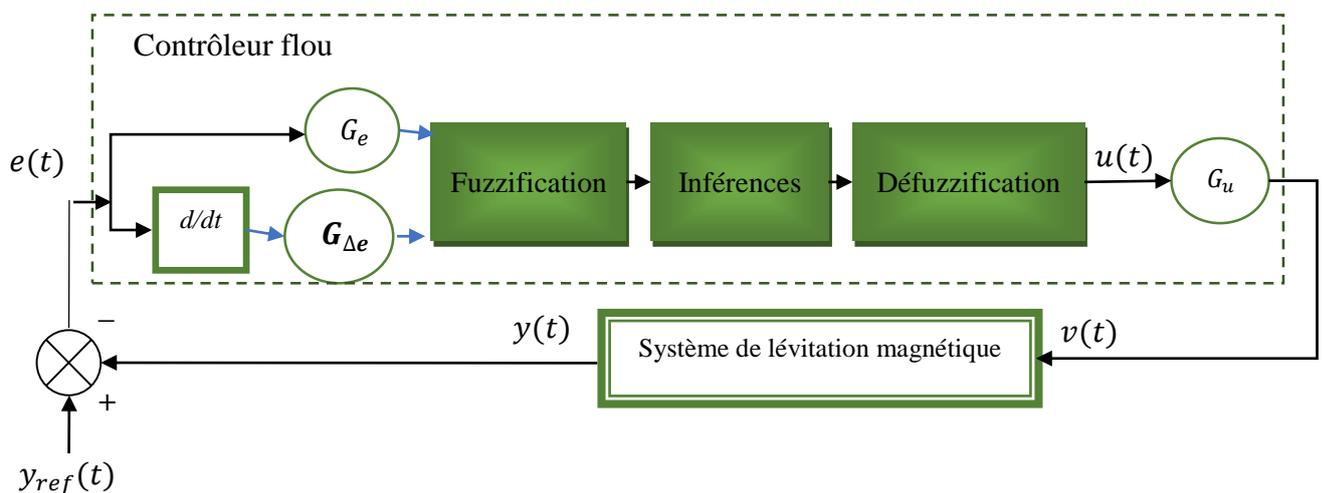


Fig.III.2 : structure du contrôleur flou de la position de la balle magnétique.

Où $e(t) = y_{ref}(t) - y(t)$ est la différence entre la sortie désirée $y_{ref}(t)$ et la sortie calculée $y(t)$. Pour avoir une flexibilité dans l'implantation du régulateur flou, les univers de discours des entrées et de sortie sont limités à un intervalle $[-1,1]$ déterminé par la normalisation

des entrées et de sortie. Pour ce faire, des gains d'adaptations (facteurs d'échelles : G_e , $G_{\Delta e}$, G_u) sont utilisés pour avoir la dynamique désirée.

La structure utilisée des contrôleurs flous est la suivante :

- Pour la Fuzzification : des fonctions d'appartenance de type triangulaire sont utilisées pour l'erreur $e(t)$ et sa variation $\Delta e(t)$.
- La base de règles est de MacVicar et Whelan, composée de 9 règles.
- Pour le mécanisme d'inférences, la méthode min max est utilisée.
- Pour la défuzzification : la méthode du centre de gravité est utilisée.

Pour choisir la classe du contrôleur flou, on a procédé à implémenter plusieurs contrôleurs flous, classe 3, classe 5 et classe 7, et selon les résultats obtenus, on va choisir la meilleure structure qu'on va optimiser par l'algorithme de chauve-souris dans la deuxième partie de simulation.

III.2.4 Résultats de simulation

III.2.4.1 Régulateur à 3 classes :

La figure III. 3 et le tableau III.2 montrent respectivement, la forme des fonctions d'appartenance utilisées pour les entrées et la sortie du contrôleur flou, ainsi que la base de règles pour un contrôleur flou de type Mamdani à trois classes.

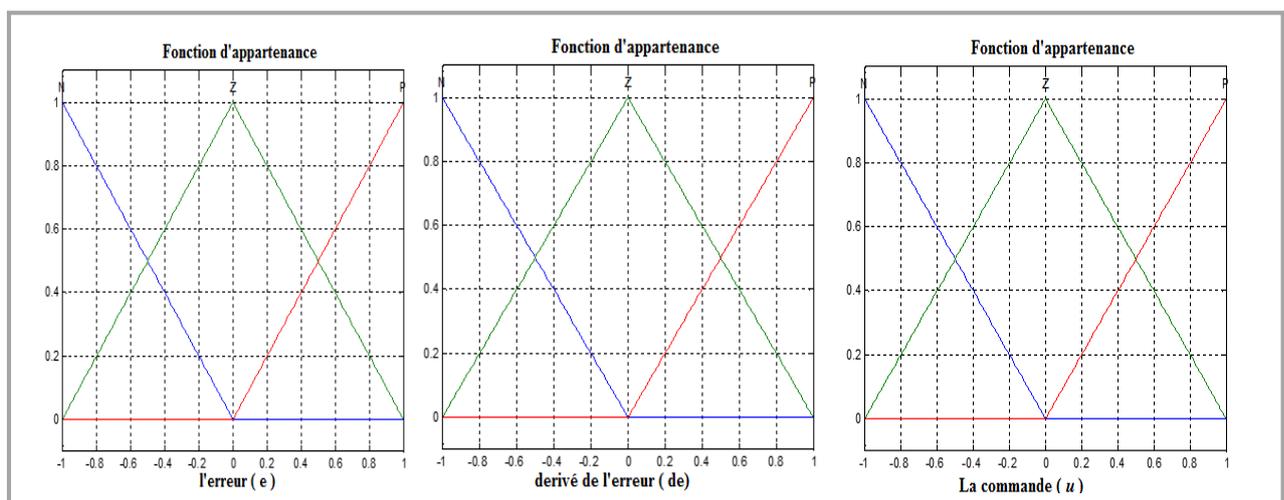


Fig.III.3: formes des fonctions d'appartenance (3x3).

$e \backslash \Delta e$	N	Z	P
N	P	P	Z
Z	P	Z	N
P	Z	N	N

Tableau III.2: Base de règles floues (3x3).

Avec : (N): Négatif ; (P): Positif ; (Z): Zéro.

Les gains de normalisation sont choisis par tâtonnement comme suit:

$$G_e = 0.05 \quad , \quad G_{\Delta e} = 0.05 \quad , \quad G_u = 20$$

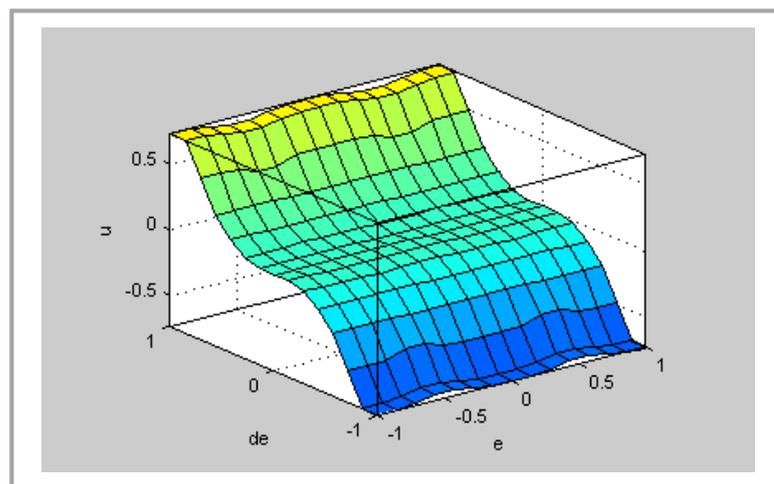


Fig.III.4 : surface caractéristique du régulateur flou 3 classes.

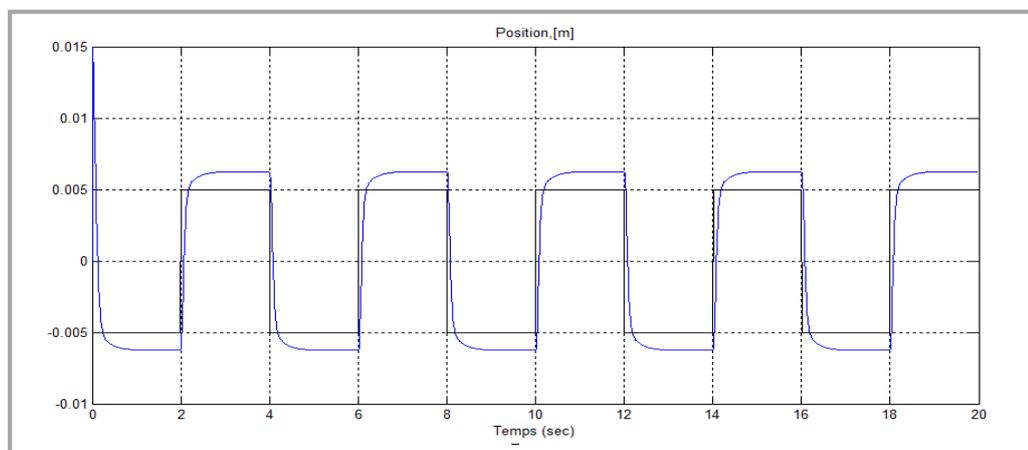


Fig.III.5 : Evolution de la position et la référence en fonction du temps.

