

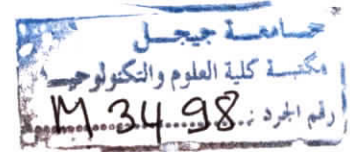
République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



M/ELE.AS

2017/05



Université Mohammed Seddik Ben Yahia-Jijel

Faculté des Sciences et de la Technologie

Département d'Electronique



Projet de fin d'Etudes pour L'obtention du Diplôme de

Master en Electronique

Option : Electronique et Analyse des Systèmes

Thème :

**Optimisation de la commande de lévitation magnétique par
la recherche locale**

Réalisé par :

HALIM Roqiya

SOUFANE Hassiba

Proposé par :

M^{me}. TALBI Nesrine

Promotion : Juin 2017

Remerciements

Remerciements

Nous remercions Dieu le tout puissant qui nous a donné la force, la volonté et le courage pour accomplir ce modeste travail.

Nous tenons à formuler notre gratitude et notre profonde reconnaissance à l'égard de notre promoteur **M^{me} : TALBI Nasrin**

Qui nous a toujours accueilli avec bienveillance qui n'a ménagé ni son temps ni ses efforts pour nous guider.

Nos remerciements aux membres de jury qui ont accepté de juger notre travail.

Nous exprimons notre profonde reconnaissance à tous les enseignants qui ont contribué de près ou de loin pour le bon déroulement de ce travail.

En fin nous remercions toute personne ayant participé de près ou de loin à l'élaboration de ce travail.

Dédicaces

Je dédie ce travail à mes chers parents pour leurs encouragements et leur soutien moral et matériel durant toutes mes années d'études que Dieu le tout puissant me les gardes.

Mon cher père *Abd El Karim* et ma chère mère *Houria*.

A mes chères sœurs : *Messaouda, Nadia, Wafa, Meriem, Zineb* et *Khaoula*.

A toute la famille *HALIM*, et surtout *Saliha, Fatima, Amar*

La famille *BOUROUISSA* et surtout mon angle *Djemal* et sa petite fille *Racha*.

A mon cher binôme *HASSIBA* et toute sa famille.

A mes chers amis : *Sarra, Hana, Zahira, Samira, Donai, Omar, Hamza, Fouzia*.

A tous mes collègues.



Roqiya.

Dédicaces

Je dédie ce travail :

A mes chères parents mon père<<Abderrezak >> et ma mère<<ZAOUANI Yamina>>

Pour leur patience, leurs amour, leurs Soutien et leur encouragement, que dieu leur garde

Pour nous.

A celui que j'aime beaucoup et qui m'a beaucoup encouragé tout au long de ce Projet :

A mon fiancé <<Halim>> et à tout sa famille.

*A mes chers frères: **Khaled, Ahmed, hamza, Abdou, Atef, et Youssef.***

A ma sœur <<Houda>>.

A tout mes oncles, tantes, cousins et cousines.

*A toute les familles : **SOUFANE, ZAOUANI et ROINA.***

A tous mes Amis et collègue

<< Nawara, Widad, Hanane, Zahira, Safia, Somia, Noura, Sabah, Sara >>

A mon cher binôme <<Roqiya>> et tout sa famille.

A tout l'étudiant de l'électronique de l'université de Jijel promotion 2017.

Hassiba.

Sommaire

SOMMAIRE

Remerciements.....	I
Dédicaces.....	II
Sommaire.....	IV
Liste des figures.....	VIII
Liste des tableaux.....	X
Introduction générale.....	1

Chapitre I

La commande floue

I.1 Introduction.....	4
I.2 Théorie des sous-ensembles flous.....	5
I.2.1 Univers du discours-sous-ensemble flou-degré d'appartenance.....	6
I.2.2 Les règles linguistiques.....	7
I.2.3 Support-Point de commutation-Singleton flou.....	7
I.2.4 Fonction d'appartenance.....	9
I.2.5 Les opérateurs de la logique floue.....	10
I.3 Commande floue.....	11
I.4 Structure générale d'un contrôleur flou.....	12
I.4.1 Mise en forme des entrées, Normalisation.....	14
I.4.2 Fuzzification.....	14
I.4.3 Règles floues.....	16
I.4.4 Inférences floues.....	17
I.4.5 Défuzzification.....	17
I.4.5.1 Défuzzification par centre de gravité.....	18
I.4.5.2 Défuzzification par centre maximum.....	18
I.4.5.3 Défuzzification par valeur maximale.....	18
I.4.6 Dénormalisation.....	19
I.5 Les différents types de la commande à base de la logique floue.....	19

Sommaire

I.5.1 La commande floue de type Mamdani	19
I.5.2 La commande floue du type PI, PD où PID	20
I.5.3 La commande neuro-floue.....	21
I.5.4 La commande floue adaptative.....	22
I.5.5 La commande floue de type T-S	23
I.6 Caractéristiques, avantages et inconvénients de la logique floue	24
I.6.1 Caractéristiques	24
I.6.2 Avantages	24
I.6.3 Inconvénients.....	24
I.7 Domaines d'application	24
I.8 Conclusion.....	25

Chapitre II

L'optimisation par les méthodes de la recherche locale

II.1 Introduction.....	27
II.2 Définition	29
II.3 Classification	29
II.4 Méta-heuristique	31
II.5 Recherche locale	31
II.5.1 Définition	32
II.5.2 Principe	32
II.5.3 Voisinage.....	32
II.6 Recuit simulé	33
II.6.1 Historique	33
II.6.2 Définition	33
II.6.3 Principe de base.....	34
II.6.4 Notions	34
II.6.4.1 Probabilité de Boltzmann	34

Sommaire

II.6.4.2 Critère de Metropolis.....	35
II.6.4.3 Chaîne de Markov et équilibre thermique	36
II.6.5 Algorithme	36
II.6.6 Paramètres	37
II.6.6.1 Nombre d'itérations à température constante.....	38
II.6.6.2 Critères d'arrêt.....	38
II.6.7 Avantages et Inconvénients du RS.....	38
II.6.7.1 Avantages	38
II.6.7.2 Inconvénients.....	38
II.7 Recherche taboue	39
II.7.1 Historique.....	39
II.7.2 Définition	39
II.7.3 Principe de base.....	39
II.7.4 Eléments de la RT	40
II.7.4.a Stratégie d'exploration régressive	40
II.7.4.b Liste Tabou.....	40
II.7.4.c Critère d'aspiration	40
II.7.4.d Critère d'intensification.....	40
II.7.4.e Critère de diversification	40
II.7.5 Critère d'arrêt.....	41
II.7.6 Algorithme	41
II.7.7 Avantages et Inconvénients.....	42
II.7.7.1 Avantages	42
II.7.7.2 Inconvénients.....	42
II.8 conclusion	42

Sommaire

Chapitre III

Simulation & Interprétations

III.1 Introduction	43
III.2 Le modèle mathématique de lévitation magnétique	44
III.3 la commande floue du système de lévitation magnétique	46
III.3.1 Le type de contrôleur utilisé	46
III.3.2 Structure de base du contrôleur flou	47
III.3.3 Résultats de la simulation	49
III.4 Optimisation du contrôleur flou par la recherche taboue et le recuit simulé :	50
III.4.1 Stratégie d'optimisation du contrôleur flou :	50
III.4.2 Optimisation du contrôleur flou par la méthode de recherche taboue.....	51
III.4.2.1 Application à la commande de la position de la balle magnétique	51
III.4.3 Optimisation du contrôleur flou par la méthode de recuit simulé :	58
III.4.3.1 Organigramme générale du recuit simulé	58
III.5 Conclusion	64
Conclusion générale.....	65
Bibliographie.....	66

LISTE DES FIGURES

Figure I.1 : Caractéristiques d'un sous ensemble flou A.....	8
Figure I.2 : formes usuelles des fonctions d'appartenance.....	10
Figure I.3 : Structure de base d'un contrôleur flou.....	13
Figure I.4 : Univers du discours partitionné par les termes linguistique.....	15
Figure I.5 : Désignations standard et fonctions d'appartenance.....	16
Figure I.6 : Structure générale d'une commande neuro-floue.....	22
Figure II.1. Classification des méthodes d'optimisation.....	29
Figure II.2 : Fonctionnement de l'algorithme de recuit simulé.....	36
Figure II.3 : L'organigramme général de l'algorithme de la recherche taboue.....	41
Figure III.1 : structure du procédé.....	44
Figure III.2 : Structure de commander la position de la balle magnétique.....	47
Figure III.3 : forme des fonctions d'appartenances.....	48
Figure III.4 : Evolution de la position et la référence en fonction du temps.....	49
Figure III.5 : Evolution du signal de commande en fonction de temps.....	49
Figure III.6 : Evolution de l'erreur.....	49
Figure III.7: Schéma bloc fonctionnel du processus d'optimisation par la recherche taboue.....	51
Figure III.8 : forme des fonctions d'appartenances à optimiser.....	51
Figure III.9 Evolution de la fonction d'évaluation.....	52
Figure III.10 Disposition et formes des fonctions d'appartenance des prémisses des conclusions après optimisation.....	53
Figure III.11 Evolution de position, commande et d'erreur.....	54
Figure III.12 : changement de valeur de masse de la balle.....	55
Figure III.13 : changement de valeur de la résistance.....	55

Liste des Figures

Figure III.14 : changement de valeur de la bobine.....	56
Figure III.15 : changement de la position initiale de la balle.....	56
Figure III.16 Organigramme général du recuit simulé utilisé pour l'optimisation.....	57
Figure III.17 : Evolution de la fonction d'évaluation.....	58
Figure III.18 Disposition et formes des fonctions d'appartenance des prémisses des conclusions après optimisation.....	59
Figure III.19 Evolution de la position, la commande et l'erreur en fonction du temps.....	60
Figure III.20 : changement de valeur de masse de la balle.....	61
Figure III.21: changement de valeur de la résistance.....	61
Figure III.22 : changement de valeur de la bobine.....	62
Figure III.23 : changement de conditions initiales.....	62

LISTE DES TABLEAUX

Tableau I.1 : Exemple de variable linguistique.....	15
Tableau III.1 : Paramètres du système.	45
Tableau III.2 : Base de règles du contrôleur flou utilisé.	48
Tableau III.3 Paramètres des prémisses optimaux obtenus par la recherche taboue.....	53
Tableau III.4 paramètres des conclusions optimaux obtenus par la recherche taboue.....	53
Tableau III.5 Paramètres des prémisses optimaux obtenus par le recuit simulé	59
Tableau III.6 paramètres des conclusions optimaux obtenus par le recuit simulé.....	59



Introduction

Générale

Introduction Générale

La représentation d'un procédé par un modèle mathématique, primordial pour la synthèse de lois de commande, est caractérisée par des équations différentielles linéaires ou non linéaires reliant les entrées, les états et les sorties. Ces équations différentielles sont parfois très difficiles à trouver, notamment pour les procédés non linéaires complexes qui font apparaître beaucoup de paramètres (moteur thermique, procédé chimique,.....). Une solution à ce problème peut être envisagée à partir d'une représentation du comportement du procédé non linéaire par un modèle flou où les relations entre les entrées, les sorties et les états sont exprimées par des règles floues [1].

L'avancement des recherches dans le domaine du flou a prouvé la capacité et la puissance des modèles flous dans l'identification (apprentissage) floue des procédés non linéaires [2].

Plusieurs chercheurs utilisent cette puissance d'apprentissage pour représenter la dynamique des procédés non linéaires. La problématique de l'identification et de la commande floue est basée sur la propriété d'approximation universelle des systèmes flous. En effet, ceux-ci sont capables d'approximer, avec un degré de précision arbitraire fixé, n'importe quelle dynamique non linéaire sur un ensemble compact.

Le choix aléatoire de la structure de la fonction d'appartenance des ensembles flous et de la base de règles conduit souvent à des mauvais résultats surtout lorsqu'on ne connaît pas le comportement du système, c'est pour cette raison qu'on doit utiliser une méthode d'optimisation des contrôleurs flous afin d'obtenir une meilleure stabilité et une bonne réponse pour le système contrôlé [2].

Introduction Générale

La recherche opérationnelle s'attache à étudier des problèmes d'optimisation combinatoire dont la résolution constitue un véritable challenge. Il s'agit de trouver une affectation de valeurs à un certain nombre de variables tout en respectant un ensemble de contraintes donné. Parmi les paradigmes de résolution, la programmation par contraintes est particulièrement adaptée pour étudier la réalisabilité d'un problème de satisfaction de contraintes, tandis que la programmation linéaire en nombres entiers s'inscrit davantage dans le cadre de la recherche d'un extremum d'une fonction linéaire. Cependant, ces approches partagent une procédure de résolution commune qui consiste en l'énumération implicite de l'ensemble des solutions du problème [3].

Il s'agit alors de parcourir l'espace de recherche et d'en extraire une solution admissible (problèmes de satisfaction de contraintes) ou optimale (problèmes d'optimisation combinatoire) ou de prouver qu'il n'en existe pas. Le schéma classique consiste en une recherche arborescente qui évalue à chaque nœud la solution partielle courante et l'étend si possible en affectant une valeur à une variable non encore instanciée [3].

Les problèmes combinatoires difficiles ont depuis longtemps attiré l'attention des chercheurs. On peut citer Garey et Johnson qui ont approfondi les bases des concepts de problèmes NP-difficiles et ont montré que de nombreux problèmes n'avaient que peu de possibilités d'être résolus efficacement par des méthodes exactes. Ces méthodes exactes permettent d'obtenir une ou plusieurs solutions dont l'optimalité est garantie. Cependant, dans certaines situations, il est nécessaire de disposer d'une solution de bonne qualité (c'est-à-dire assez proche de l'optimale) dans un contexte de ressources (temps de calcul et/ou mémoire) limitées. Dans ce cas, l'optimalité de la solution n'est pas garantie, ni même l'écart avec la valeur optimale. Néanmoins, le temps nécessaire pour obtenir cette solution est beaucoup plus faible que dans le cas d'une méthode exacte. Ce type de méthodes, dites heuristiques, est particulièrement utile pour les problèmes nécessitant une solution en temps limité ou pour résoudre des problèmes difficiles. Ces méthodes approchées peuvent se classer en différentes catégories [3]:

- Constructives (algorithmes gloutons),
- Recherche locale (algorithmes de descente, recherche à grand voisinage,...),
- Méta-heuristiques (recuit simulé, recherche Tabou,.....),

Introduction Générale

- Evolutionnaires (algorithmes génétiques, algorithmes d'optimisation par colonies de fourmis, algorithmes mimétiques,...)

Les méta-heuristiques les plus classiques sont celles fondées sur la notion de parcours. Dans cette optique, l'algorithme fait évoluer une seule solution sur l'espace de recherche à chaque itération. La notion de voisinage est alors primordiale.