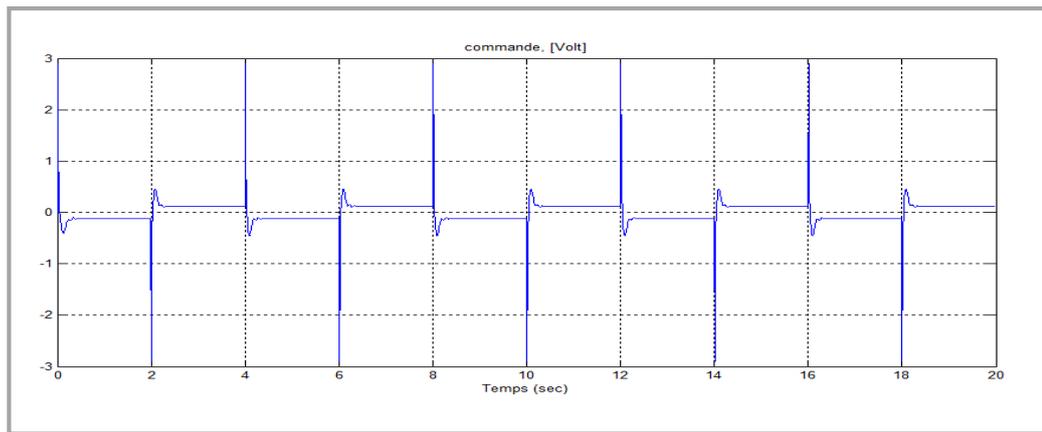


Fig.III.6: Evolution du signal de commande en fonction du temps.

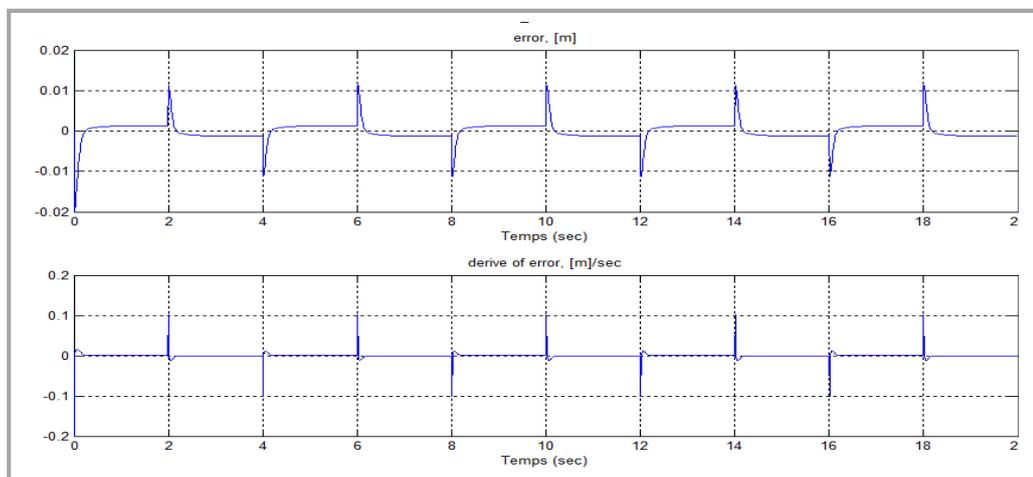


Fig.III.7: Evolution de l'erreur et sa variation.

III.2.4.2 Régulateur à 5 classes :

La figure III. 8 et le tableau III.3 montrent respectivement, la forme des fonctions d'appartenance utilisées pour les entrées et la sortie du contrôleur flou, ainsi que la base de règles pour un contrôleur flou de type Mamdani à cinq classes.

Les gains de normalisation sont choisis comme suit:

$$G_e = 0.05, \quad G_{\Delta e} = 0.05, \quad G_u = 20$$

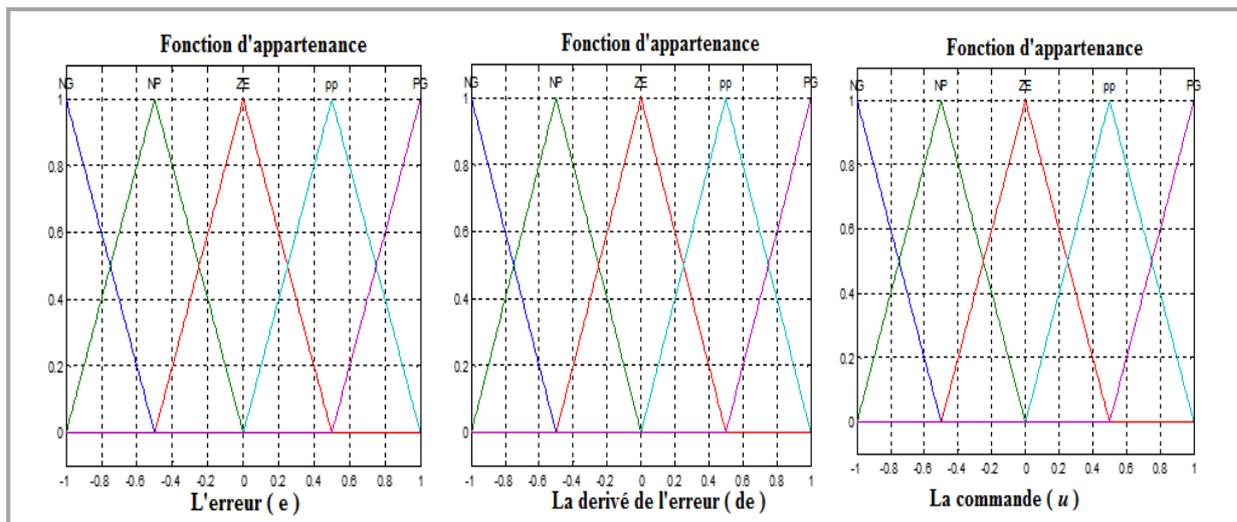


Fig.III.8: formes des fonctions d'appartenance (5x5).

$e \backslash \Delta e$	NG	NP	ZE	PP	PG
NG	PG	PG	PG	PP	ZE
NP	PG	PG	PP	ZE	NP
ZE	PG	PP	ZE	NP	NG
PP	PP	ZE	NP	NG	NG
PG	ZE	NP	NG	NG	NG

Tableau III.3 : Base des règles (5x5).

Avec: (NG): Négatif Grand ; (NP): Négatif Petit ; (ZE): Environ zéro ; (PP): Positif Petit ; (PG): Positif Grand.

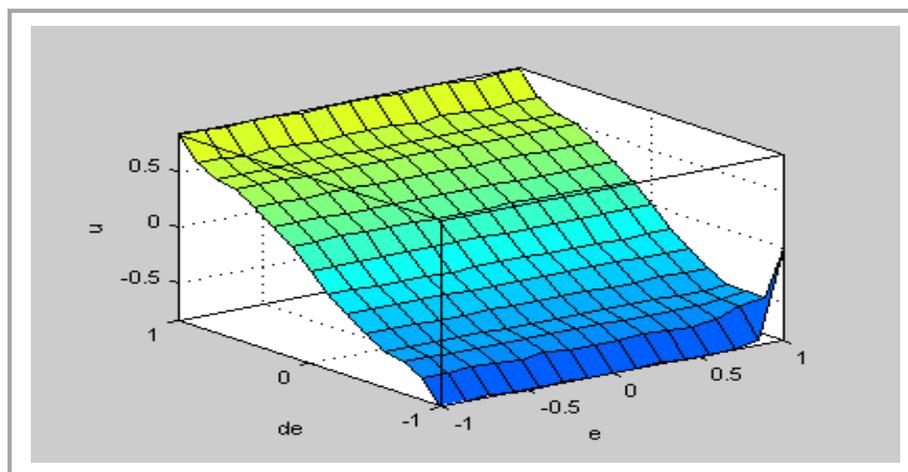


Fig.III.9: surface caractéristique du régulateur flou 5 classes.