L'objectif de notre travail est de l'optimisation d'un contrôleur flou par un algorithme de recherche tabou et un algorithme du recuit simulé. Donc ce mémoire est organisé comme suit :

Dans le premier chapitre, nous décrivons les principes de base de la logique floue. Nous introduisons la théorie des éléments de base de la logique floue ainsi que les diverses opérations nécessaires à la commande floue. Ensuite, nous donnons la structure d'un contrôleur flou, ainsi leurs caractéristiques. Puis nous présentons les deux types de contrôleurs flous les plus utilisés (Mamdani et Takagi-sugeno), et nous terminons le chapitre par les avantages et les inconvénients de la logique floue.

Dans le deuxième chapitre, nous introduisons l'origine des méta-heuristiques, leurs notions de base, la notion de voisinage, ensuite nous décrivons l'algorithme du recuit simulé et l'algorithme de recherche tabou en détails.

Le dernier chapitre, est consacré dans un premier temps, à la mise au point d'un régulateur flou de type Takagi-Sugeno pour commander la position de la balle du système de lévitation magnétique, la partie suivante traite l'optimisation des paramètres du contrôleur flou par les algorithmes du recuit simulé et de la recherche tabou. Nous nous intéressons aux valeurs modales (sommets) des fonctions d'appartenance des entrées des contrôleurs flous ainsi aux conclusions des règles, l'optimisation consistera donc à minimiser un critère d'erreur quadratique moyenne entre la consigne et la sortie du système. Ce chapitre illustre les réponses du système de lévitation magnétique ainsi des tests de robustesses vis-à-vis aux changements de leurs paramètres et conditions initiales du système désirée.

Nous terminons par une conclusion générale sur l'ensemble de cette étude et nous proposons des perspectives de ce travail.



Chapitre I

La Commande Floue

Chapitre I

La Commande Floue

I.1 Introduction

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

La logique floue (en anglais « *fuzzy logic* ») est un axe de recherche important sur lequel se focalisent de nombreux scientifiques. La logique floue a connu un intérêt important dans la communauté scientifique au cours des 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 en 1990. La logique floue constituant une manière simple de gérer les capteurs intégrés dans ces nouveaux produits [36].

C'est entre 1965 et 1970 que Lotfi ZADEH étendit explicitement le principe des sous-ensembles Classiques à la logique multi-valuée ; en effet, il a publié un article sur l'algorithme flou, en remarquant qu'un être humain sait garer plus facilement une voiture, qu'un ordinateur manipulant les équations de la physique. En 1973, il a formalisé ce problème sous un ensemble de règles du type « si...Alors... ».

La théorie de la logique floue permet la représentation des connaissances données sous formes de variables linguistiques sachant que les valeurs de ces variables sont des mots ou des expressions du langage naturel.

L'équipe de MAMDANI en 1974 a établi le premier rapprochement entre la logique floue et le domaine du contrôle par la conception du premier contrôleur flou [5].

Entre 1975 et 1980, il y eu une intense activité de recherche en commande floue des processus en Europe, particulièrement au Danemark, la compagnie SMIDTH a utilisé cette stratégie pour la commande d'un four à ciment, il fonctionne depuis 1979.

Entre 1985 et 1990, les japonais se lancent dans ce domaine, plus précisément SUGENO publie en 1985 une application sur le pilotage d'une voiture en modèle réduit.

A partir de 1988 de nombreuses applications ont été développées, dont on cite : TV (Sony, Canon), Robots industriels, voitures (Subaru), fusées spatiales (NASA) [30].

I.2 Théorie des sous-ensembles flous

Dans cette section, nous décrivons rapidement les fondements mathématiques de la théorie des sous-ensembles flous. Dans la théorie ensembliste classique, l'appartenance d'un élément à un sous-ensemble est définie par une valeur logique standard : 1 si l'élément appartient au sous-ensemble, 0 sinon. Dans la théorie 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 [12].

I.2.1 Univers du discours-Sous-ensemble flou-Degré d'appartenance

Définition 1.1 : (sous-ensemble)

Soit X un ensemble de référence et soit x un élément quelconque de X . Un sous-ensemble flou A de X est défini comme l'ensemble des couples :

$$A = \{(x, \mu_A(x)), x \in X\} \quad (I.1)$$

Avec :

$$\mu_A: X \rightarrow [0,1] \quad (I.2)$$

Ainsi, un sous-ensemble flou A de X est caractérisé par une fonction d'appartenance $\mu_A(x)$ qui associe, à chaque point x de X un réel dans l'intervalle $[0,1]$; $\mu_A(x)$ représente le degré d'appartenance de x à A .

En attribuant les éléments x de X un **degré d'appartenance**, on remarque les trois cas suivants :

$$\begin{cases} \mu_A(x) = 0 \\ 0 < \mu_A(x) < 1 \\ \mu_A(x) = 1 \end{cases} \quad (I.3)$$

Où, $\mu_A(x) = 0$ si x n'appartient pas à A ; $0 < \mu_A(x) < 1$ si x appartient partiellement à A ; et $\mu_A(x) = 1$ si x appartient entièrement à A . La fonction d'appartenance $\mu_A(x)$ inclut ou exclut donc à ses extrémités, tout élément x au sous-ensemble A , mais entre les valeurs extrêmes le degré d'appartenance varie à proportion de la proximité à l'ensemble [4].

Définition 1.2 : (univers du discours)

L'ensemble de référence d'un mot du langage naturel s'appelle l'univers du discours. L'univers du discours d'un mot est un ensemble de termes qui évoquent le même concept mais à degrés différents. Il peut être fini ou non.

Définition 1.3 : (Variable linguistique)

Une variable linguistique représente un état dans le système à régler. Chaque variable linguistique est caractérisée par un ensemble tel que :

$$\{X, T(x), U, G, M\} \quad (I.4)$$

Où : x : est le nom de la variable,

$T(x)$: est l'ensemble des valeurs linguistiques que peut prendre x ,

U : est l'univers du discours associé avec la valeur de base,

G : est la règle syntaxique pour générer les valeurs linguistiques de x ,

M : est la règle sémantique pour associer un sens à chaque valeur linguistique.

1.2.2 Les règles linguistiques

L'idée principale des systèmes basés sur la logique floue, est d'exprimer la connaissance humaine sous la forme de 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 de conditions est satisfait) **alors** (un ensemble de conséquences peut être exécuté).

Exemple [2] :

Si température élevée **et** pression forte **ALORS** ventilation forte **et** soupape grande ouverte

1.2.3 Support-Point de commutation-Singleton flou

Définition 2.1 : (Support)

Le support de A est l'ensemble des éléments de X appartenant au moins un peu à A . Autrement dit, c'est l'ensemble :

$$S(A) = \{x \in X | \mu_A(x) > 0\} \quad (I.5)$$

En particulier, 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** » [31].

Définition 2.2 : (hauteur)

C'est la valeur maximale de la fonction d'appartenance, généralement, égale à 1. Elle est définie par [6]:

$$h(A) = \max_{x \in X} \{\mu_A(x)\} \quad (I.6)$$

Définition 2.3: (noyau)

Le noyau est défini par :

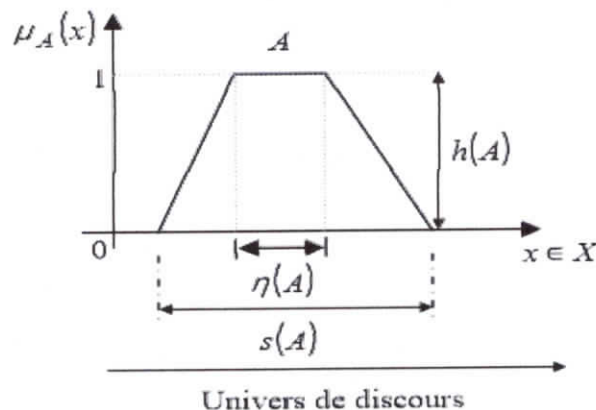
$$\eta(A) = \{x \in X | \mu_A(x) = 1\} \quad (I.7)$$

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.

Définition 2.4 :(type)

Le types peuvent être triangulaire, gaussienne, trapézoïdale, sigmoïdale ou de type pic 'singleton', ...etc. La figure I.1 représente les caractéristiques d'une fonction trapézoïdale [6].



Figure(I.1) : Caractéristiques d'un sous ensemble flou A

I.2.4 Fonction d'appartenance

Une partie floue A de B est caractérisée par une application de B dans [0,1]. Cette application, appelée « fonction d'appartenance » et notée μ_A , représente le degré de validité de la proposition « x appartient à A » pour chacun des éléments x de B.

Le plus souvent, la fonction d'appartenance est déterminée par l'une des fonctions suivantes [32], [33], [34], [35] :

➤ Fonction triangulaire

Elle est définie par trois paramètres {a, b, c}, qui déterminent les coordonnées des trois sommets (Figure I.2-a).

$$\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-d}\right), 0\right) \quad (I.8)$$

➤ Fonction trapézoïdale

Elle est définie par quatre paramètres {a, b, c, d} (Figure I.2-b) [4].

$$\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right) \quad (I.9)$$

➤ Fonction gaussienne

Elle est définie par deux paramètres { σ , m} (Figure I.2-c).

$$\mu(x) = \exp\left(-\frac{(x-m)^2}{2\delta^2}\right) \quad (I.10)$$

➤ Fonction sigmoïde

Elle est définie par deux paramètres {a, c} (Figure I.2-d).

$$\mu(x) = \frac{1}{1 + \exp(-a(x-c))} \quad (I.11)$$

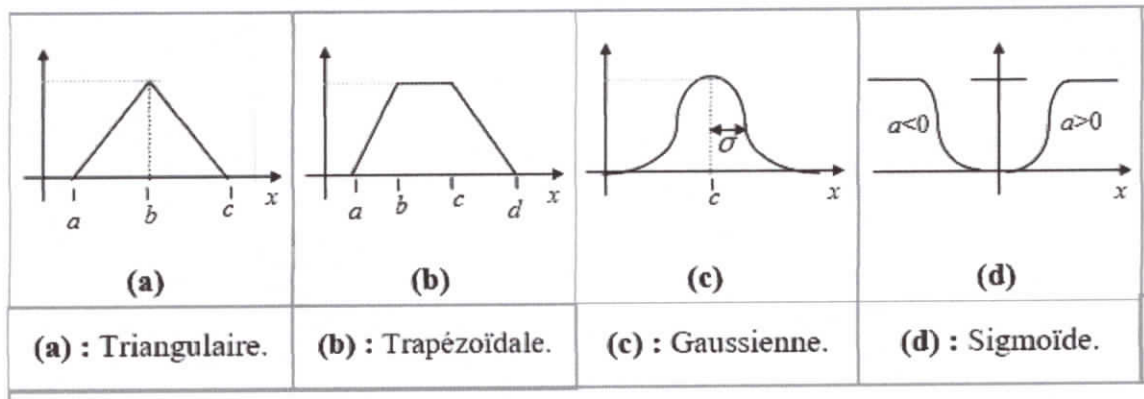


Figure (I.2) : formes usuelles des fonctions d'appartenance

I.2.5 Les opérateurs de la logique floue

Étant donné que le concept de sous-ensemble flou peut être vu comme une généralisation du concept d'ensemble classique, on est conduit à introduire des opérations sur les sous-ensembles flous qui sont équivalentes aux opérations classiques de la théorie des ensembles lorsqu'on a affaire à des fonctions d'appartenance à valeurs 0 ou 1. On présente ici, les opérations les plus couramment utilisées.

➤ Égalité

Deux sous-ensembles flous A et B de X sont égaux, si leurs fonctions d'appartenance prennent la même valeur pour tous l'élément x de X. Formellement $A = B$ si et seulement si :

$$\forall x \in X, \mu_A(x) = \mu_B(x) \quad (I.12)$$

➤ Complément

Le complémentaire d'un sous-ensemble flou A de X noté \bar{A} est défini par :

$$\forall x \in X, \mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (I.13)$$

Contrairement aux sous-ensembles classiques, la propriété de non contradiction n'est pas satisfaite ici ($A \cap \bar{A} \neq \emptyset$). De même que la propriété du tiers exclus ($A \cup \bar{A} \neq X$). Par contre, les autres propriétés sont conservées, notamment : $\bar{\bar{A}} = A$; $X = \emptyset$; $\bar{X} = \emptyset$; $|A| + |\bar{A}| = |X|$ si X est fini.

➤ **Inclusion [4]**

Soit A et B deux sous-ensembles flous de X. Si pour n'importe quel élément x de X, x appartient toujours moins à A qu'à B, alors on dit qu'A est inclus dans B ($A \subseteq B$). Formellement, $A \subseteq B$ si et seulement si :

$$\forall x \in X, \mu_A(x) \leq \mu_B(x) \quad (I.14)$$

➤ **Union [4]**

L'union de deux sous-ensembles flous A et B de X est le sous-ensemble flou constitué des éléments de X affectés du plus grand des degrés avec lesquels ils appartiennent à A et B. Formellement, $A \cup B$ est donné par :

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) \quad (I.15)$$

➤ **Intersection [4]**

L'intersection de deux sous-ensembles flous A et B de X est le sous-ensemble flou constitué des éléments de X affectés du plus petit des degrés avec lesquels ils appartiennent à A et B. Formellement, $A \cap B$ est donné par :

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad (I.16)$$

I.3 Commande floue

De façon générale, un système de commande a pour objectif de piloter l'entrée d'un processus afin d'obtenir un fonctionnement correcte de ce dernier. Lorsqu'on dispose d'un modèle plus ou moins précis du système à commander, on peut utiliser un contrôleur de structure standard, fixe ou adaptatif, dont les paramètres seront évalués à partir du modèle. Malheureusement, lorsque le système est difficilement modélisable, la conception du contrôleur peut s'avérer très complexe, sinon impossible.

Lorsqu'un opérateur humain commande manuellement un système, les actions qu'il réalise sont dictées par une connaissance subjective du fonctionnement de ce système. Par exemple, s'il fait « froid » dans une pièce, on « augmente » le chauffage ; s'il fait « très froid », on « chauffe plus ».

Cette commande du système peut être envisagée de façon différente selon la personne qui la réalise : la sensation de « froid » n'est pas directement liée à une mesure de la température[38].

Ce principe est à la base de la commande floue. La mesure réalisée sur le système (« température ») est prise en compte par l'intermédiaire d'une variable linguistique (« froid », « tiède », « chaud »), qui est issue d'une analyse par un expert humain. Ensuite, l'action à réaliser est déduite à la fois d'un ensemble de règles de commande (« s'il fait froid, on chauffe plus ») et de l'état du système, qualifié par la variable linguistique. Enfin, la commande finale du système est créée en utilisant les conclusions de la déduction.

En résumé, un contrôleur flou comporte les différents éléments suivants :

- Un sous-système composé en général d'un ensemble de variables linguistiques.
- Une base de connaissances : « base de données » et « base de règles linguistiques de commande ».
- Un sous-système réalisant un raisonnement en utilisant des méthodes issues de la *logique floue* [38].