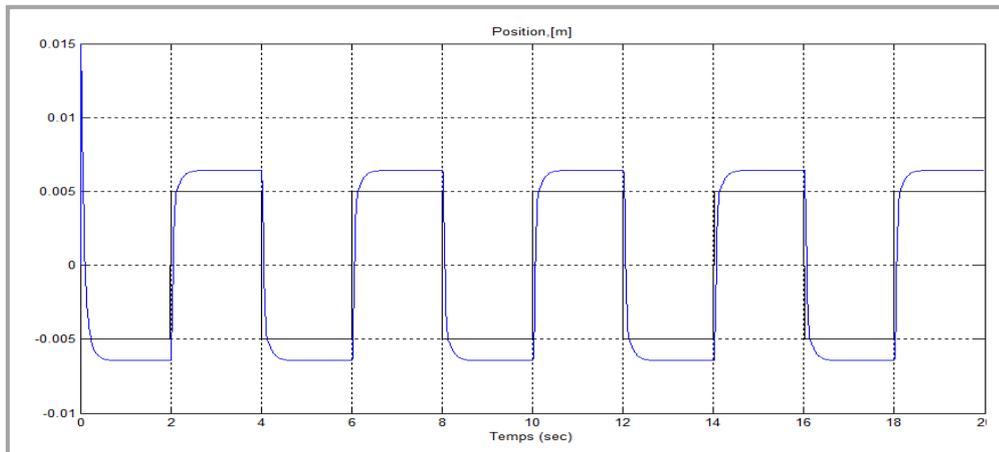


Fig.III.10 : Evolution de la position et la référence en fonction du temps.

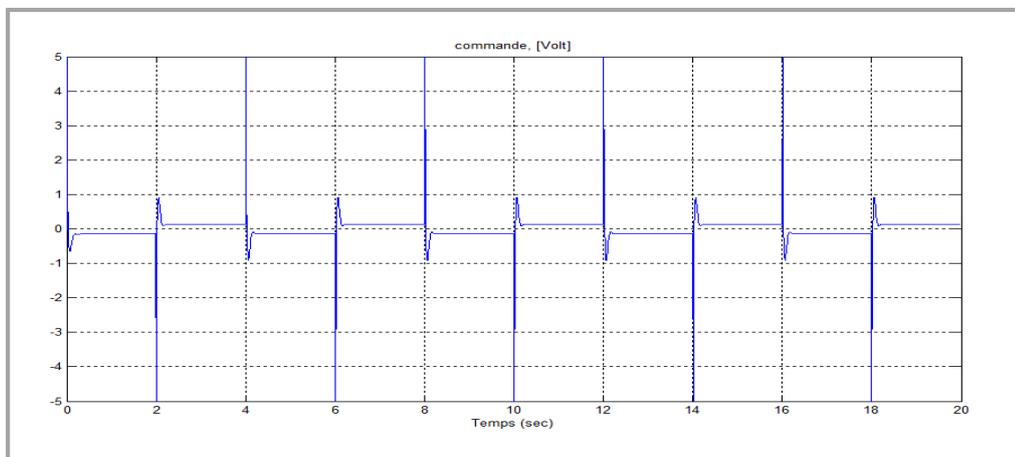


Fig.III.11: Evolution du signal de commande en fonction de temps.

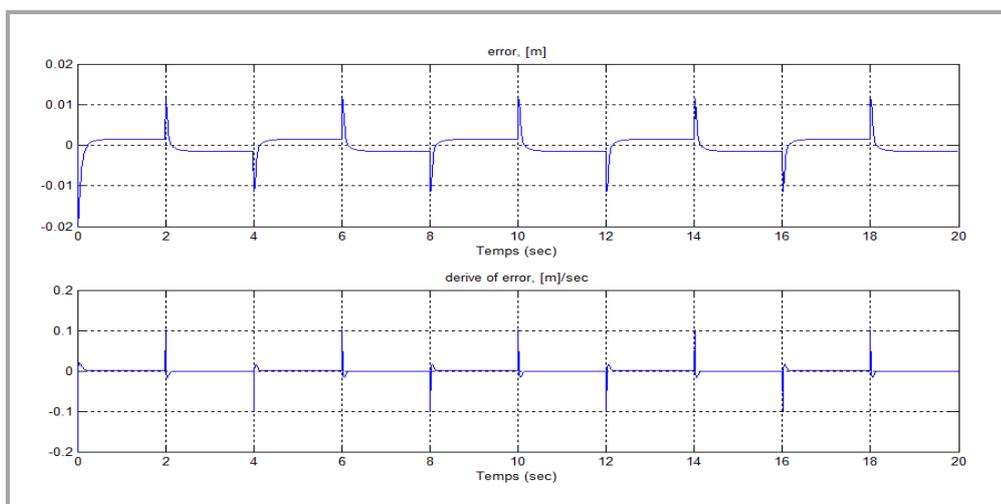


Fig. III.12: Evolution de l'erreur et dérivé d'erreur.

III.2.4.3 Régulateur à 7 classes :

La figure III. 13 et le tableau III.4 montrent respectivement, la forme des fonctions d'appartenance utilisées pour les entrées et la sortie du contrôleur flou, ainsi que la base de règles pour un contrôleur flou de type Mamdani à sept classes.

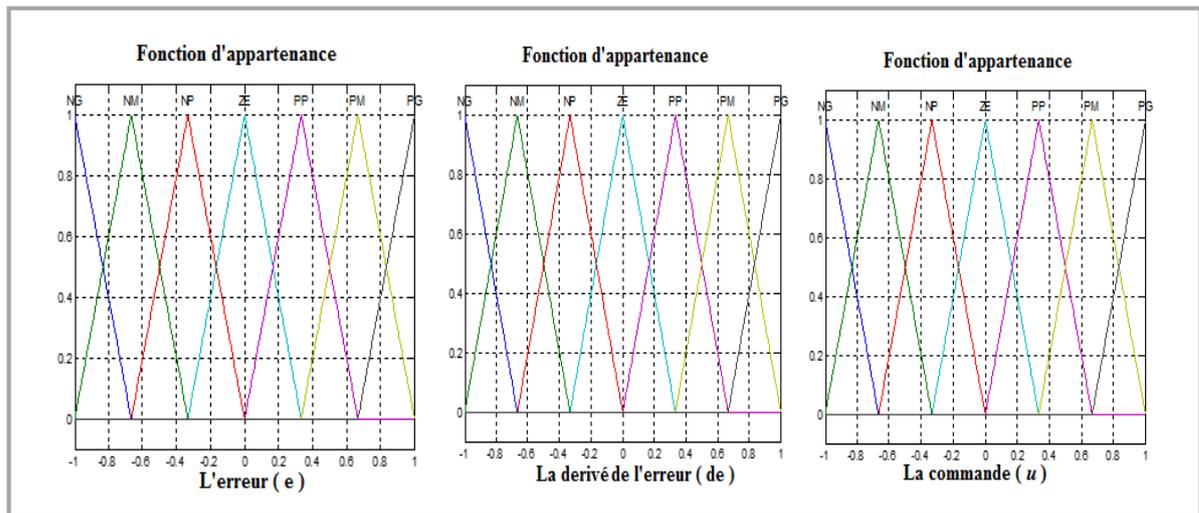


Fig.III.13 : formes des fonctions d'appartenance (7x7).

Les gains de normalisation sont choisis comme suit:

$$G_e = 0.03 \quad , \quad G_{\Delta e} = 0.03 \quad , \quad G_u = 15$$

Avec: (NG): Négatif Grand ; (NP): Négatif Petit ; (NM): Négatif Moyen ; (EZ): Environ zéro ; (PP): Positif Petit ; (PM): Positif Moyen ; (PG): Positif Grand.

e \ Δe	NG	NM	NP	ZE	PP	PM	PG
NG	PG	PG	PG	PG	PM	PP	ZE
NM	PG	PG	PG	PM	PP	ZE	NP
NP	PG	PG	PM	PP	ZE	NP	NM
ZE	PG	PM	PP	ZE	NP	NM	NG
PP	PM	PP	ZE	NP	NM	NG	NG
PM	PP	ZE	NP	NM	NG	NG	NG
PG	ZE	NP	NM	NG	NG	NG	NG

Tableau III.4: Base des règles (7x7).

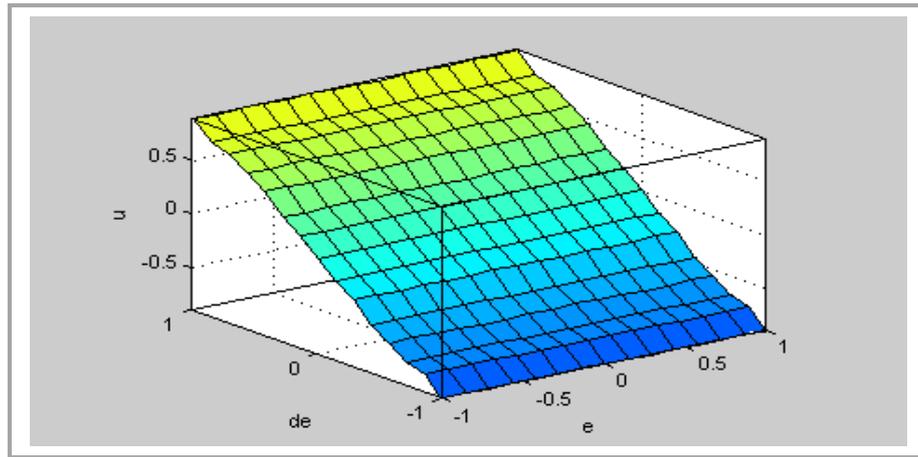


Fig.III.14: surface caractéristique du régulateur flou 7 classes.

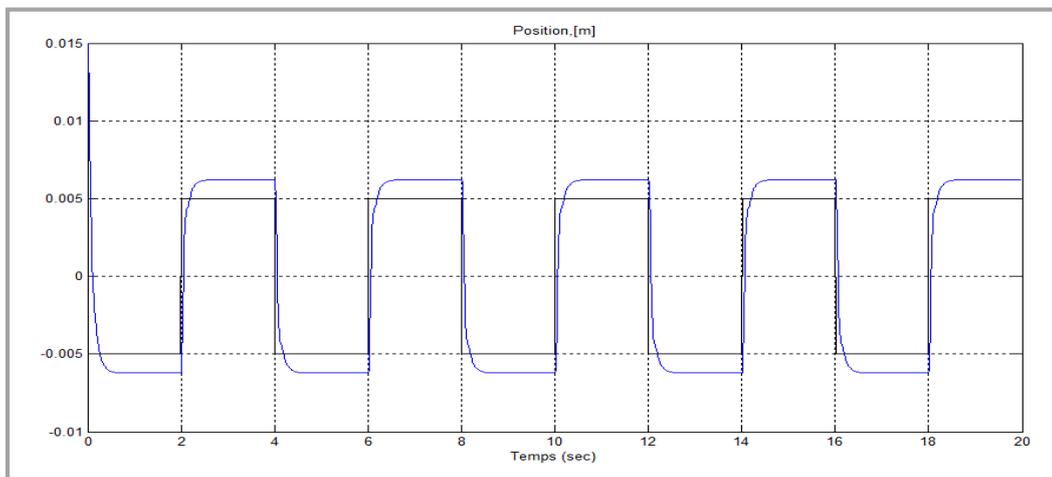


Fig.III.15 : Evolution de la position et la référence en fonction du temps.

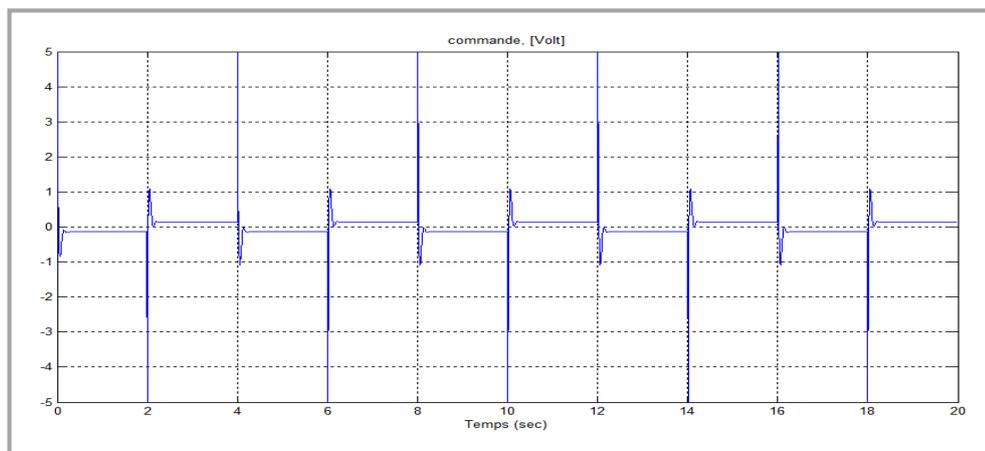


Fig.III.16: Evolution du signal de commande en fonction de temps.

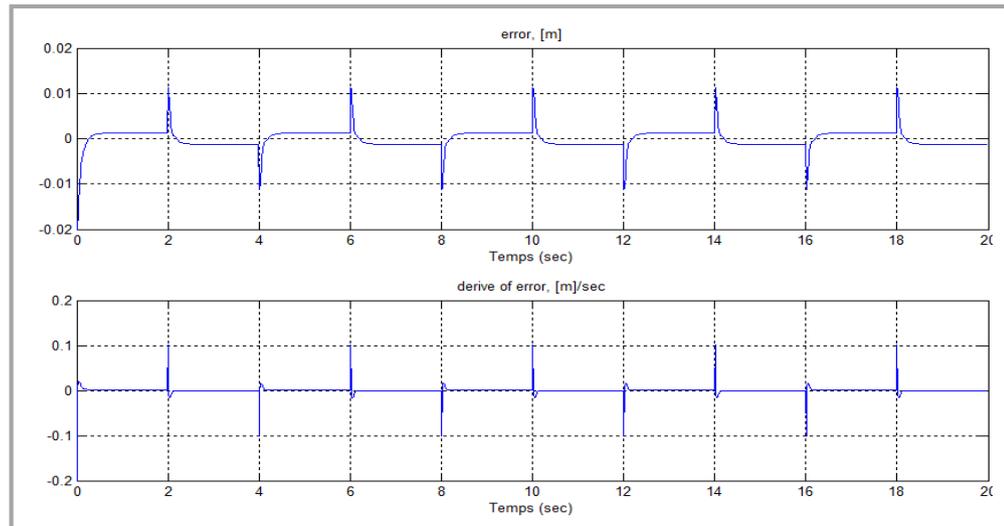


Fig. III.17: Evolution de l'erreur et sa variation.

D'après les figures précédentes, on remarque que le système converge et stabilise à la position de la balle désirée mais avec une erreur de 20 % environ. Les résultats de simulation obtenus montrent que le réglage de la position par les trois contrôleurs flous (à 3, 5 et 7 classes) conduit presque à la même réponse. Les résultats obtenus nous conduisent à choisir le contrôleur flou avec moins de règles et moins d'ensembles flous pour les entrées et la sortie, en évitant pour cela la complexité du contrôleur et en minimisant le temps de calcul. Par conclusion c'est le cas du contrôleur flou à 3 classes.

III.3 Optimisation du contrôleur flou à 3 classes par l'algorithme de chauve-souris

Dans cette partie, l'objectif est d'optimiser les paramètres des prémisses et des conclusions des règles floues d'un régulateur de type Mamdani pour la commande du système de lévitation magnétique. Pour ce faire, l'algorithme de Chauves-souris est mis en œuvre.

Le régulateur a une structure de deux entrées composées de trois fonctions d'appartenance pour l'erreur $e(t)$ et la variation de l'erreur $\Delta e(t)$, ainsi que d'une commande $u(t)$, avec une base de neuf règles.

La structure des fonctions d'appartenances des entrées de chaque contrôleur flou est représentée par la figure (III.18) ; où N, Z, P sont des ensembles flous de la variable d'entrée ; leur signification est successivement, Négative, zéro et positive.

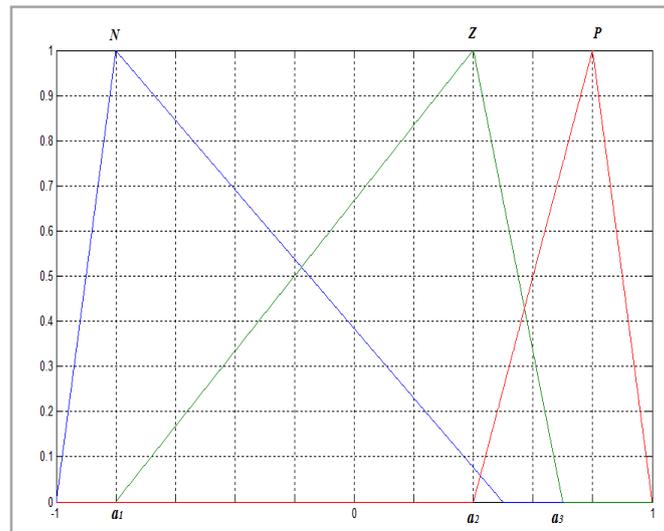


Fig.III.18 : Forme de fonction d'appartenance à optimiser.

III.3.1 Stratégie d'optimisation du contrôleur flou :

La description du processus d'optimisation de l'algorithme de Chauve-souris repose sur le schéma bloc fonctionnel illustré dans la figure (III.19). Ce dernier se compose de quatre blocs principaux:

- ✓ Bloc structurel représenté par le contrôleur flou.
- ✓ Bloc du système à commander (système de lévitation magnétique).
- ✓ Bloc d'optimisation caractérisé par l'algorithme de chauve-souris.
- ✓ Bloc décisionnel caractérisé par le critère de performances désirées.

La stratégie d'optimisation en utilisant l'algorithme de chauve-souris est réalisé selon la procédure suivante :

1. Génération d'une population initiale de solutions caractérisant les paramètres du contrôleur flou.
2. Pour toutes les solutions trouvées :
 - ✓ Calculer la fonction objective.
 - ✓ Classer les solutions obtenues selon leur fonction objective.
 - ✓ Construction d'une nouvelle population par processus de mise à jour des fréquences, des vitesses et des solutions.