I.4 Structure générale d'un contrôleur flou

On peut distinguer plusieurs étapes dans le traitement des règles. Un schéma représentatif peut-être le suivant : Les variables caractéristiques du système à commander et les consignes définissent les variables d'entrée du contrôleur flou. Les variables caractéristiques sont en général, les grandeurs de sortie du processus et, le cas échéant, d'autres mesures déterminantes pour saisir l'évolution dynamique du processus. Les variables de sortie du contrôleur flou sont les commandes à appliquer au processus.

La base de connaissances est composée d'une base de données et d'une base de règles. La base de données regroupe :

- Les ensembles flous associés aux variables d'entrée et de sortie du contrôleur flou,
- Les facteurs d'échelle (gains) en entrée (normalisation) et en sortie (dénormalisation).

La figure suivante représente la structure générale d'un contrôleur flou :

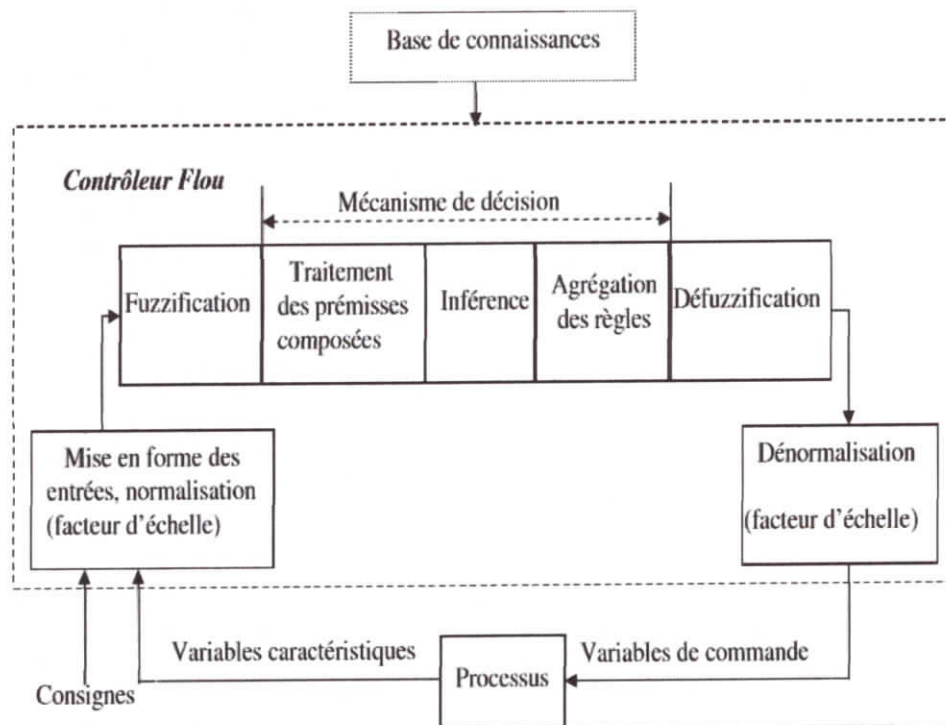


Figure (I.3) : Structure de base d'un contrôleur flou.

La base de règle contient des règles de la forme :

« Si X_1 est A_1 et X_2 est A_2 et X_n est A_n Alors Y est B »

Où X_1, X_2, \dots, X_n et Y sont des grandeurs physiques caractéristiques du système et du problème de commande. A_1, A_2, \dots, A_n et B sont des labels linguistiques. Suivant la nature de B on parlera de :

- Règles à conclusion symbolique (contrôleur de type Mamdani) : B est une valeur linguistique. *Exemple* :

Si l'erreur est « Négatif Moyen » et la variation de l'erreur est « Positif Petit » Alors la commande est « Négatif Petit ».

- Règles à conclusion algébrique (contrôleur de Takagi-Sugeno) : B est une valeur numérique (singleton) ou une équation mathématique bien précise (non floue). *Exemple* :

Si l'erreur est « Négatif Moyen » et la variation de l'erreur est « Positif Petit » Alors la commande est -0.3.

Lorsque B est une valeur numérique on parle de règles de Takagi-Sugeno « d'ordre zéro », sinon, de règles à conclusion polynomiale.

Bien que les sorties des contrôleurs flous de type Takagi-Sugeno soient généralement des fonctions non linéaires statiques de leurs entrées, il ne faut pas oublier de mentionner les contrôleurs dits « flous dynamiques » de Takagi-Sugeno, ou B est un modèle dynamique, certain ou incertain, à temps continu ou discret. L'utilisation de tels contrôleurs permet d'étendre certains résultats de l'automatique classique à la commande floue.

On distingue classiquement trois parties dans la structure d'un contrôleur flou : la fuzzification, le mécanisme de décision et la défuzzification :

- La fuzzification est l'étape qui permet de transformer une grandeur mesurée sur le processus en un ensemble flou.
- Le mécanisme de décision permet de calculer les ensembles flous associés à la commande.
- La défuzzification est l'étape qui permet de transformer l'ensemble flou, obtenu par le calcul précédent, en une grandeur de commande à appliquer au processus. Les opérations de normalisation et de dénormalisation sont des étapes optionnelles.

I.4.1 Mise en forme des entrées, Normalisation

Cette première étape permet le traitement des variables d'entrée du contrôleur flou, par exemple, calcul de l'erreur et des variations d'erreurs. L'utilisation de domaines normalisés (univers de discours compris entre -1 et 1) nécessite une transformation d'échelle transformant les grandeurs physiques des entrées en des valeurs normalisées appartenant à l'intervalle $[-1,1]$ [7].

I.4.2 Fuzzification

La fuzzification est réalisée dans l'interface d'entrée du contrôleur flou. Durant cette phase, les informations issues du système sont tout d'abord normalisées. Ensuite, les données normalisées sont transformées en qualifications linguistiques, en utilisant des règles sémantiques définies par un expert. Durant la phase de normalisation, chaque mesure issue du système est modifiée pour fournir une valeur appartenant à un univers du discours relativement simple [39]. On peut choisir comme univers de discours un intervalle centré sur zéro : $[-c, +c]$.

Si la mesure initiale x est comprise dans un autre intervalle $[a, b]$, la normalisation est souvent réalisée par transformation linéaire, selon :

$$Y = \frac{2c}{b-c} \left[x - \frac{a+b}{2} \right] \quad (I.17)$$

L'univers du discours est ensuite représenté par une variable linguistique, qui comporte un nombre assez restreint de termes (en général trois, cinq ou sept) de façon à limiter le nombre de règles. Enfin, les valeurs normalisées déduites de chacune des entrées sont transformées en qualifications linguistiques, en utilisant les variables linguistiques correspondantes.

Exemple :

Une variable linguistique sur l'univers de discours $[-6, +6]$, peut être définie de la façon précisée dans le tableau I.1 :

Terme linguistique	Signification	Règle sémantique
NG	Négatif grand	Environ -6
NM	Négatif moyen	Environ -4
NP	Négatif petit	Environ -2
EZ	Environ Zéro	Environ 0
PP	Positif petit	Environ +2
PM	Positif moyen	Environ +4
PG	Positif grand	Environ +6

Tableau I.1 : Exemple de variable linguistique

La figure suivante montre les fonctions d'appartenance associées aux différents termes linguistiques.

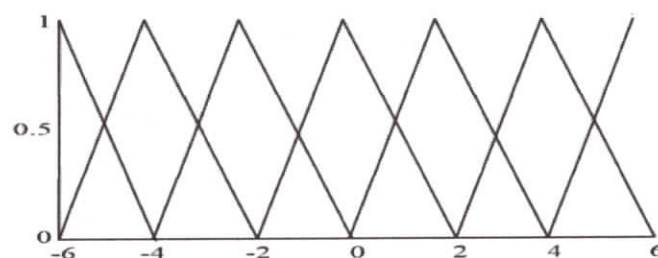


Figure (I.4): Univers du discours partitionné par les termes linguistique.

Dans cet exemple, les termes linguistiques utilisés sont ceux initialement proposés par Mamdani lorsque la variable linguistique en comporte sept. Les fonctions d'appartenance correspondant à des variables comportant trois et cinq termes sont représentées sur la figure I.5.

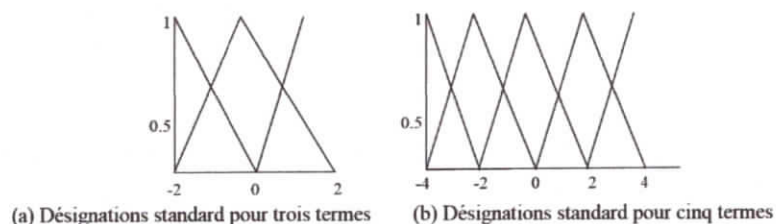


Figure (I.5) : Désignations standard et fonctions d'appartenance.

La fuzzification est une étape clé dans tout contrôleur flou. La variable linguistique, qui est définie par une expertise, doit respecter un certain nombre de critères afin d'être efficace [39] :

- Chaque terme linguistique est un nombre flou, de noyau non nul et dont la fonction d'appartenance est convexe.
- Les fonctions d'appartenance doivent respecter l'ordre linguistique.
- Les fonctions d'appartenance ne doivent pas trop se superposer. On tolère en général un chevauchement qui ne dépasse pas la mi-hauteur des termes linguistiques consécutifs.

I.4.3 Règles floues

Les règles floues permettent de déduire des connaissances concernant l'état du système en fonction des qualifications linguistiques fournies par l'étape de fuzzification. Ces connaissances sont également des qualifications linguistiques.

Habituellement, les règles floues sont déduites des expériences acquises par les opérateurs ou les experts. Ces connaissances sont traduites en règles simples pouvant être utilisées dans un processus d'inférence floue. Par exemple, si un expert exprime la règle «si la température de l'eau est chaude, il faut ajouter de l'eau froide », le système utilisera une règle du genre «si p alors q ».

Aujourd'hui, il est cependant possible de constituer une base de règles floues grâce à des méthodes d'apprentissage, sans avoir nécessairement besoin d'un expert humain [39].

I.4.4 Inférences floues

Elle repose sur l'utilisation d'un opérateur d'implication permettant d'évaluer le degré de vérité d'une règle R de la forme « Si X est A et 2 X Alors Y est B ».

En d'autres termes, cet opérateur quantifie la force de la liaison entre la prémisse et la conclusion de la règle.

Les opérateurs les plus courants en commande sont de type conjonctif :

- L'implication de Mamdani (1974) :

$$\mu_R(X, Y) = \min (\mu_A(x), \mu_B(x)) \quad (I.18)$$

- L'implication de Larsen (1980) [7] :

$$\mu_R(X, Y) = \mu_A(x) * \mu_B(x) \quad (I.19)$$

Le moteur d'inférence réalise le traitement numérique des règles d'inférence, décrites par des opérateurs flous, pour obtenir la sortie linguistique ou floue du régulateur. Cette opération est faite par différentes méthodes, on cite principalement :

a) La méthode d'inférence max-min : L'opérateur **ET** est réalisé par la formation du minimum, l'opérateur **OU** est réalisé par la formation du maximum, et l'implication (**ALORS**) est réalisée par la formation du minimum.

b) La méthode d'inférence max-produit : L'opérateur **ET** est réalisé par la formation du produit, l'opérateur **OU** est réalisé par la formation du maximum, et l'implication (**ALORS**) est réalisée par la formation du produit.

c) La méthode d'inférence somme-prod : On réalise au niveau de la condition, l'opérateur **OU** par la formation de la somme (valeur moyenne), et l'opérateur **ET** par la formation du produit. Pour la conclusion, l'opérateur **ALORS** est réalisé par un produit.

I.4.5 Défuzzification

Les méthodes d'inférence fournissent un résultat qui est une fonction d'appartenance. Or, la sortie du contrôleur est en général une grandeur continue, prenant sa valeur dans un intervalle. La défuzzification est le traitement qui permet de définir une correspondance entre le résultat de l'inférence et la grandeur continue fournie en sortie.

1.4.5.1 Défuzzification par centre de gravité

La défuzzification par centre de gravité consiste à calculer l'abscisse du centre de gravité de la fonction d'appartenance selon [39]:

$$y_{cg} = \frac{\int y \cdot \mu_{B_{res}}(y) dy}{\int y \cdot B_{res}(y) dy} \quad (I.20)$$

En pratique, on estime le centre de gravité en calculant la moyenne d'un certain nombre de points échantillonnés sur la fonction :

$$y_{cg} = \frac{\sum y_i \cdot \mu_{B_{res}}(y_i)}{\sum \mu \cdot B_{res}(y_i)} \quad (I.21)$$

Le temps nécessaire au traitement est directement proportionnel au nombre de points retenus pour le calcul de la moyenne. Selon les contraintes fixées par l'application, il y a un compromis à réaliser entre la précision souhaitée et le temps de calcul disponible.

1.4.5.2 Défuzzification par centre maximum

Dans cette méthode, la valeur de sortie est estimée par l'abscisse du point correspondant au centre de l'intervalle pour lequel la fonction d'appartenance est maximale. Cette valeur est fournie par l'expression [39]:

$$y_{cm} = \frac{\inf M + \sup M}{2} \quad (I.22)$$

Dans laquelle M est l'ensemble des points pour lesquels la fonction d'appartenance est maximale :

$$M = \{y \in [-c, +c] | \mu \cdot B_{res}(y) = H(B_{res})\} \quad (I.23)$$

Dans le cas discret, on explore en fait la liste de tous les points pour lesquels la fonction d'appartenance est maximale afin de trouver le plus petit et le plus grand.

1.4.5.3 Défuzzification par valeur maximale

Cette méthode ne s'utilise que dans le cas discret. On choisit comme sortie y_m l'abscisse de la valeur maximale de la fonction d'appartenance résultante.

On prend la moyenne des abscisses du maximum

$$y_m = \frac{\sum_{y_i \in M} y_i}{|M|} \quad (I.24)$$

Où M est défini dans l'équation (I.23) [39].

I.4.6 Dénormalisation

Cette dernière étape transforme les valeurs normalisées des variables de commande en des valeurs appartenant à leur domaine physique respectif.

I.5 Les différents types de la commande à base de la logique floue

En se basant sur les différentes règles de commande floue et de leurs méthodes de génération, les approches de commande en utilisant la logique floue peuvent être classées dans les catégories suivantes :

I.5.1 La commande floue de type Mamdani

Dans un système flou de Mamdani (Conventionnel) les règles sont de type :

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

Où A_i, B_j et C_k représentent des termes linguistiques auxquels est attribuée une signification floue que l'on suppose normalisée.

Pour des entrées numériques précises x_0, y_0 un sous-ensemble flou de sortie, noté F , est généré selon la relation :

$$\mu_F(Z) = \max_{(i,j,k) \in I} \cdot \min(\mu_{A_i}(x_0), \mu_{B_j}(y_0), \mu_{C_k}(z)) \quad (I.25)$$

μ est appelée fonction d'appartenance.

L'opération $\max_{(i,j,k) \in I}$ correspond à l'application de l'opérateur \max sur l'ensemble des règles actives. La valeur numérique de sortie délivrée par le système flou est obtenue par défuzzification, classiquement implantée par la méthode du centre de gravité :

$$Z = \frac{\int Z \mu_F(Z) dz}{\int \mu_F(Z) dz} \quad (I.26)$$

Dans ce type de systèmes flous, la prémisse et la conclusion sont floues. Après l'inférence, le résultat est un ensemble flou caractérisé par sa fonction d'appartenance. Afin d'obtenir la valeur réelle de la commande à appliquer il faut passer par une étape de « défuzzification »

Les règles de la commande floue « Si-Alors » sont obtenues à partir de l'expertise d'un opérateur. Quoique la performance d'une telle technique de commande soit généralement satisfaisante en pratique le non garanti de la stabilité du système de commande en boucle fermée est souvent critiquée.

Plusieurs approches ont été proposées pour étudier le problème de stabilité. L'idée principale de ces approches est de considérer le contrôleur flou comme un contrôleur non linéaire et d'utiliser la théorie des systèmes non linéaire pour l'analyse de la stabilité.

I.5.2 La commande floue du type PI, PD ou PID

La principale difficulté des régulateurs PID réside dans le réglage des gains afin d'obtenir de bonnes performances surtout lorsque le procédé à commander est d'ordre élevé et à dynamique non linéaire. En effet, le problème dans le réglage des gains devient plus difficile en présence de variations paramétriques et de perturbations externes.

La nature complexe des procédés industriels et la nécessité de maintenir de bonnes performances exigent un réglage en ligne des gains du régulateur PID. Cette opération peut être réalisée par des régulateurs numériques autos ajustables. Dans ce cas, l'ajustement en ligne des paramètres de synthèse est, généralement, basé sur un algorithme de réglage qui fait appel, à chaque période, à la fonction de transfert du procédé, souvent difficile à déterminer avec une précision suffisante. Ceci rend ce principe de contrôle assez sensible aux variations du comportement de procédé industriel.

Dans l'objectif de maintenir de bonnes performances face à d'éventuelles variations de la dynamique du procédé, il est évident que le réglage ne peut se faire aisément qu'avec la mesure de l'écart entre les performances réelles et désirées.

L'exploitation d'une supervision basée sur la logique floue en vue de l'adaptation des paramètres de synthèse d'un régulateur du type PID permet d'offrir une alternative très intéressante pour ramener, quelle que soit la situation en ligne, les performances réelles à celles souhaitées. Il est possible d'améliorer les performances du système de commande par la combinaison de la logique floue avec un régulateur PID ce qui s'appelle un contrôleur PID flou.

I.5.3 La commande neuro-floue

La commande neuronale et la commande floue sont deux techniques de la commande intelligente les plus populaires. Elles possèdent plusieurs similarités, par exemple, les systèmes flous et les réseaux de neurones ont le même objectif, tous les deux tentent de modéliser le fonctionnement du cerveau humain. Les réseaux de neurones tentent de modéliser l'architecture du cerveau en créant une modélisation de l'entité du cerveau : le neurone. Les systèmes flous eux modélisent le cerveau par son mode de fonctionnement (apprentissage et déduction).

Dans les deux cas le principe est de stocker la connaissance et l'employer pour prendre des décisions (lois) de commande, et la combinaison de ces deux techniques appelée commande neuro-floue est fondamentalement une commande floue augmentée par les réseaux de neurone et permet d'améliorer les caractéristiques comme la flexibilité, la possibilité de traitement de données et l'adaptabilité.

Le processus du raisonnement flou est réalisé par les réseaux neurones dont les poids correspondent aux paramètres du raisonnement flou. En utilisant la méthode de rétro propagation ou n'importe quel algorithme d'apprentissage de réseaux de neurone, un système de commande neuro-floue peut identifier les règles de commande et apprendre les fonctions d'appartenances floues.

Il convient de noter que le modèle flou de type Takagi–Sugeno est l'un des systèmes flous les plus employés pour réaliser la commande neuro-floue. Un des avantages principaux de la commande neuro-floue est qu'elle n'exige pas des informations sur le modèle mathématique du système à commander. Cette classe de commande floue offre une nouvelle voie dans la résolution de plusieurs problèmes de commande où le modèle mathématique du système pourrait être non disponible. Cependant, une de ses limitations principales est l'analyse systématique de la stabilité des systèmes de commande en boucle fermée et la convergence des algorithmes d'adaptation.

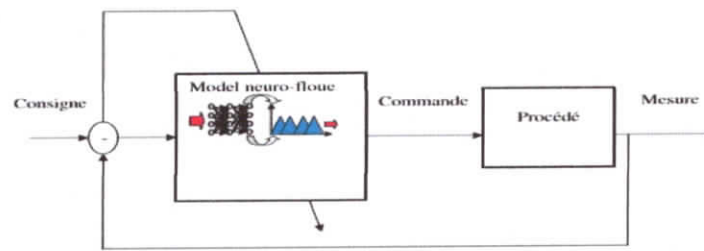


Figure (I.6) : Structure générale d'une commande neuro-floue

1.5.4 La commande floue adaptative

La commande adaptative est basée sur l'utilisation d'un mécanisme d'adaptation pour la commande des systèmes partiellement connus. La commande adaptative des systèmes linéaires et certaines classes spéciales des systèmes non linéaires a été bien développée depuis les années 70. Alors que la commande adaptative des systèmes non linéaires présente toujours un défi à la communauté de commande. Les modèles mathématiques ne pourraient pas être disponibles pour beaucoup des systèmes complexes en pratique, et le problème de la commande adaptative de ces systèmes est loin d'être résolu d'une manière satisfaisante.

Des systèmes flous basés sur les fonctions floues de base sont employés pour représenter les fonctions non linéaires inconnues des systèmes qui possèdent des variations intrinsèques inconnues. L'objectif est de conserver une bonne performance du système complet en adaptant le régulateur en fonction des variations du système. Les paramètres des systèmes flous tels que les fonctions d'appartenance caractérisées par des limites linguistiques sont mises à jour selon des lois d'adaptation.

L'idée principale de la commande floue adaptative est d'employer les systèmes flous pour rapprocher les comportements non linéaires inconnus des systèmes non linéaires et d'appliquer les techniques bien développées de la commande adaptative.

Les régulateurs flous adaptatifs peuvent être divisés en deux classes :

- Les régulateurs adaptatifs directs : les paramètres des régulateurs sont modifiés en ligne afin de minimiser l'erreur entre le modèle de référence et le système réel.
- Les régulateurs adaptatifs indirects : les paramètres du système réel sont estimés et le régulateur est choisi en supposant que les valeurs estimées du modèle sont les vraies va-

leurs du système réel. Si le modèle du procédé réel utilise un système à base de logique floue on parle alors de régulateur flou adaptatif indirect.

La commande floue adaptative se compose donc d'un régulateur flou adaptatif. La différence principale entre un régulateur flou adaptatif et un régulateur adaptatif conventionnel est que le premier peut utiliser des données linguistiques. Ce qui est très important lorsque le système possède des incertitudes qu'un opérateur humain a appris à anticiper. Cependant, des approches robustes doivent être adoptées pour la commande floue adaptative à cause des erreurs inhérentes d'approximation des systèmes flous approchant les fonctions non linéaires originales.

1.5.5 La commande floue de type T-S

Les contrôleurs flous de Takagi–Sugeno sont, comme ceux de Mamdani, construits à partir d'une base de règles "Si ... Alors...". Les prémisses sont toujours exprimées linguistiquement et donc similaires à celles utilisées dans un contrôleur flou de Mamdani alors que les conclusions sont de nature numérique.

Un système flou de type Takagi-Sugeno (T-S) utilise des règles écrites de la manière suivante :

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

Où f_k représente une fonction réelle quelconque.

Les fonctions de sortie f_k peuvent être en principe des fonctions arbitraires des entrées, mais d'une manière générale elles sont choisies telles qu'elles soient une combinaison linéaire des entrées.

La particularité d'un modèle T-S est que la logique floue est seulement utilisée dans la partie prémisses des règles. La partie conclusion est décrite par des valeurs numériques.

Pour les valeurs d'entrée précises x_0 et y_0 , la sortie z est évaluée selon le mécanisme suivant :

$$z = \frac{\sum_{(i,j,k) \in I} w_{i,j} f_k(x_0, y_0)}{\sum_{(i,j,k) \in I} w_{i,j}} \quad (I.27)$$

$$\text{Avec : } w_{i,j} = \mu_{A_i}(x_0) \cdot \mu_{B_j}(y_0) \quad (I.28)$$

Ce type de modèle est aussi très intéressant pour la représentation de systèmes non linéaires tels que les systèmes mécaniques ou chaotiques [7].

I.6 Caractéristiques, avantages inconvénients de la logique floue

I.6.1 Caractéristiques

- La connaissance est structurée en règles linguistiques.
- Les traitements sont numériques.

I.6.2 Avantages

- La théorie est simple et s'applique à des systèmes complexes.
- Pas de modèles mathématiques requis du procédé à asservir.
- Robustesse de la commande floue vis-à-vis des incertitudes.
- Possibilités de commande auto-adaptative aux variations du procédé [8].

I.6.3 Inconvénients

- Technique de réglage essentiellement empirique.
- Performances dépendent de l'expertise.
- Il n'existe pas de théorie générale qui caractérise rigoureusement la stabilité, la robustesse (difficultés de certification dans le transport, espace...).

I.7 Domaines d'application

- Appareils électroménagers (lave-linges, aspirateurs, autocuiseurs,etc).
- Systèmes audio-visuels (appareils de photos, caméscope à stabilisateur d'images, photocopieurs, ...).
- Systèmes automobiles embarqués (BVA, ABS, suspension, climatisation, ...ect).
- Systèmes autonomes mobiles.
- Systèmes de commande dans la plupart des domaines industriels de production, transformation, traitement de produit et déchet.
- Systèmes de transport (train, métro, ascenseur...).
- Systèmes de décision, diagnostic, renaissance [29].

I.8 Conclusion

La logique floue est une branche des mathématiques qui permet de faire le lien entre modélisation numérique et modélisation symbolique, ce qui a permis des développements industriels spectaculaires à partir d'algorithmes très simples de traduction de connaissances symboliques en entité numérique et inversement. C'est-à-dire la représentation des connaissances données sous formes de variables linguistiques sachant que les valeurs de ces variables sont des mots ou des expressions du langage naturel. La théorie des ensembles flous a également donné naissance à un traitement original de l'incertitude, fondée sur l'idée d'ordre, et qui permet de formaliser le traitement de l'ignorance partielle et de l'inconsistance dans les systèmes d'informations avancées.

Elle a le grand avantage de constituer une représentation mathématique de labels linguistiques largement utilisés dans l'expression de connaissances expertes, qualitatives et manipulées dans le raisonnement approché qui sera fait à partir de cette connaissance.



Chapitre II

L'optimisation par les méthodes de la recherche locale

Chapitre II

L'optimisation par les méthodes de la recherche locale

II.1 Introduction

L'optimisation combinatoire occupe une place très importante en recherche opérationnelle, en mathématiques discrètes et en informatique. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation [9] et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation combinatoire [40]. Bien que les problèmes d'optimisation combinatoire soient souvent faciles à définir, ils sont généralement difficiles à résoudre.

Etant donné l'importance de ces problèmes, de nombreuses méthodes de résolution ont été développées en recherche opérationnelle (RO) et en intelligence artificielle (IA). Ces méthodes peuvent être classées sommairement en deux grandes catégories : les méthodes exactes (complètes) qui garantissent la complétude de la résolution et les méthodes approchées (incomplètes) qui perdent la complétude pour gagner en efficacité.

Le principe essentiel d'une méthode exacte consiste généralement à énumérer, souvent de manière implicite, l'ensemble des solutions de l'espace de recherche. Pour améliorer l'énumération des solutions, une telle méthode dispose de techniques pour détecter le plus tôt possible les échecs (calculs de bornes) et d'heuristiques spécifiques pour orienter les différents choix. Parmi les méthodes exactes, on trouve la plupart des méthodes traditionnelles (développées depuis une trentaine d'années) telles les techniques de séparation et évaluation progressive (SEP) ou les algorithmes avec retour arrière. Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable.

Malgré les progrès réalisés (Notamment en matière de la programmation linéaire en nombres entiers), comme le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème, les méthodes exactes rencontrent généralement des difficultés face aux applications de taille importante.

Les méthodes approchées constituent une alternative très intéressante pour traiter les problèmes d'optimisation de grande taille si l'optimalité n'est pas primordiale. En effet, ces méthodes sont utilisées depuis longtemps par de nombreux praticiens. On peut citer les méthodes gloutonnes et l'amélioration itérative : par exemple, la méthode de Lin et Kernighan qui resta longtemps le champion des algorithmes pour le problème du voyageur de commerce [41].

Depuis une trentaine d'années, des progrès importants ont été réalisés avec l'apparition d'une nouvelle génération de méthodes approchées puissantes et générales, souvent appelées méta heuristiques [42,43]. Une méta-heuristique est constituée d'un ensemble de concepts fondamentaux (par exemple, la liste tabou et les mécanismes d'intensification et de diversification pour le méta heuristique Recherche Tabou), qui permettent d'aider à la conception de méthodes heuristiques pour un problème d'optimisation. Ainsi les méta-heuristiques sont adaptables et applicables à une large classe de problèmes.

Les méta-heuristiques sont représentées essentiellement par les méthodes de voisinage comme le recuit simulé et la recherche Tabou, et les algorithmes évolutifs comme les algorithmes génétiques et les stratégies d'évolution. Grâce à ces méta-heuristiques, on peut proposer aujourd'hui des solutions approchées pour des problèmes d'optimisation classiques de plus grande taille et pour de très nombreuses applications qu'il était impossible de traiter auparavant [44,45]. On constate, depuis ces dernières années, que l'intérêt porté aux méta-heuristiques augmente continuellement en recherche opérationnelle et en intelligence artificielle.

Ce chapitre passe en revue les méthodes de résolution (issues le plus souvent des communautés de la Recherche Opérationnelle et de l'Intelligence Artificielle) proposées pour traiter les problèmes d'optimisation combinatoire émergeant dans le cadre du notre projet.

II.2 Définition

L'optimisation combinatoire est un outil indispensable combinant diverses techniques de la mathématique discrète et de l'informatique afin de résoudre des problèmes d'optimisation combinatoire de la vie réelle.

Un problème d'optimisation combinatoire consiste à trouver la meilleure solution dans un ensemble discret de solutions appelé ensemble des solutions réalisables. En général, cet ensemble est fini mais de cardinalité très grande.

Il s'agit, en général, de maximiser (problème de maximisation) ou de minimiser (problème de minimisation) une fonction d'évaluation sous certaines contraintes. Le but est de trouver une solution optimale dans un temps d'exécution raisonnable [10].