L'étape 2 est répétée jusqu'à ce qu'un nombre maximum d'itérations soit effectué. Après le processus d'évolution, l'itération finale de l'algorithme consiste en la solution bien ajustée qui fournit la meilleure solution.

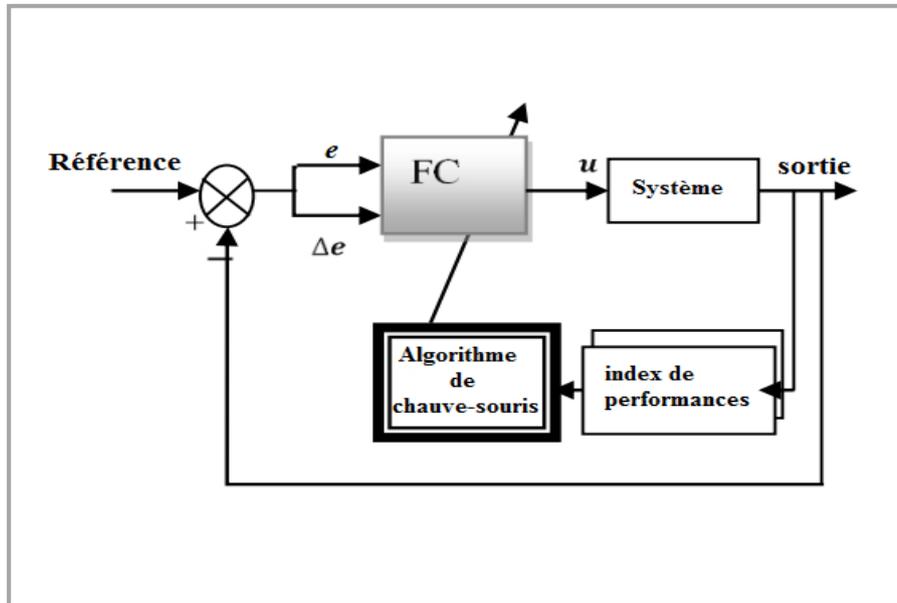


Fig.III.19 : Structure d'optimisation du contrôleur flou.

III.3.2 Application à la commande de la position de la balle magnétique

Le contrôleur flou utilisé est un contrôleur flou (CF) de type Mamdani, Les paramètres introduits dans l'algorithme de chauve-souris (9 paramètres) sont représentés comme suit :

- Les valeurs modales des fonctions d'appartenance triangulaire respectivement (a_1, a_2, a_3) pour l'erreur (e) et (b_1, b_2, b_3) pour la variation de l'erreur Δe , tout en respectant la contrainte : $(a_1 < a_2 < a_3)$ et $(b_1 < b_2 < b_3)$.
- La base des règles floues du contrôleur flou respectivement N, Z, P, sont choisies entre $[-1 \ 1]$ tout en respectant la contrainte : $(N < Z < P)$.

L'objectif global du système de commande est de réduire au minimum l'erreur à chaque instant K entre la réponse réelle du système et le point de consigne, l'indice de performance, F est choisi comme suit :

$$F = \frac{1}{nT} \sum_{k=1}^n e^2(k) \quad (\text{III. 5})$$

Où : n est le nombre total d'échantillons et T est le temps d'échantillonnage, $e(t)$ est la différence entre la valeur de sortie souhaitée et la valeur du processus de sortie mesuré sous contrôleur flou.

III.3.2.1 Résultats de la simulation

Dans cette section, nous allons présenter les résultats de simulations de la commande floue optimisée par l'algorithme de chauve-souris. L'ensemble des simulations est réalisé sous MATLAB Editor. Les valeurs des paramètres de l'algorithme BAT sont données par le tableau (III.5).

Paramètres	Désignation	Valeur
n	Taille de population	20
n_g	Nombre de générations	100
A	Intensité	0.5
r	Taux de pulsation	0.5
f_{min}	Fréquence minimale	0
f_{max}	Fréquence maximale	1

Tableau III.5 : paramètres de l'algorithme de chauve-souris.

La figure III.20 présente l'évolution de la fonction d'évaluation (coût) F pendant l'exécution de l'algorithme BAT. Cette fonction est l'erreur quadratique moyenne calculée par l'équation III.5

Après 100 itérations, On constate une décroissance de la fonction objective durant les premières itérations avant de converger vers la valeur minimale. La figure III.20 montre l'évolution de la fonction objective F . A la fin d'exécution la fonction d'évaluation est égale à $F = 3.8987 \times 10^{-6}$. L'algorithme va donner les meilleurs paramètres du contrôleur flou de la position de la balle du système de lévitation magnétique

Les tableaux III.6 et III.7 donnent les meilleurs paramètres des centres de fonctions d'appartenance de l'erreur et sa variation pour le contrôleur flou obtenus par l'algorithme BAT après 100 itérations.

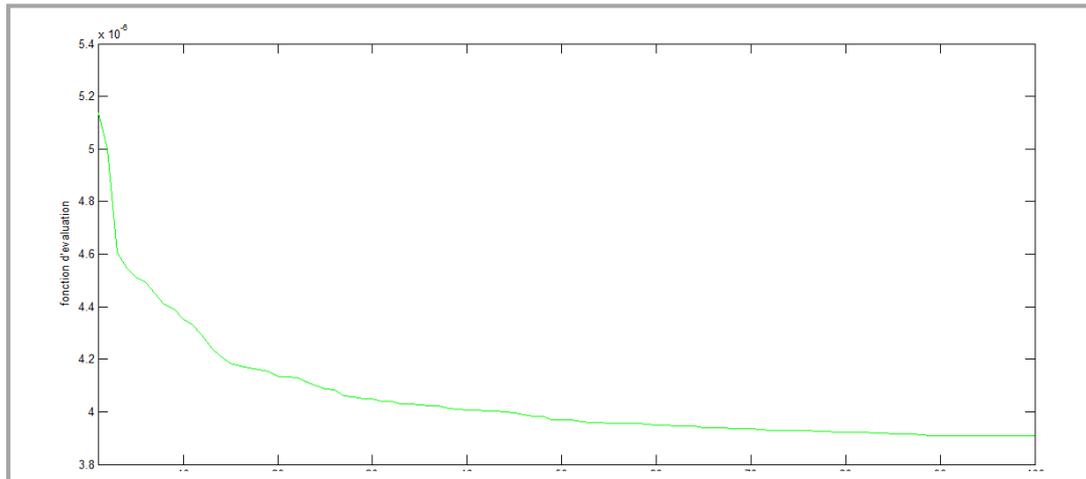


Fig.III.20: L'évolution de la fonction d'évaluation.

a	a_1	a_2	a_3
e	-0.88755	0.53829	0.99265

Tableau III.6: Paramètres optimaux des centres de fonctions d'appartenance de $e(t)$

b	b_1	b_2	b_3
Δe	-0.91747	-0.056049	0.55906

Tableau III.7: Paramètres optimaux des centres de fonctions d'appartenance de $\Delta e(t)$

La base de règles obtenues après le processus d'optimisation est représentée par le tableau III.8.

u		e		
		N	Z	P
Δe	N	0.97602	0.97602	0.074861
	Z	0.97602	0.074861	-0.88749
	P	0.074861	-0.88749	-0.88749

Tableau III.8: Paramètres optimales de la base de règles floues.

La figure (III.21) représente les formes des fonctions d'appartenance des entrées e et Δe et de la sortie du contrôleur flou obtenues par l'algorithme de chauve-souris.

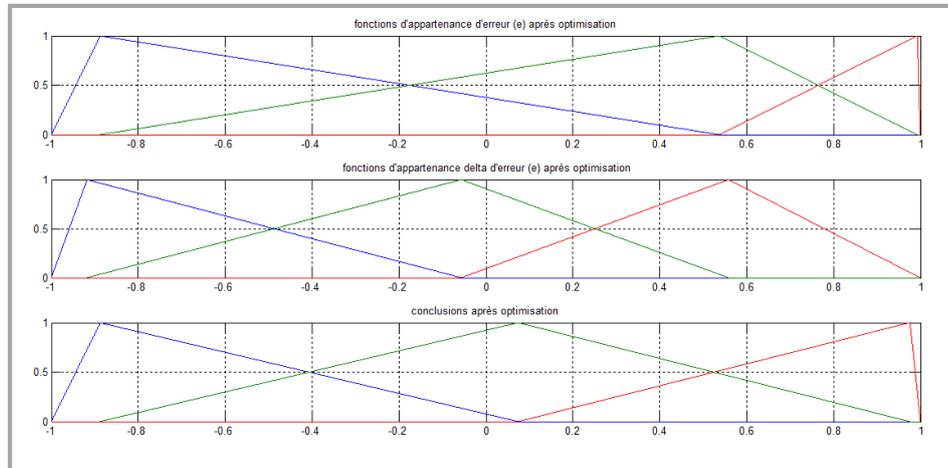


Fig.III.21 : Disposition et formes des fonctions d'appartenance des prémisses et des Conclusions du CF après optimisation.