II.3 Classification

Les méthodes d'optimisation peuvent être réparties en deux catégories :

- Les méthodes exactes
- Les méthodes approchées

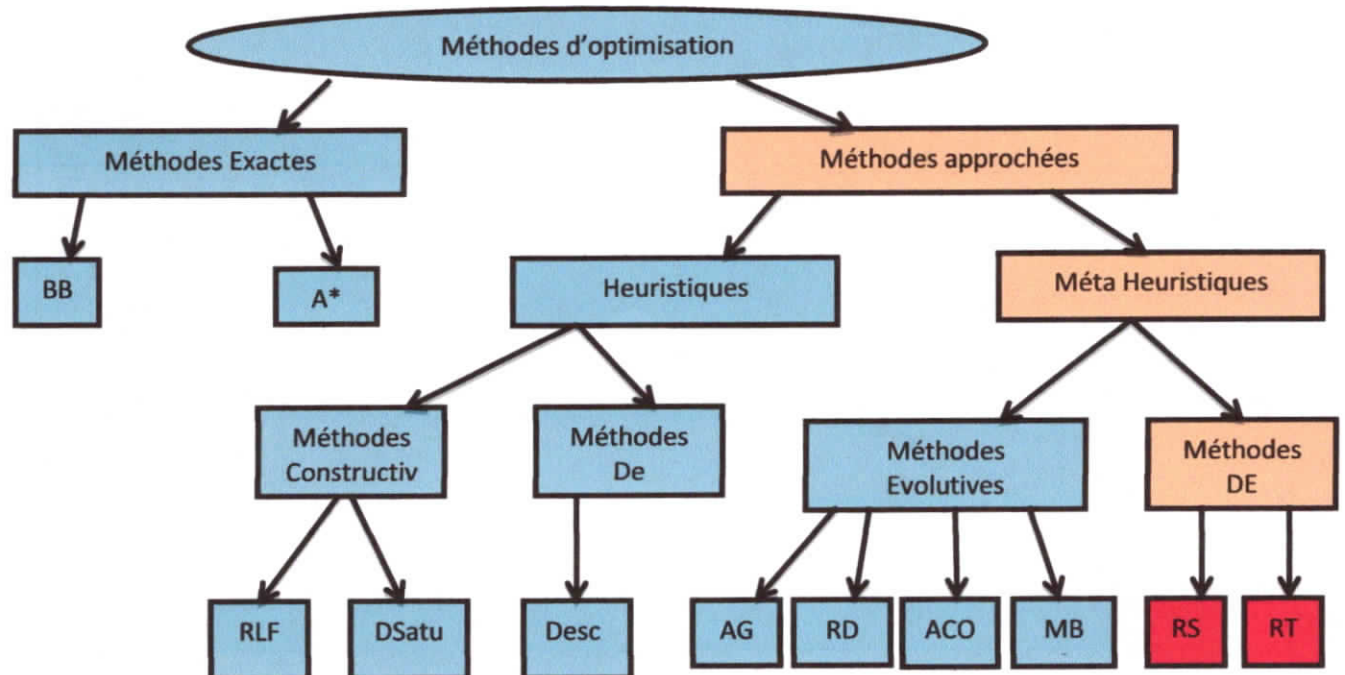


Figure II.1. Classification des méthodes d'optimisation

Avec :

BB : Branche & Bound.

RT : Recherche Tabou.

RLF : Le plus grand récursifs abord.

AG : Algorithmes Génétiques.

DSatur : Degré Saturation maximal.

RD : Recherche Dispersée.

Desc : Méthode de descente.

ACO : Optimisation par Colonie de Fourmis.

RS : Recuit Simulé.

MBO : Optimisation par colonie d'abeilles [8].

L'utilisation de méthodes exactes n'est pas toujours possible pour un problème donné à cause d'un certain nombre de contraintes, donc nous utilisons des méthodes approchées, qui contiennent deux types :

- **heuristique** : est un algorithme qui fournit rapidement (en un temps polynomial) une solution approchée et réalisable, pas nécessairement optimale, pour un problème d'optimisation difficile.
- **Méta-heuristiques** : Des heuristiques plus poussées, adaptables à un grand nombre de problèmes différents, sans changements majeurs dans l'algorithme, ont été mises au point et ont donné naissance à une nouvelle famille d'algorithmes d'optimisation stochastiques [11].

On peut distinguer deux grandes approches dans les méta-heuristiques:

- ☞ **les approches « trajectoire »**: Ces méthodes partent d'une solution initiale (obtenue de façon exacte, ou par tirage aléatoire) et s'en éloignent progressivement, pour réaliser une trajectoire, un parcours progressif dans l'espace des solutions. Dans cette catégorie, se rangent : la méthode de descente, le recuit simulé, la méthode Tabou. Le terme de recherche locale est de plus en plus utilisé pour qualifier ces méthodes.
- ☞ **les approches « population » (ou évolutionnaires)** : Elles consistent à travailler avec un ensemble de solutions simultanément, que l'on fait évoluer graduellement. L'utilisation de plusieurs solutions simultanément permet naturellement d'améliorer l'exploration de l'espace des configurations. Dans cette seconde catégorie, on recense : les algorithmes génétiques, les algorithmes par colonies de fourmi, l'optimisation par essaim particulaire...

II.4 Méta-heuristique

Une méta-heuristique est une stratégie générale applicable à un grand nombre de problèmes, à partir de laquelle on peut dériver un algorithme heuristique pour un problème particulier [3]. Sont apparues dans les années 80.

On peut dire que les propriétés fondamentales des méta-heuristiques sont les suivantes :

- Les méta-heuristiques sont des stratégies qui permettent de guider la recherche d'une solution optimale
- Le but visé par les méta-heuristiques est d'explorer l'espace de recherche efficacement afin de déterminer des solutions (presque) optimales.
- Les techniques qui constituent des algorithmes de type méta-heuristique vont de la simple procédure de recherche locale à des processus d'apprentissage complexes.
- Les méta-heuristiques sont en général non-déterministes et ne donnent aucune garantie d'optimalité
- Les méta-heuristiques peuvent contenir des mécanismes qui permettent d'éviter d'être bloqué dans des régions de l'espace de recherches.
- Les concepts de base des méta-heuristiques peuvent être décrits de manière abstraite, sans faire appel à un problème spécifique.
- Les méta-heuristiques peuvent faire appel à des heuristiques qui tiennent compte de la spécificité du problème traité, mais ces heuristiques sont contrôlées par une stratégie de niveau supérieur.
- Les méta-heuristiques peuvent faire usage de l'expérience accumulée durant la recherche de l'optimum, pour mieux guider la suite du processus de recherche [13].

II.5 Recherche locale

L'espace de recherche associé à un problème d'optimisation combinatoire est souvent non énumérable en un temps raisonnable. On essaie donc de relier certaines solutions entre elles. Ainsi, à partir d'une solution, on peut en trouver une autre et ainsi de suite. Il est nécessaire de définir une relation de voisinage qui est une application qui associe à toute solution de l'espace de recherche un voisinage c.-à-d. un ensemble de solutions appelées voisins. Les recherches locales sont des méthodes fondées sur une relation de voisinage et sur une procédure exploitant ce voisinage [14].

II.5.1 Définition

La recherche locale appelée aussi la descente stochastique ou l'amélioration itérative ou même le *Hill Climbing*, représente une classe de méthodes heuristiques très anciennes. Les algorithmes de recherche locale sont largement utilisés dans les problèmes d'optimisation difficiles, tels que les problèmes informatiques (en particulier l'intelligence artificielle), mathématiques, en recherche opérationnelle, d'ingénierie et de bio-informatique.

II.5.2 Principe

Le principe de la méthode de recherche locale consiste à partir d'une solution s et à choisir une solution s' dans un voisinage de s , telle que s' améliore la recherche (généralement telle que $f(s') < f(s)$). On peut décider soit d'examiner toutes les solutions du voisinage et prendre la meilleure de toutes (ou prendre la première trouvée), soit d'examiner un sous-ensemble du voisinage.

II.5.3 Voisinage

- Un voisinage ou une structure de voisinage est une fonction N qui associe un sous-ensemble de Ω à toute solution $x \in \Omega$. Une solution $x' \in N(x)$ est dite voisine de x .
- Une solution $x \in \Omega$ est un minimum local relativement à la structure de voisinage N si $f(x) \leq f(x')$ pour tout $x' \in N(x)$.
- Une solution $x \in \Omega$ est un minimum global si $f(x) \leq f(x')$ pour tout $x' \in \Omega$.
- Les voisinages dépendent du problème. Cet aspect est donc laissé générique lors de la définition d'une méta-heuristique [15].

II.6 Recuit simulé

II.6.1 Historique

Le recuit simulé (RS) a été introduit par (Kirkpatrick et al. 1983) et (Cerný 1985) comme une méthode de recherche locale normale, utilisant une stratégie pour éviter les minima locaux. Cette méta-heuristique est basée sur une technique utilisée depuis longtemps par les métallurgistes qui, pour obtenir un alliage sans défaut, faisant alterner les cycles de réchauffage (ou de recuit) et de refroidissement lent des métaux. Le recuit simulé s'appuie sur des travaux faits par (Metropolis et al. 1953), qui ont pu décrire l'évolution d'un système de thermodynamique (recuit physique) [16].

Le recuit physique est un processus utilisé en métallurgie pour améliorer la qualité d'un solide. On cherche à atteindre un état d'énergie minimale qui correspond à une structure stable du métal. En partant d'une haute température à laquelle la matière est devenue liquide, la phase de refroidissement conduit la matière à retrouver sa forme solide par une diminution progressive de la température [17].

II.6.2 Définition

L'idée principale du recuit simulé tel qu'il a été proposé par Metropolis en 1953 est de simuler le comportement de la matière dans le processus du recuit très largement utilisé dans la métallurgie. Le but est d'atteindre un état d'équilibre thermodynamique, cet état d'équilibre (où l'énergie est minimale) représente - dans la méthode du recuit simulé - la solution optimale d'un problème ; L'énergie du système sera calculée par une fonction d'évaluation spécifique à chaque problème [17]. La méthode va donc essayer de trouver la solution optimale en optimisant une fonction d'évaluation, pour cela, un paramètre fictif de température a été ajouté par Kirkpatrick, Gelatt et Vecchi [18].

En gros le principe consiste à générer successivement des configurations à partir d'une solution initiale S_0 et d'une température initiale T_0 qui diminuera tout au long du processus jusqu'à atteindre une température finale ou un état d'équilibre (optimum global) [18].

II.6.3 Principe de base

L'idée est d'effectuer un mouvement selon une distribution de probabilité qui dépend de la qualité des différents voisins :

- Les meilleurs voisins ont une probabilité plus élevée.
- Les moins bons ont une probabilité plus faible.

On utilise un paramètre, appelé la température (notée T) :

- T élevée : tous les voisins ont à peu près la même probabilité d'être acceptés.
- T faible : un mouvement qui dégrade la fonction d'évaluation a une faible probabilité d'être choisi.
- $T = 0$: aucune dégradation de la fonction d'évaluation n'est acceptée.

La température varie au cours de la recherche : T est élevée au début, puis diminue et finit par tendre vers zéro (fonction d'évaluation minimale).

II.6.4 Notions

Il y a plusieurs notions à définir telles que la probabilité de Boltzmann et le critère de Metropolis.

II.6.4.1 Probabilité de Boltzmann

La probabilité de Boltzmann, notée P_T mesure la probabilité de trouver un système dans une configuration i avec une énergie E_i , à une température T donnée, dans l'espace des configurations S [19]. Elle est définie par :

$$P_T(X = 1) = \frac{1}{Z(T)} e^{\frac{-E_i}{kT}} \quad (II.1)$$

Où X est une variable stochastique qui désigne l'état actuel du solide, K est appelé la constante de Boltzmann et $Z(T)$ est une fonction appelé fonction de répartition définie par :

$$Z(T) = \sum_{j=1}^t e^{\frac{-E_j}{kT}} \quad (II.2)$$

Où t représente tous les états énergétiques possibles.

Dans cette expression, le facteur KT montre que lorsque la température est très élevée, tous les états sont à peu près équiprobables, c'est-à-dire qu'un grand nombre de configuration sont accessibles. Au contraire quand la température est basse, les états à haute énergie deviennent peu probables par rapport à ceux de faible énergie.

Pour simuler l'évolution d'un solide vers l'équilibre thermique pour une température T , Metropolis [20] a proposé un critère appelé critère de Metropolis et qui est dérivé de la probabilité de Boltzmann.

II.6.4.2 Critère de Metropolis

Dans le contexte d'optimisation par la méthode du (RS), l'énergie est remplacée par la fonction d'évaluation, ainsi l'obtention d'un solide à énergie minimum est équivalente à la recherche de l'optimum global de la fonction d'évaluation. Cette recherche se fait par explorations successives de différentes configurations.

Après chaque passage d'une configuration X à une configuration Y , la variation de la fonction d'évaluation est $\Delta f = f(Y) - f(X)$. La transformation est acceptée selon la probabilité $P(X, Y)$ telle que :

$$P(X, Y) = e^{\frac{-\Delta f}{T}} \quad (II.3)$$

Lorsque la variation Δf est négative ou nulle, l'exponentielle est supérieure ou égale à 1 et la nouvelle configuration est acceptée.

- Si $\Delta f > 0$, $P(X, Y)$ est comparé à un nombre aléatoire $\mathbf{rand} \in [0,1]$.
- Si $\mathbf{rand} < P(X, Y)$ la configuration Y est acceptée.
- Sinon elle est rejetée et une autre configuration est essayée.

Les configurations ayant une augmentation en Δf , c'est-à-dire une dégradation de la fonction d'évaluation sont donc moins probables pour une température donnée, d'autant moins que la température est faible.

II.6.4.3 Chaîne de Markov et équilibre thermique

La chaîne de Markov est l'ensemble des configurations explorées à température constante. L'équilibre thermique est caractérisé par la convergence de la distribution des énergies calculées sur les différentes configurations de la chaîne de Markov vers une loi normale. En pratique, ce critère d'équilibre thermique est trop sévère et il est remplacé par la notion de quasi-équilibre. Ce dernier est atteint lorsque la longueur de la chaîne de Markov, c'est-à-dire le nombre de configurations explorées, est suffisamment grande [21].

La figure suivante représente l'organigramme de fonctionnement de méthode du recuit simulé.

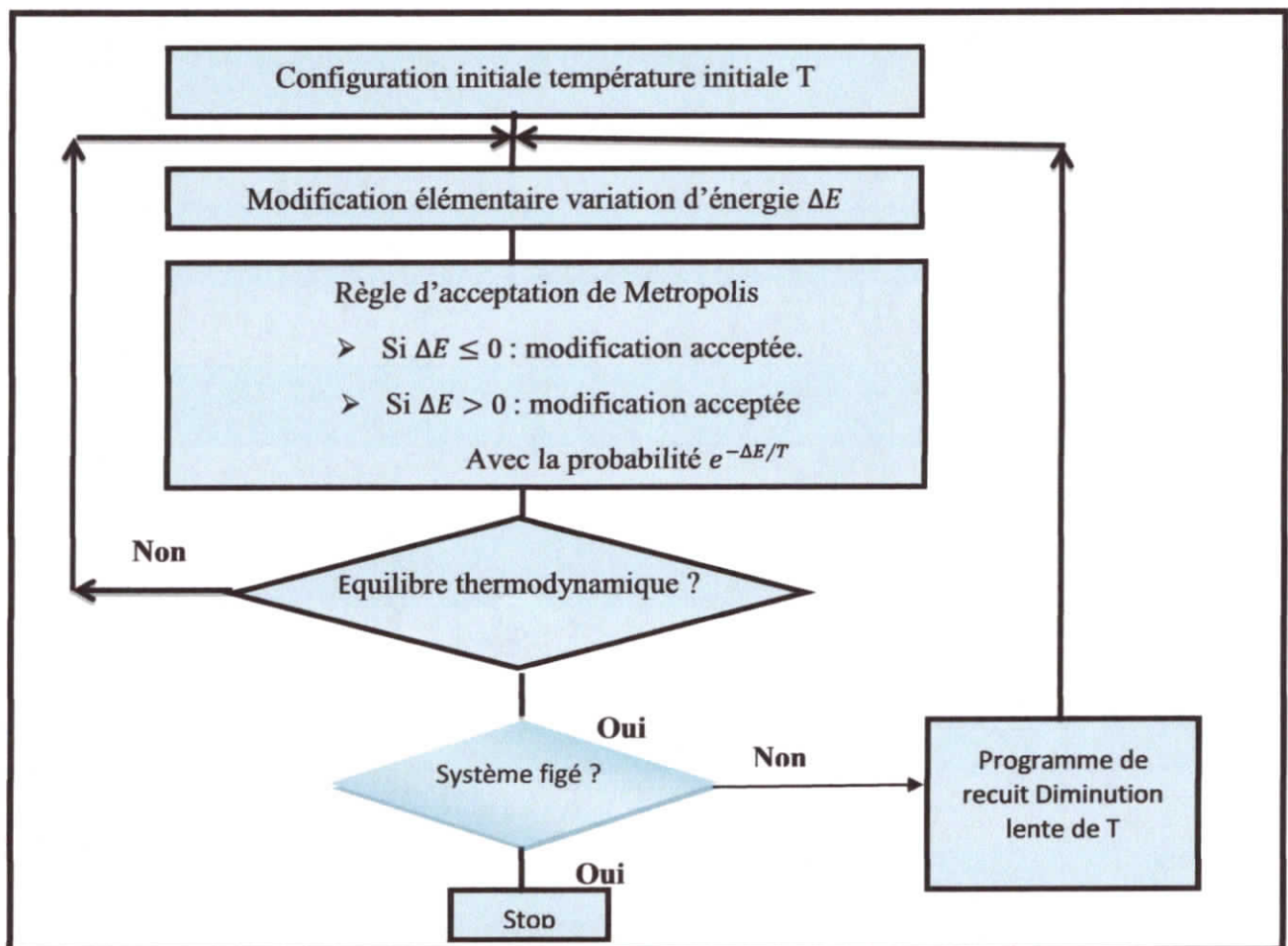


Figure II.2 : Fonctionnement de l'algorithme de recuit simulé.

II.6.5 Algorithme

1. Engendrer une configuration initiale S_0 de $S : S \leftarrow S_0$
2. Initialiser la température T en fonction du schéma de refroidissement
3. Répéter
4. Engendrer un voisin aléatoire S' de S
5. Calculer $\Delta E = f(S') - f(s)$.
6. Si $\Delta E \leq 0$ alors $S \leftarrow S'$
7. Sinon accepter S' comme la nouvelle solution avec la probabilité $P(E, T) = e^{-\Delta E/T}$
8. Fin si
9. Mettre T à jour en fonction du schéma de refroidissement (réduire la température) :

$$T \leftarrow T * \alpha ; \text{ avec } \alpha \in [0 \ 1]$$
10. Jusqu'à la condition d'arrêt
11. Retourner la meilleure configuration trouvée

II.6.6 Paramètres

Le choix de la température est primordial pour garantir l'équilibre entre l'intensification et la diversification des solutions dans l'espace de recherche. Premièrement, le choix de la température initiale dépend de la qualité de la solution de départ. Si cette solution est choisie aléatoirement, il faut prendre une température relativement élevée. On utilise souvent la règle suivante :

$$T_{k+1} \leftarrow T_{k+\alpha} \quad (II.4)$$

Où $\alpha \in [0, 1]$, un paramètre qui exprime la diminution de la température de l'itération k à $k+1$. La décroissance de la température peut également être réalisée par paliers (en d'autres termes, elle ne change qu'après un certain nombre d'itérations). Certains préconisent l'utilisation de stratégies non monotones. On peut ainsi rehausser la température lorsque la recherche semble bloquée dans une région de l'espace de recherche. On peut alors considérer une grande augmentation de la température comme un processus de diversification alors que la décroissance de la température correspond à un processus d'intensification [16].

II.6.6.1 Nombre d'itérations à température constante

Cette valeur doit être assez élevée pour atteindre le quasi-équilibre et suffisamment faible pour réduire le temps d'optimisation [8].

II.6.6.2 Critères d'arrêt

Un grand nombre de critères existent, le processus d'optimisation est arrêté quand aucune amélioration sensible n'est réalisée, ou quand le paramètre de contrôle est inférieur à une certaine valeur. *Vanderbilt et al* ont suggéré comme critère d'arrêt l'expression suivante [22]:

$$\frac{f' - f_{opt}}{f'} \leq \varepsilon \quad (II.5)$$

Où f_{opt} est la valeur de la fonction d'évaluation à l'optimum courant, f' est la moyenne des fonctions d'évaluation des configurations générées depuis le début de l'optimisation et ε est la précision fixée par l'utilisateur [22].

II.6.7 Avantages et Inconvénients du RS

II.6.7.1 Avantages

- Donne généralement des bonnes solutions par rapport aux algorithmes de recherche classiques.
- Peut être utilisé dans la plupart des problèmes d'optimisations.
- converge vers un optimum global.

II.6.7.2 Inconvénients

- Le principal inconvénient du recuit simulé est qu'une fois l'algorithme piégé à basse température dans un minimum local, il lui est impossible de s'en sortir.
- La difficulté de déterminer la température initiale.
- Dégradation des performances pour les problèmes où il y a peu de minimas locaux.

II.7 Recherche taboue

II.7.1 Historique

La recherche taboue (*Tabu Search*) est une méthode de recherche locale combinée avec un ensemble de techniques permettant d'éviter d'être piégé dans un minimum local ou la répétition d'un cycle. La recherche taboue (RT) est introduite principalement par Glover (Glover 1986), Hansen (Hansen 1986), Glover et Laguna dans (Glover et Laguna 1997). Cette méthode a montré une grande efficacité pour la résolution des problèmes d'optimisation difficiles [16].

II.7.2 Définition

C'est une méthode méta-heuristique utilisée pour la résolution des problèmes d'optimisation, destinée principalement à guider d'autres méthodes afin de trouver de meilleures solutions à partir d'une solution initiale obtenue par l'une des heuristiques [23].

II.7.3 Principe de base

C'est une procédure itérative qui, partant d'une solution initiale, tente de converger vers la solution optimale en exécutant, à chaque pas, un mouvement dans l'espace de recherche. Chaque pas consiste d'abord à engendrer un ensemble de solutions voisines de la solution courante pour ensuite en choisir la meilleure, même si ce choix entraîne une augmentation de la fonction d'évaluation à minimiser. En acceptant de détériorer la valeur de la solution courante, le minimum local peut être évité mais, en contrepartie, des parcours répétitifs sont déplorés. Aussi, pour palier l'inconvénient majeur des méthodes de recherche locale, la recherche taboue a pour but d'améliorer à chaque étape, la valeur de la fonction d'évaluation, en utilisant une mémoire afin de conserver les informations sur les solutions déjà visitées. Cette mémoire constitue la liste taboue qui va servir à interdire l'accès aux dernières solutions visitées. Lorsqu'un optimum local est atteint, il y a interdiction de revenir sur le même chemin. Un critère d'aspiration, est également utilisé pour lever l'interdiction l'utilisation d'un mouvement si ce dernier conduit à une meilleure solution [24].

II.7.4 Eléments de la RT

Les éléments de base de la RT sont brièvement indiqués et définis comme suit :

II.7.4.a Stratégie d'exploration régressive

La mémoire à court terme correspond à une stratégie d'exploration régressive, représente la partie principale de la *recherche Tabou* et a pour but de choisir le meilleur mouvement, ce mouvement pour être accepté doit vérifier certaines contraintes appelées restrictions Tabou. Ces restrictions Tabou sont conçues pour éviter les répétitions et les mouvements n'améliorant pas la solution.

II.7.4.b Liste Tabou

La liste Tabou représente la mémoire à court terme, elle contient les attributs des mouvements les plus réalisés.

Cette liste est maintenue dans le but d'orienter la recherche.

II.7.4.c Critère d'aspiration

Ce critère est souvent utilisé pour enlever les restrictions Tabou d'un mouvement de haute qualité. Il permet de passer outre certains cas interdits. Son utilisation principale consiste à passer l'interdiction d'un mouvement s'il permet d'obtenir un élément meilleur que la solution trouvée jusqu'à présent (consiste à révoquer le statut Tabou d'un mouvement si ce dernier permet d'atteindre une solution de qualité supérieure à celle de la meilleure solution trouvée).

II.7.4.d Critère d'intensification

L'intensification consiste à retourner à l'une des meilleures solutions trouvées jusqu'à présent, puis de reprendre la recherche à partir de cette solution.

La stratégie d'intensification est matérialisée dans l'algorithme suivant par renforcement de la recherche dans la liste des meilleurs mouvements.

II.7.4.e Critère de diversification

Elle consiste à générer une nouvelle solution, différente de celle déjà explorées, dans le but de partir dans une nouvelle direction, pour explorer une autre région.

II.7.5 Critère d'arrêt

En règle générale, pour interrompre l'algorithme, on prend deux critères, tout d'abord on vérifie à chaque itération que le compteur d'itérations totales n'a pas dépassé un nombre maximal depuis le début de processus. On se base sur une autre valeur seuil, qui correspond au nombre maximal d'itérations que l'on s'autorise entre deux modifications (améliorations) consécutives de la (meilleure) solution.

Mais au lieu de considérer le nombre d'itérations, on pourrait aussi se baser sur le temps total, qui devrait être inférieur à une valeur maximale [25].

II.7.6 Algorithme

La figure suivante représente l'organigramme standard de la recherche taboue [26].

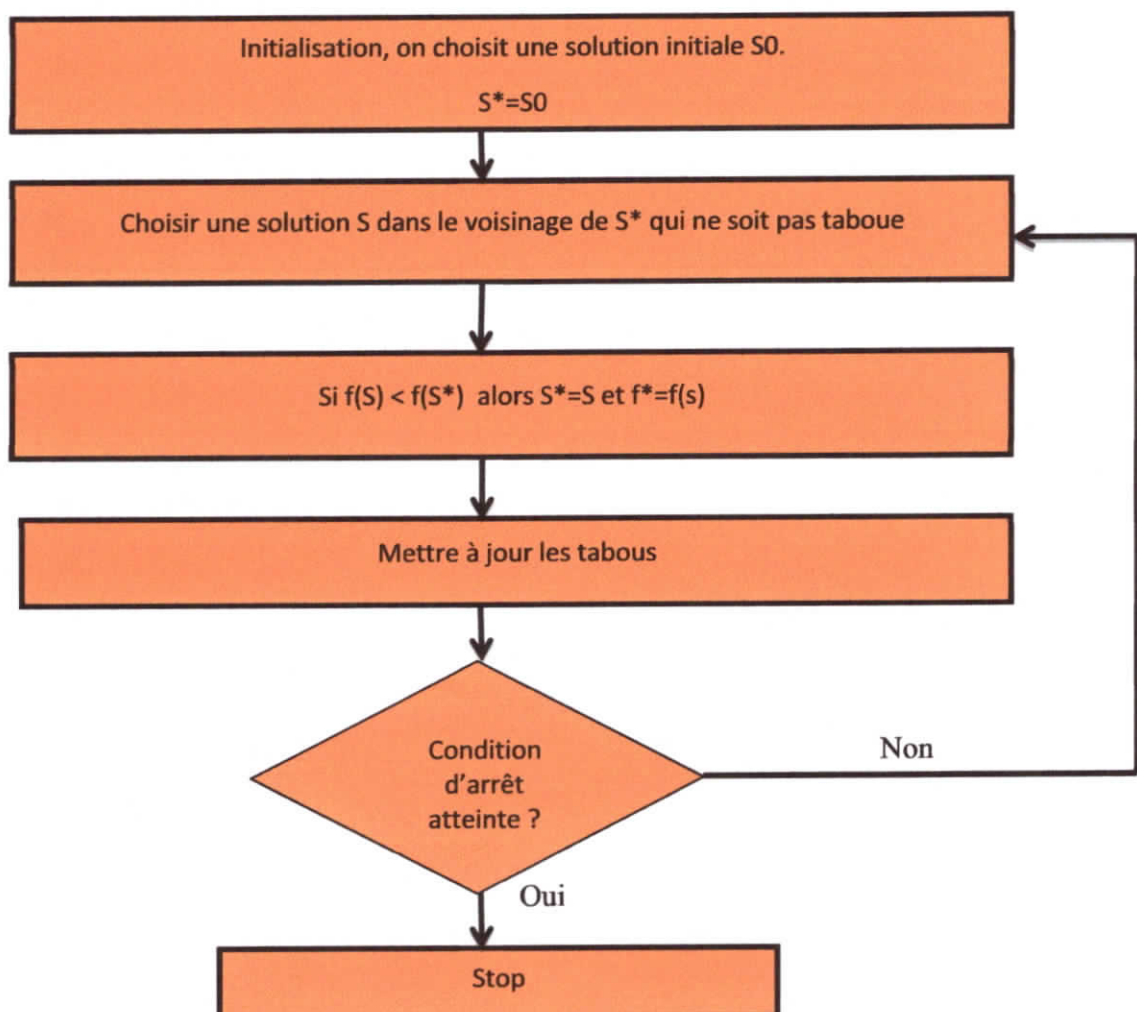


Figure II.3 : L'organigramme général de l'algorithme de la recherche taboue.

On note :

S_0 : solution initiale

S^* : meilleure solution jusqu'à présent.

S : nouvelles solutions du voisinage de S^*

$f(x)$: fonction d'évaluation à minimiser.

$f(S^*)$: fonction d'évaluation de la meilleure solution

II.7.7 Avantages et Inconvénients

II.7.7.1 Avantages

- Offre des économies de temps de résolution pour des programmes de grosse taille.
- Très bons résultats sur certains types de problèmes.
- Algorithmes faciles à mettre en œuvre.

II.7.7.2 Inconvénients

- Paramètres peu intuitifs.
- Demande en ressources importantes si la liste des tabous est trop imposante.
- Aucune démonstration de la convergence [23].



II.8 conclusion

Les méta-heuristiques sont souvent employées pour leur facilité de programmation et de manipulation. Elles sont en effet facilement adaptables à tout type de problème d'optimisation. Toutefois, elles sont le plus judicieusement employées sur des problèmes d'optimisation difficiles, ou des méthodes d'optimisation plus classiques (méthodes déterministes, notamment) montrent leurs limites.