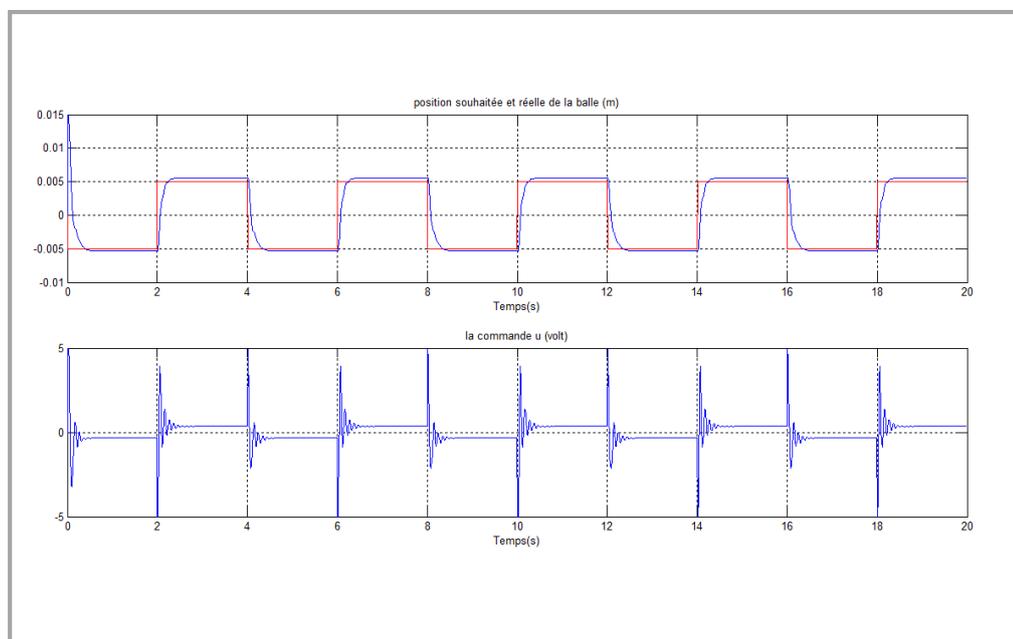


Fig.III.22 : Evolution de la position de la balle et la commande floue optimisée.

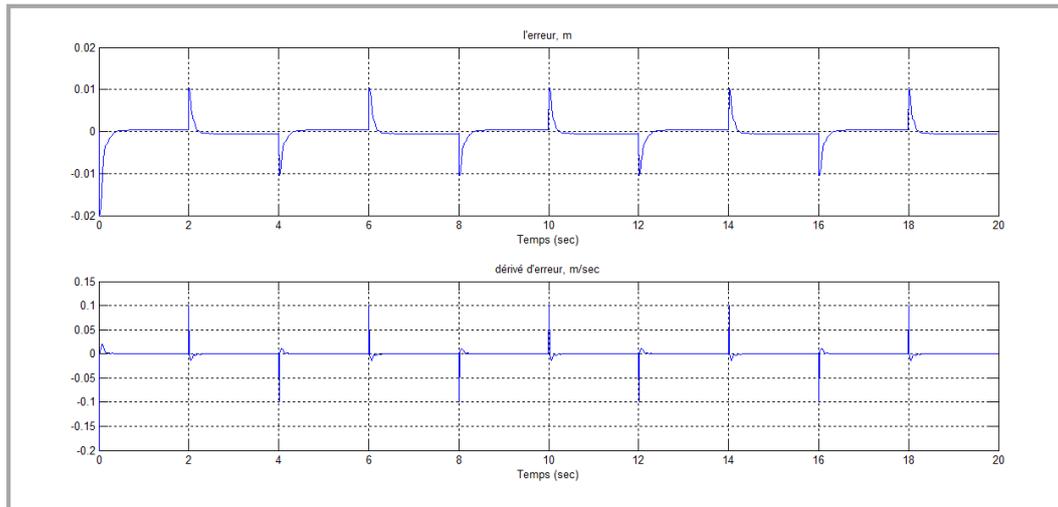


Fig.III.23 : Evolution de l'erreur et sa variation après optimisation.

III.3.2.2 Interprétation des résultats

Les Figure (III.22), (III.23) montrent une bonne convergence de la position de la balle vers sa consigne, en rassurant la stabilité et la précision du système commandé. Cela indique que les paramètres du contrôleur flou ont été ajustés convenablement par l'algorithme de chauve-souris.

III.3.2.3 Test de robustesse

La robustesse du contrôleur flou est justifiée par sa capacité de bien face aux changements des conditions initiales de fonctionnement ainsi que le changement des paramètres du système de lévitation magnétique. Pour tester la réaction du contrôleur flou aux changements des conditions initiales et de paramètres du système, nous avons procédé à une variation de 0.5% à 300% de y_0, m, R, L du système, avec y_0 est la position initiale de la balle, m est la masse de balle, R, L sont successivement la résistance et la bobine de l'enroulement.

a) Changement de de la position initiale de la balle y_0

On a changé la position initiale de la balle et on a trouvé les résultats présentés dans la figure III.24. Le système converge toujours à sa consigne avec une bonne stabilité et précision.

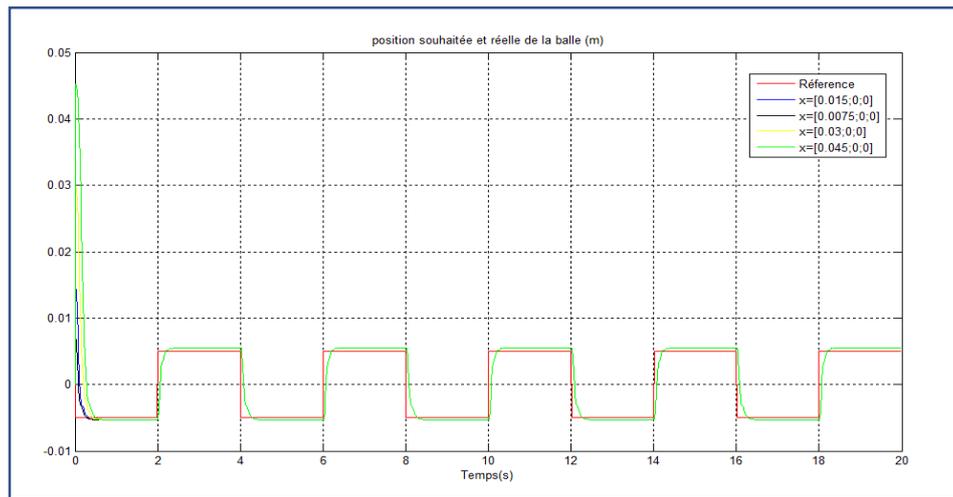


Fig. III.24 : Réponses aux changements des conditions initiales.