Les méta-heuristiques, comprenant notamment la méthode de recherche Tabou, le recuit simulé sont des algorithmes d'optimisation de type stochastiques qui progressent vers un optimum par échantillonnage d'une fonction d'évaluation dont le but est la résolution de problèmes d'optimisation difficile.

Grâce à la simplicité et la souplesse de leurs principes, ils peuvent être un outil d'optimisation et de conception des systèmes de contrôle des processus complexes dont la dynamique n'est pas encore maîtrisée.



Chapitre III

Simulation

&

Interprétations

Simulation & Interprétations

III.1 Introduction

Devant les problèmes délicats de la commande de systèmes complexes, les outils utilisés deviennent de plus en plus pointus. L'un des axes de recherche concerne la commande des systèmes non linéaires mono variable qu'on va utiliser pour illustrer l'algorithme de la recherche taboue et l'algorithme de recuit simulé. Pour cela on a choisi le système de lévitation magnétique pour appliquer la commande floue ainsi que la recherche taboue et le recuit simulé qui vont optimiser le contrôleur flou appliqué au système.

Le problème de commande de lévitation magnétique est vraisemblablement un des systèmes les plus utilisés récemment pour tester différentes méthodes de commandes et optimisations.

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 Takagi-Sugeno d'ordre Zéro 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 de la recherche taboue et un algorithme basée sur le recuit simulé. Cette optimisation concerne les paramètres des prémisses et des conclusions de règles floues.

III.2 Le modèle mathématique de lévitation magnétique

La structure du procédé de contrôle de la position de la balle magnétique de système de lévitation magnétique est donnée par la figure suivante [27] :

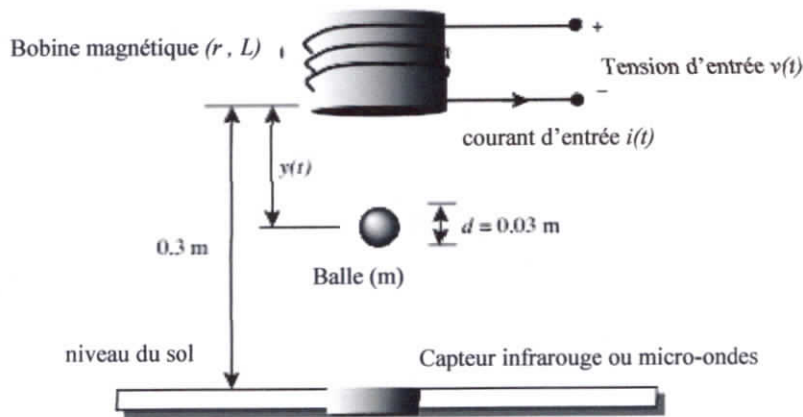


Figure III.1 : structure du procédé.

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

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

$$v(t) = r * i(t) + L * \frac{di(t)}{dt} \quad (III.2)$$

Avec :

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

m : La masse de la boule.

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 et 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:

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

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 à $x_3^2(t)$ 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 jacobienne en $y(0)$.

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

$$\begin{cases} \frac{dx_1}{dt} = x_2(t) \\ \frac{dx_2(t)}{dt} = \frac{g}{y(0)} x_1(t) - 2 \sqrt{\frac{g}{m y(0)}} x_3(t) \\ \frac{dx_3(t)}{dt} = -\frac{r}{L} x_3(t) + \frac{1}{L} v(t) \end{cases} \quad (III.4)$$

Puisque la position de la balle $y(t)$ est la seule sortie physique de la plante, en supposant que toutes les conditions initiales sont nulles pour le modèle de perturbation linéaire, nous pouvons réécrire le modèle comme fonction de transfert :

$$\frac{y(s)}{v(s)} = \frac{-\frac{2}{L} \sqrt{\frac{g}{m \cdot y(0)}}}{(s^2 - \frac{g}{y(0)})(s - \frac{r}{L})} \quad (III.5)$$

Notez qu'il existe trois pôles, dans la fonction de transfert dans l'équation (III.5). À partir de la figure(III.1), la distance totale entre la bobine magnétique et le niveau du sol est de 0,3 m, et la position initiale de la balle $y(0)$ peut être comprise entre 0,015 m (toucher la bobine) et 0,285 m (toucher le sol). Pour cette gamme, le numérateur de la fonction de transfert :

$$num = -\frac{2}{L} \sqrt{\frac{g}{m \cdot y(0)}} \quad (III.6)$$

Varies entre -323.3 (balle de 0.015m) et -74.17 (balle de 0.285m)[27].

III.3 la commande floue du système de lévitation magnétique

III.3.1 Le type de contrôleur utilisé

Le but de cette partie est de commander la position de la balle magnétique de lévitation magnétique. Le contrôleur utilisé est de type PD flou où la sortie du contrôleur flou est la commande $u(t)$.

III.3.2 Structure de base du contrôleur flou

On considère un contrôleur flou de type Takagi-Sugeno d'ordre zéro à deux entrées, l'erreur $e(t)$ et sa variation $\Delta e(t)$ et une sortie $u(t)$, la commande appliquée au système (figure III.2) :

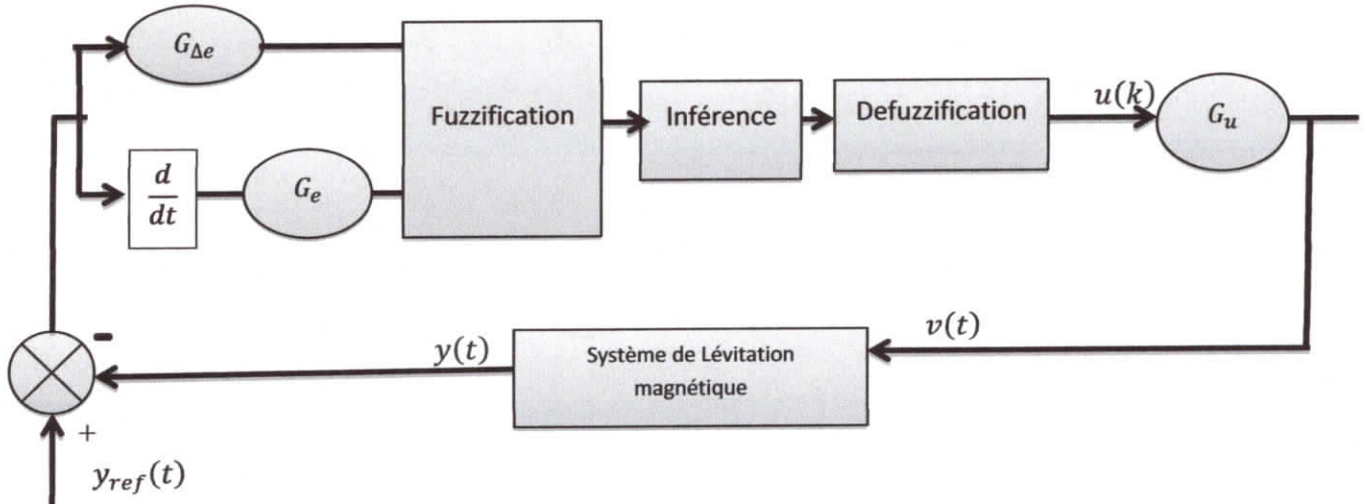


Figure III.2 : Structure de commander la position de la balle magnétique

Où $e(t) = y(t) - y_{ref}(t)$ est la différence entre la sortie désirée y_{ref} et la sortie mesurée du système commandé. Pour avoir une flexibilité dans l'implantation du régulateur, 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) sont utilisés pour avoir la dynamique désirée.

Pour l'optimisation paramétrique d'un contrôleur flou, on doit alors définir :

- Le nombre des ensembles flous pour chaque variable d'entrée et de sortie.
- Le type de fonctions d'appartenance qui constituent la partition floue de chaque variable linguistique.
- La structure des règles floues.
- Le type de mécanisme d'inférence, les opérateurs de connections et la méthode de Défuzzification.

Dans ce projet, le contrôleur flou utilisé est de type Takagi-Sugeno d'ordre zéro, ses blocs sont construits comme suit:

- Les fonctions d'appartenances triangulaire-trapézoïdales (TPE) sont utilisées pour la **fuzzification** de l'erreur e et sa dérivée Δe .
- La base de règles utilisée est celle de Macvicar et Whelan composée de **9 règles**.
- La méthode somme-produit est utilisée pour le **mécanisme d'inférences**.
- La méthode centre de gravité est utilisée pour la **défuzzification** de la sortie.
- $G_e, G_{\Delta e}, G_u$ sont les facteurs d'échelles de l'erreur, dérivée de l'erreur, et la commande respectivement.

Les figures III.3 et le tableau III.2 montrent respectivement, la forme des fonctions d'appartenance de l'erreur, sa dérivée et la commande. On utilise trois fonctions d'appartenance de type triangulaire-trapézoïdale pour chaque variables d'entrée, et 3

Singletons flous pour la sortie.

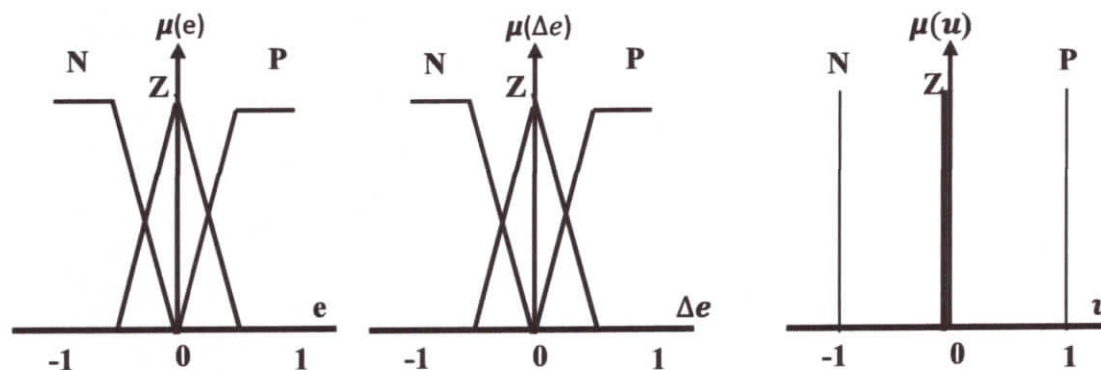


Figure III.3 : forme des fonctions d'appartenances.

		e		
		N	Z	P
Δe	N	P	P	Z
	Z	P	Z	N
	P	Z	N	N

Tableau III.2 : Base de règles du contrôleur flou utilisé.

III.3.3 Résultats de la simulation

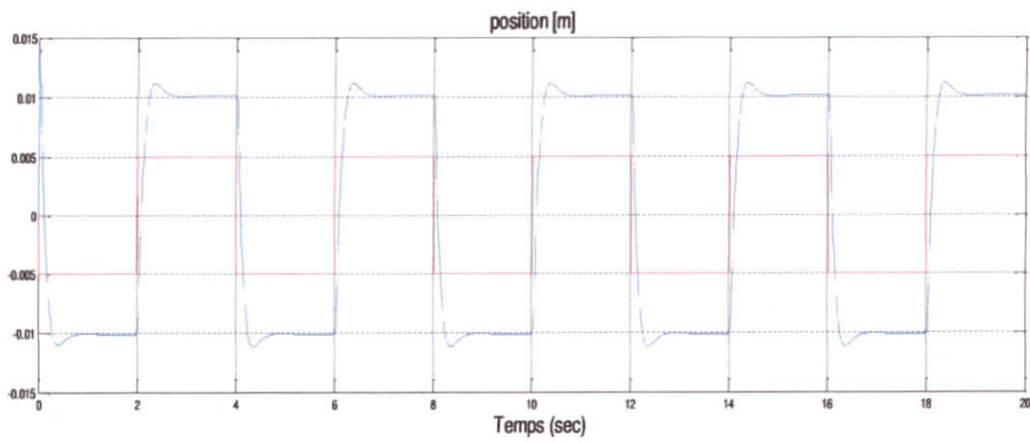


Figure III.4 : Evolution de la position et la référence en fonction du temps.

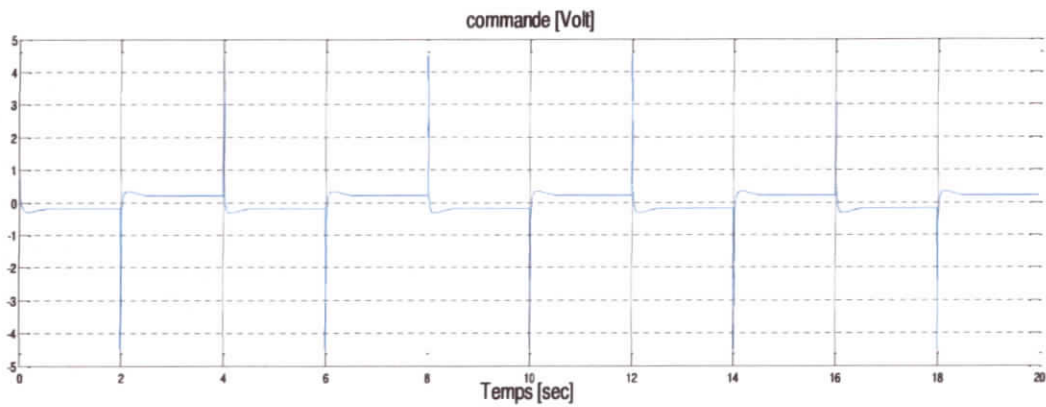


Figure III.5 : Evolution du signal de commande en fonction de temps.

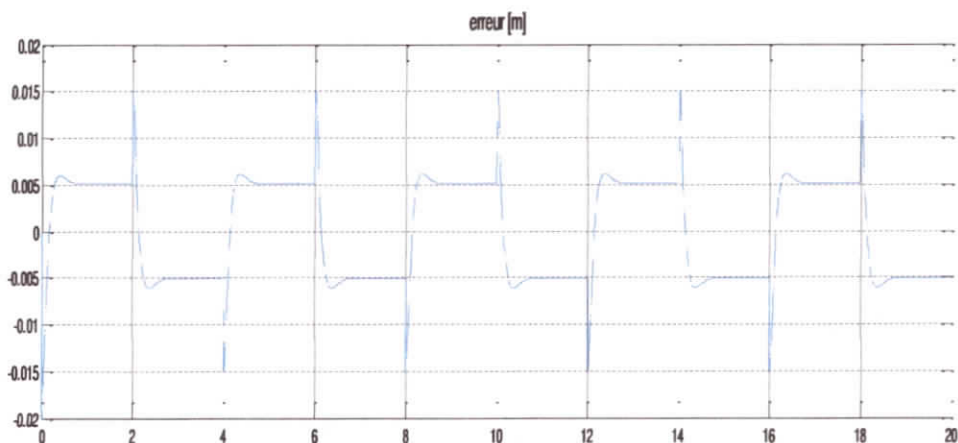


Figure III.6 : Evolution de l'erreur

D'après les figures précédentes, on remarque que le contrôleur arrive à stabiliser la balle de lévitation magnétique mais avec une erreur de 50% en un temps de réponse moins de 1s. Pour améliorer la précision, on va développer dans la partie suivante une méthode d'optimisation du contrôleur flou en utilisant les deux méthodes de recherche locale : la recherche taboue et le recuit simulé.