b) Changement de la masse de la balle m

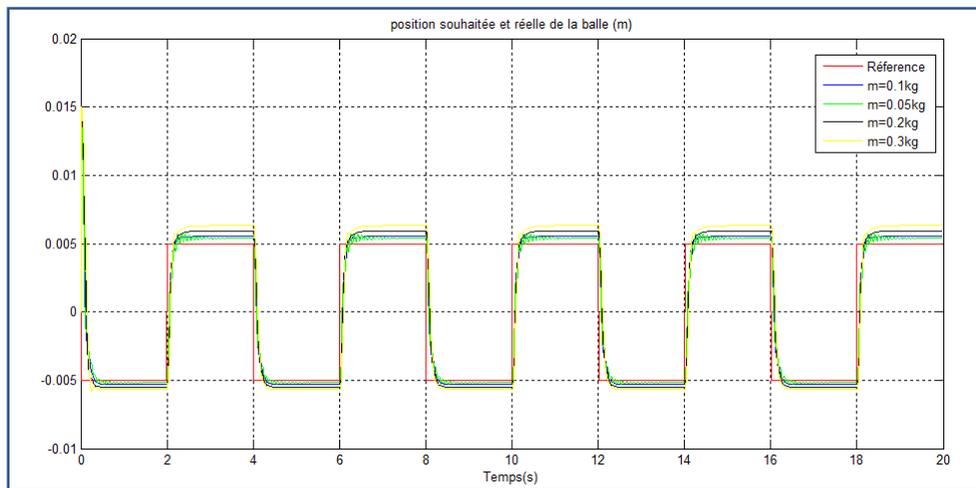


Fig. III.25 : Réponses aux changements de la masse de la balle m

c) Changement de la résistance d'enroulement R

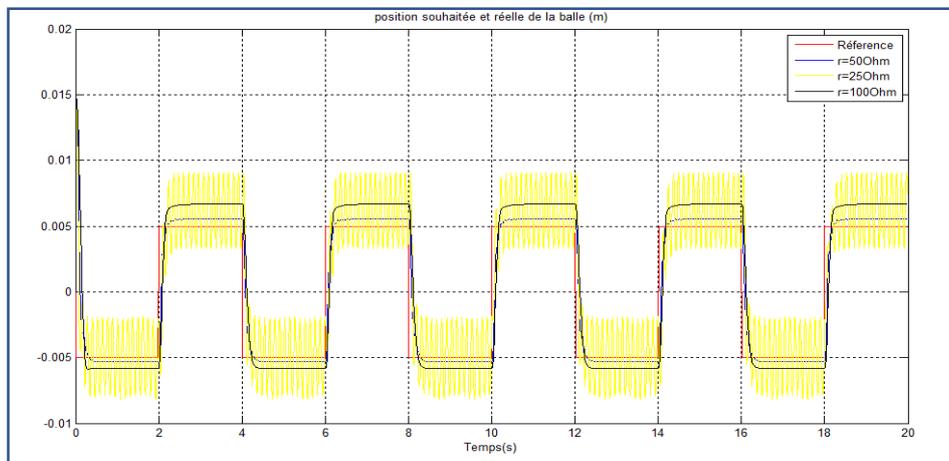


Fig. III.26 : Réponses aux changements de la résistance d'enroulement R

d) Changement de la bobine d'enroulement L

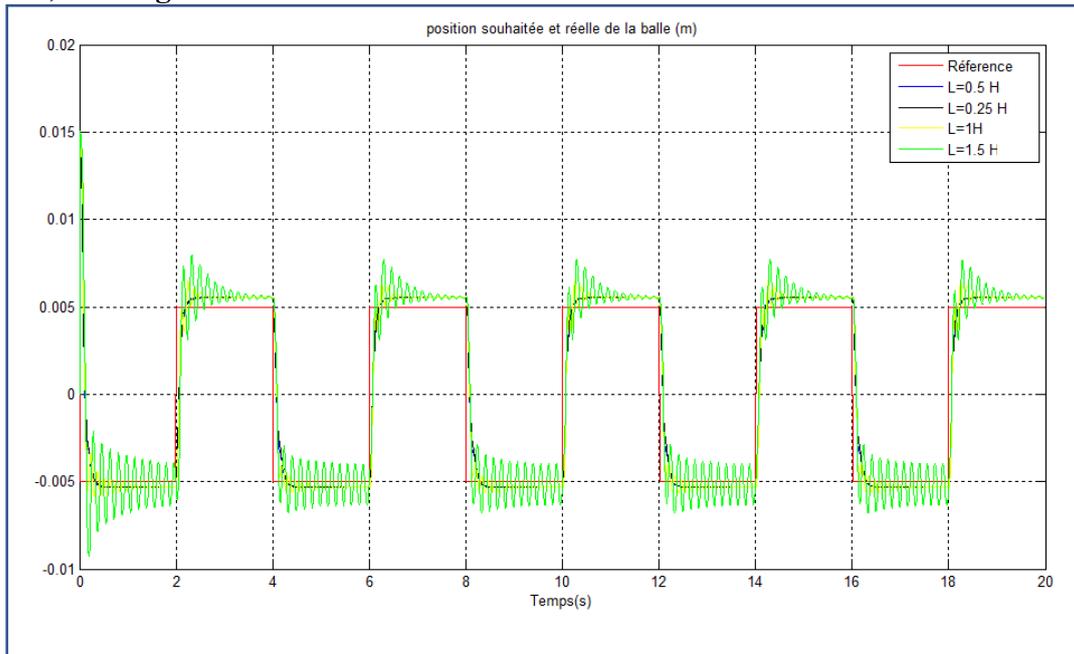


Fig. III.27 : Réponses aux changements de la bobine d'enroulement L

Les résultats présentés respectivement sur les figures (III.24), (III.25) montrent que le contrôleur flou optimisé par l'algorithme de chauve-souris est robuste car le changement de la valeur de la masse et la position initiale de la balle n'influent pas sur les résultats. Les figures III.26 et III.27 montrent qu'un petit changement de valeur de la résistance et de la bobine d'enroulement, à savoir une augmentation de 100%, donne de bon résultat coté convergence et stabilité ; mais lorsque l'augmentation atteint 200% une légère oscillation de la position de la balle par rapport à la position de référence est remarquée. En effet le changement de valeur de la résistance et de la bobine influent sur le courant $i(t)$ et par conséquent sur la position de la balle.

III.3.3 Comparaison de l'algorithme de chauve-souris et différentes méthodes d'optimisation

Dans cette partie, on va comparer l'algorithme de chauve-souris avec la recherche taboue (RT) et le recuit simulé (RS) qui ont été appliqués pour optimiser le même contrôleur flou pour le système de lévitation magnétique. Les travaux ont été faits par des étudiants de Master 2 en Electronique des systèmes embarqués de l'université de Jijel, en 2017 [14].

La figure III.28 montre la convergence de la balle pour les trois algorithmes avec une petite différence de précision.

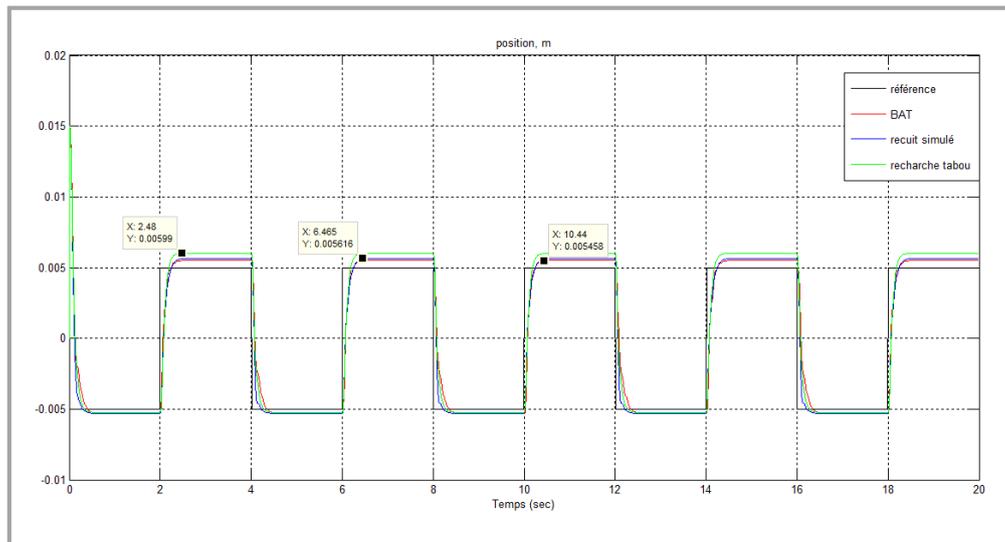


Fig.III.28: Comparaison de l’algorithme de chauve-souris et d’autres méthodes d’optimisation.

	Temps de calcul (sec)	Temps de réponse (sec)	L’erreur de position (m)	L’erreur quadratique moyenne	Complexité de l’algorithme
BAT	157.1866	0.44	0.0004	3.907×10^{-6}	Plus compliqué
RS	16.0369	0.46	0.0006	4.1894×10^{-6}	Plus simple
RT	481.3740	0.48	0.0009	4.5853×10^{-6}	Peu compliqué

Tableau III.9 : La comparaison entre l’algorithme de BAT et RS, RT.

En comparant les résultats d'optimisation de systèmes de lévitation magnétique à l'aide de différents algorithmes : l'algorithme de chauve-souris (BAT), le recuit simulé (RS) et la recherche tabou (RT), qui sont présentés sur la figure (III.28) et le tableau (III.9), on peut constater que les trois méthodes donnent de bons résultats en termes de convergence et de stabilité du système. L’algorithme de chauve-souris donne une bonne précision mais l’algorithme de recuit simulé à un bon temps de calcul. Cependant, il est important de noter que ces résultats ne permettent pas de juger quel est le meilleur algorithme d'optimisation, car

chaque méthode a ses avantages et ses inconvénients spécifiques dans le contexte du système de lévitation magnétique ou d'autres systèmes.

III.4 Conclusion

En conclusion, l'algorithme Chauves-souris est un outil efficace pour l'optimisation des contrôleurs flous de la position de la balle magnétique. En minimisant l'erreur quadratique moyenne et en ajustant les paramètres des règles floues, il permet d'améliorer les performances du système tout en simplifiant sa structure. L'utilisation de cet algorithme permet ainsi d'obtenir des résultats précis et fiables pour le contrôle de la position de la balle magnétique. Les résultats de simulation montrent que l'approche proposée présentée ici est un outil puissant pour le contrôle des systèmes non linéaires. On peut constater que l'algorithme de chauve-souris présente un outil très puissant pour la conception des contrôleurs intelligents.

Conclusion générale

Le travail présenté dans ce mémoire concerne la commande intelligente d'un système physique non linéaire mono-variable en utilisant les techniques de l'automatique telle que la logique floue et les outils d'optimisation tel que l'algorithme de chauve-souris.

Lorsqu'un modèle est partiellement inconnu ou complexe, un contrôleur flou s'avère très intéressant. Les algorithmes de réglage se basent sur des règles linguistiques de la forme SI...ALORS. L'obtention d'un contrôleur flou optimal par un réglage manuel n'étant pas systématique, nous avons donc appliqué une méthode d'optimisation des caractéristiques d'un contrôleur flou basée sur l'utilisation de l'algorithme de chauve-souris.

Le premier chapitre a permis de rappeler les généralités de la théorie de la logique floue, ainsi que de la commande floue, nous avons illustré la structure élémentaire d'un régulateur par logique floue, ainsi on analyse sa structure. Les deux types de régulateurs les plus connus sont, ceux de TAKAGI-SUGENO et de MAMDANI.

Le deuxième chapitre met l'accent sur les différentes méthodes méta heuristiques, parmi lesquelles, on a présenté l'algorithme de chauve-souris où son principe de calcul se base sur la simulation du comportement d'écholocation des chauves-souris.

Le dernier chapitre est divisé en deux parties : Dans la première partie, une commande de lévitation magnétique par logique floue a été synthétisée. Elle a permis la poursuite de trajectoire de la position de la balle du système de lévitation magnétique. Par ailleurs les résultats obtenus ont montré d'une part que l'augmentation du nombre des classes n'apporte

pas toujours d'améliorations, et d'autre part, que l'ajustement manuel des paramètres d'un régulateur flou n'est pas toujours suffisant.

Dans la deuxième partie, nous avons proposé une stratégie d'optimisation des paramètres du contrôleur flou basée sur l'utilisation de l'algorithme de chauve-souris, la procédure était d'optimiser les paramètres des prémisses des règles, ainsi que les paramètres des conclusions du régulateur flou. Après, on a fait un test de robustesse du contrôleur obtenu. Les résultats de simulation ont montré que l'algorithme de chauve-souris a amélioré les performances du régulateur flou en particulier ; la précision. A la fin du chapitre, on a fait une comparaison des résultats obtenus de ce travail et ceux obtenus par les travaux précédents sur le même système, qui ont été réalisés par les étudiants de master 2 en électronique des systèmes embarqués, en 2017 au niveau de l'université de Jijel.

On conclue que l'algorithme de chauve-souris est un outil puissant pour l'optimisation de la commande intelligente des systèmes non linéaires.

Bibliographie

- [1] L.X. Wang, “Fuzzy Systems as Nonlinear Dynamic System identifiers Part I: Design”, *Proceedings of the 31st Conference on Decision and Control*, Tucson, AZ, 1992, pp. 897-902.
- [2] I. Hayashi, H. Nomura, and N. Wakami, “Acquisition of Inference Rules by Neural Network Driven Fuzzy Reasoning”, *Japanese Journal of Fuzzy Theory and Systems*, 2(4), 1990, pp. 453-469.
- [3] L. A. Zadeh, “Fuzzy sets”, *Information and Control*, Vol. 8, 1965, pp.338-353.
- [4] F. Hoffmann, “Evolutionary algorithms for fuzzy control system design,” *Proceedings of the IEEE*, 89(9), 2001, pp.1318–1333.
- [5] S. LABED, “Méthodes bio-inspirées hybrides pour la résolution de problèmes complexes”, Thèse de doctorat en science en Informatique, université de Constantine2, 2013.
- [6] A. Bedboudi et M. Bouras “Amélioration de l'algorithme de chauve-souris par modification de règles d'évolution et introduction de mécanisme de croisement”, *revue des sciences et de technologie*, vol.36, pp1-8, 2018.
- [7] J. Godjevac, ”Idées nettes sur la logique floue ”, presses polytechniques et universitaires, romandes, 1999.
- [8] N.Tkouti, “ Optimisation Des Systèmes Photovoltaïques Connectes Au Réseau Par La Logique Floue”, Mémoire De Magister, Université de Biskra, 2004.
- [9] S.Meghriche, F.Bounaas, “Conception De La Base De Règles Floues En Utilisant l'Algorithme D'optimisation Par Essaim Particulaire ”, Mémoire de Master, Université de Jijel, 2015.
- [10] H. Buhler, ” Réglage par logique floue, Presses polytechniques et universitaires ROMANDES”, pp 32-65,1994
- [11] T.Zoheir, “Commande D'un Moteur Asynchrone Par Logique Floue”, Mémoire de Fin d'études de Master Académique, Université d'EL-Oued, 2014.
- [12] R. Messaoudi, “Modélisation Floue Et Commande PDC D'un Système Non Linéaire”, Master Académique, 2016.
- [13] H.Hezza, “Conception De Modelés Flous En Utilisant l'Algorithme d'Optimisation Par Essaim Particulaire”, Master II en Electronique, Université de Jijel, 2014.

-
- [14] H.Soufane, R .Halim, “Optimisation De La Commande Lévitacion Magnétique Par La Recherche Locale”, Mémoire de Fin d'études de Master en Electronique, Université de Jijel, 2017.
- [15] F.Chevrie, F.Guély, La logique floue, Cahier technique n° 191, p.28
- [16] N.Talbi, “Conception Des Systèmes d'Inférence Floue Par Des Approches Hybrides : Application Pour La Commande Et La Modélisation Des Systèmes Non Linéaires,” Thèse Doctorat en Science, Université de Constantine 1, 2014.
- [17] W.Boutana, N.Ykhelfoune, “ Etude Comparative En Simulation Entre Un Régulateur PID Et Un Régulateur Flou”, Mémoire de Master, Université Jijel, 2019.
- [18] M.MOUSSAOUI, “ Contrôleur Flou Testable,” Université de Sétif, 2009.
- [19] M.Fodil, “Commande Par Logique Floue Type-2-Appliquée A La Machine Asynchrone”, Mémoire de Master, 2016.
- [20] S.Naseer et al. “Fuzzy Inference System: Short Review and Design”, Vol. 6, N.4, pp.1974-6059, 2013.
- [21] S.Belhouane, “Étude et simulation d'une régulation thermique par la logique floue” Master en Electronique, Université Abderrahmane Mira de Bejaia, 2014.
- [22] S.Doudou, “Contribution à la commande moderne des systèmes non linéaires multi-variables non affines”, Thèse Doctorat Sciences en Automatique, Université Sétif 1,2013.
- [23] Lyes BELHOUL, “Résolution de problèmes d'optimisation combinatoire mono et multi-objectifs par énumération ordonnée”, pour l'obtention du grade de docteur en informatique : université Paris-Dauphine, Laboratoire d'analyse et Modélisation de systèmes pour l'aide à la Décision, paris, 2014.
- [24] A.Bedboudi, M.C.Bouras, “Amélioration de l'algorithme de chauve-souris par modification de règles d'évolution et introduction de mécanisme de croisement : Enhancing Bat Algorithm by Modifying Evolution Rules and Introducing Crossover Mechanism ”, Vol. 36, pp.1-8 ,2018.
- [25] E. GLIZE, “Méthodes exactes pour les problèmes combinatoires bi-objectifs : Application aux problèmes de tournées de véhicules, l'obtention du Doctorat, université de Toulouse Midi-Pyrénées, 2019
- [26] L.Saad, A.Leham,“ Problème de placement , application et résolution”, Mémoire de Master ,Université Tizi-Ouzou,2018
- [27] Ch.Lemize ,I.Attoum , “Optimisation du contrôleur PID pour la Commande des systèmes non linéaires par l'algorithme de chauve-souris” , Mémoire de Master Université Jijel, 2021.
- [28] H.Bouraoui, « Optimisation d'un Conùolan Flou par les Algorithmes Génétiques : Application à Commande d`m Pendule hversé », Master 1[en Electronique, Université de Jijel,2010.
- [29] X-S. Yang, “ A New Metaheuristic Bat-Inspired Algorithm”, in: Nature Inspired Cooperative Strategies for Optimization (NISCO 2010) ,Studies in Computational Intelligence, Springer Berlin, 284, Springer, pp. 65-74 ,2010.

- [30] S. Induja, V.P. Eswaramurthy, Bat Algorithm : An Overview and its Applications, International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 1, January 2016.
- [31] X.-S. Yang and X. He, “Bat algorithm : literature review and applications,” Int. J. Bio-Inspired Comput., vol. 5, no. 3, pp. 141–149, 2013.
- [32] B.MEDJAHED, D.HAIDRA, “ Implémentation d’une méthode d’optimisation des coûts dans une chaîne logistique”, Mémoire de Master, Université Mostaganem, 2014
- [33] A.kumarchoudhary, S.K.nagar, J.P.Tiwari, «implementation of Fuzzy Controller to Magnetic Levitation System », La IX control instrumentation system conference, p201-206 ,2012
- [34] K. M. Passino, S. Yurkovich, “Fuzzy control”, *Addison Wesley Longman, Inc*, 1998.