III.4 Optimisation du contrôleur flou par la recherche taboue et le recuit simulé :

Nous allons commencer par présenter la recherche taboue utilisée pour l'optimisation d'un régulateur de type Takagi-Sugeno d'ordre Zéro en vue de la commande de la position de la balle magnétique.

III.4.1 Stratégie d'optimisation du contrôleur flou :

Pour décrire le processus d'optimisation de la recherche taboue utilisée, considérant le schéma bloc fonctionnel donné dans la figure (III.7) ou e et Δe désignent respectivement l'erreur et la variation de l'erreur.

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 l'erreur quadratique moyenne défini comme suit :

$$F = \left(\frac{1}{nT}\right) * \sum_{k=1} e^2(k) \quad (III.8)$$

Où n est le nombre total d'échantillons, $e(k) = y_{ref}(k) - y(k)$ et T est le temps d'échantillonnage.

Les paramètres déterminés par le processus d'exploitation, et qui garantissent un contrôle optimal, sont :

- D'une part, les paramètres des prémisses (valeurs modales des ensembles flous de l'erreur et sa dérivée).
- D'autre part, les paramètres des conclusions des règles (dans notre cas des singletons) sachant que la base de règles est constituée de 9 règles présentées sur le tableau III.2.

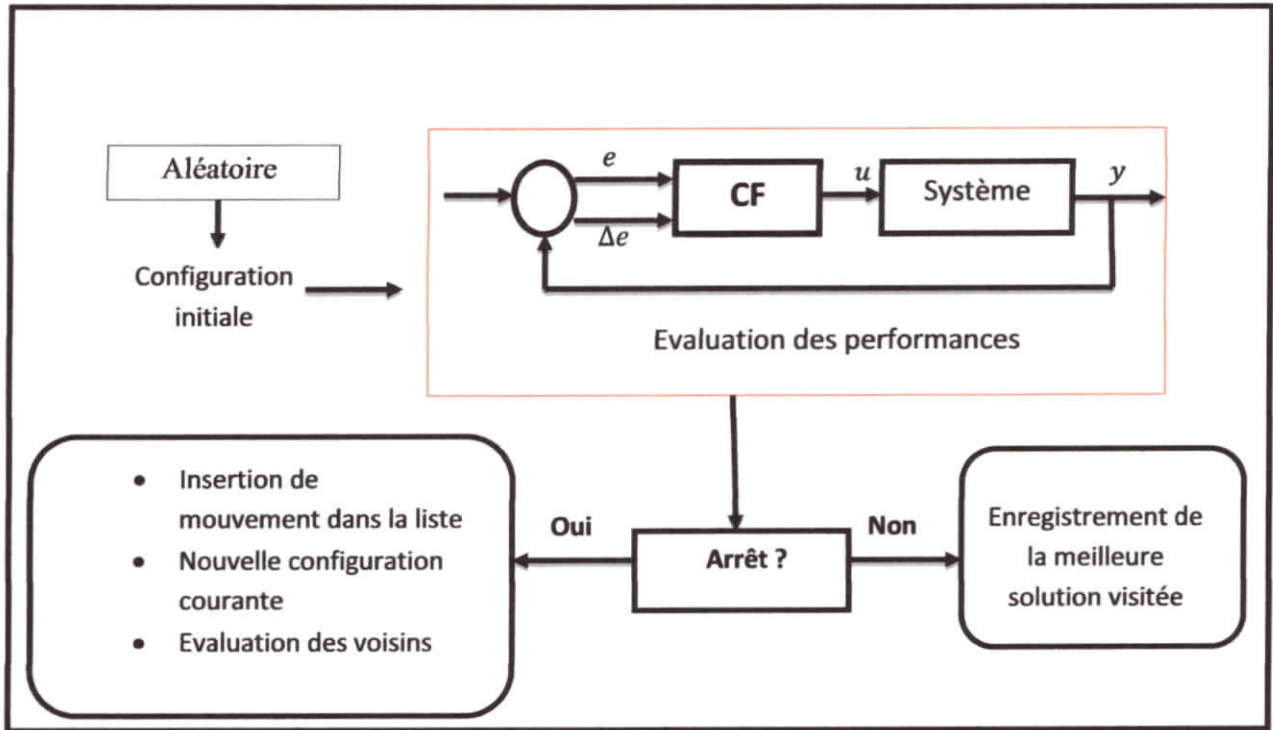


Figure III.7 : Schéma bloc fonctionnel du processus d'optimisation par la recherche taboue.

III.4.2 Optimisation du contrôleur flou par la méthode de recherche taboue

III.4.2.1 Application à la commande de la position de la balle magnétique

Le contrôleur flou initial est un contrôleur flou (CF) type Takagi-Sugeno d'ordre zéro.

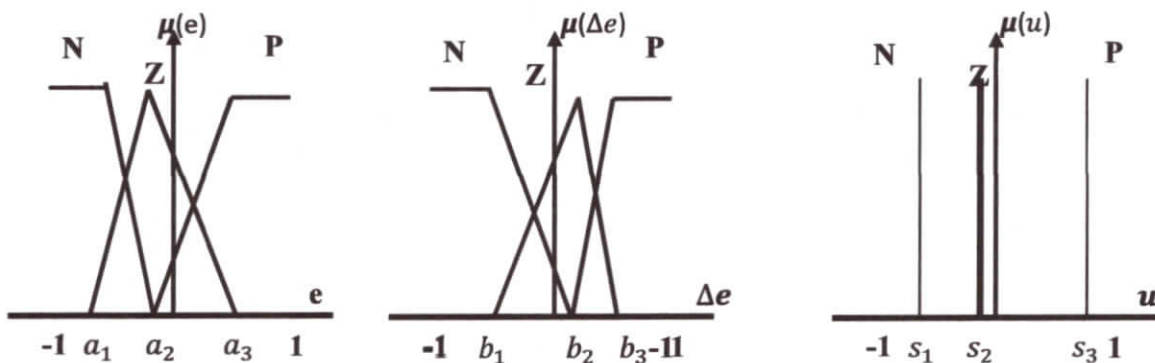


Figure III.8 : forme des fonctions d'appartenances à optimiser.

- ✓ Les valeurs modales (les sommets) des fonctions d'appartenance triangulaires respectivement a_1, a_2, a_3 , pour l'erreur et b_1, b_2, b_3 , pour la variation de l'erreur, tout en respectant la contrainte $\begin{cases} a_1 < a_2 < a_3 \\ b_1 < b_2 < b_3 \end{cases}$.
 - ✓ Les valeurs des conclusions des règles respectivement s_1, s_2, s_3 , tout en respectant la contrainte ($s_1 < s_2 < s_3$).
- La configuration de la recherche Taboue est la suivante :
- ✓ Le nombre de solutions voisines : 20.
 - ✓ La taille de liste taboue : 20.
 - ✓ Le nombre d'itérations : 10.

A. Résultats de simulation

L'évolution de la fonction objective est donnée par la figure (III.9). L'erreur quadratique moyenne est égale à : 3.39×10^{-6}

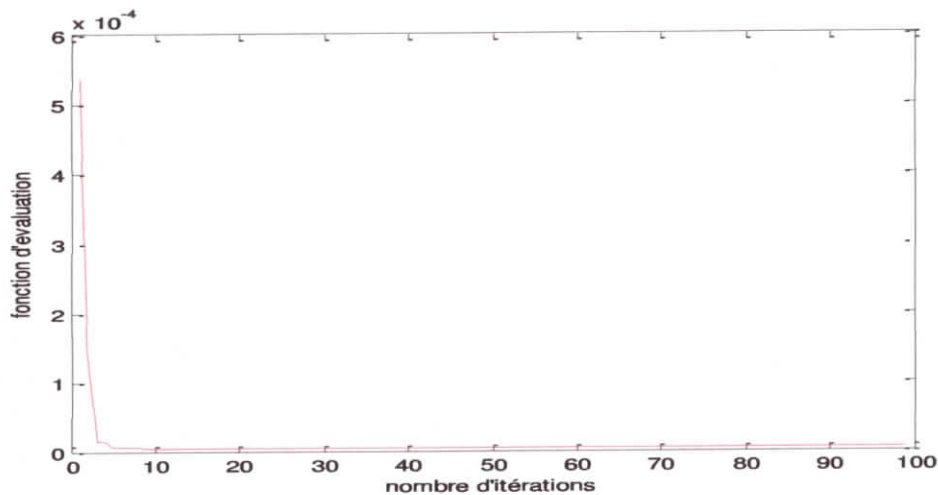


Figure III.9 Evolution de la fonction d'évaluation.

Les tableaux suivants donnent les paramètres optimaux obtenus par la recherche taboue après 10 itérations :

Les points modaux des fonctions d'appartenance			
e	-1.0000 (a_1)	-0.5000 (a_2)	0.8667 (a_3)
Δe	-0.9333 (b_1)	-0.1000 (b_2)	0.8000 (b_3)

Tableau III.3 Paramètres des prémisses optimaux obtenus par la recherche taboue

Les conclusions des règles		
N	Z	P
-1.0000	0.5000	0.6667

Tableau III.4 paramètres des conclusions optimaux obtenus par la recherche taboue

La figure (III.10) représente les formes des fonctions d'appartenance des entrées et de la sortie du contrôleur flou trouvées après optimisation par la RT:

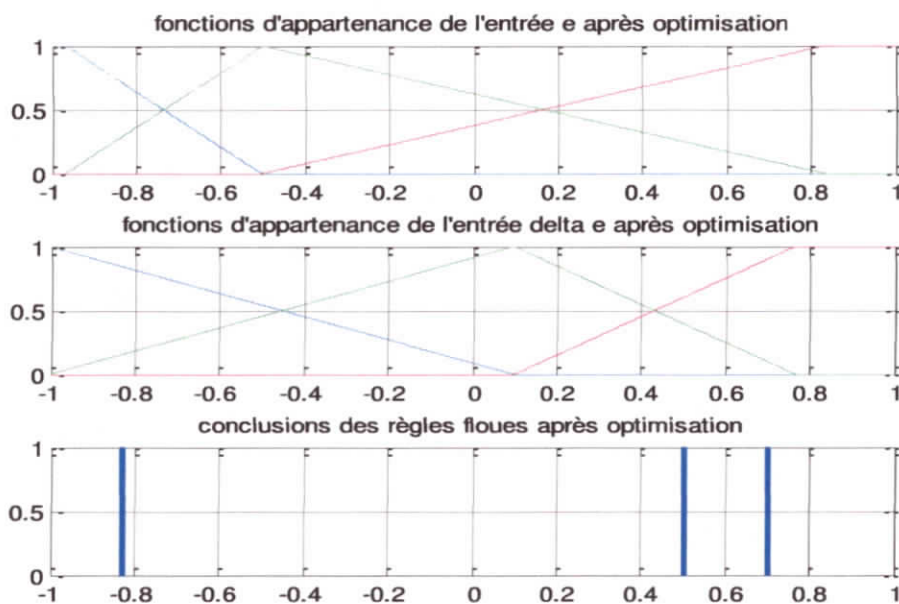


Figure III.10 Disposition et formes des fonctions d'appartenance des prémisses des conclusions après optimisation

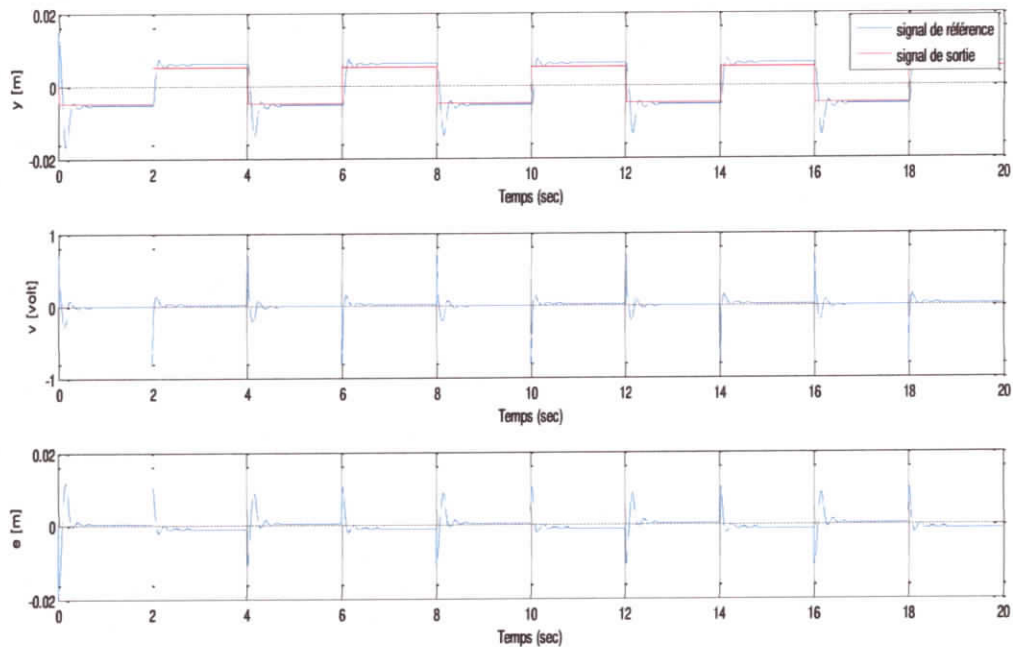


Figure III.11 Evolution de position, commande et d'erreur

B. Interprétation des résultats

La figure (III.11) montre, respectivement, l'évolution de la position, la commande appliquée et l'erreur pour le système nominal contrôlé. Ces résultats montrent que le contrôleur optimisé a pu stabiliser la balle de lévitation magnétique tout en suivant la trajectoire désirée avec de bonnes performances.

C. Test de robustesse

La robustesse du contrôleur est justifiée par sa capacité de bien réagir devant le changement des conditions initiales de fonctionnement ainsi que le changement des paramètres du système.

Premièrement, on a commencé par un changement du poids de la balle (voir la figure III.12). Les valeurs considérées du poids sont (50g, 100g, 200g, 300g) après par le changement de la valeur de la résistance (voir la figure III.13), Les valeurs considérées de la résistance sont (25Ω , 50Ω , 100Ω , 200Ω) ensuite par le changement des paramètres de la bobine magnétique (voir la figure III.14). Les valeurs considérées sont (0.25H, 0.5H, 1H, 2H).

Deuxièmement, on a fait le changement des conditions initiales (position initiale de la balle) illustrée dans la figure III.15, les valeurs considérées sont (7mm, 15mm ,30mm ,40mm).

➤ **Changement de valeur de masse de la balle**

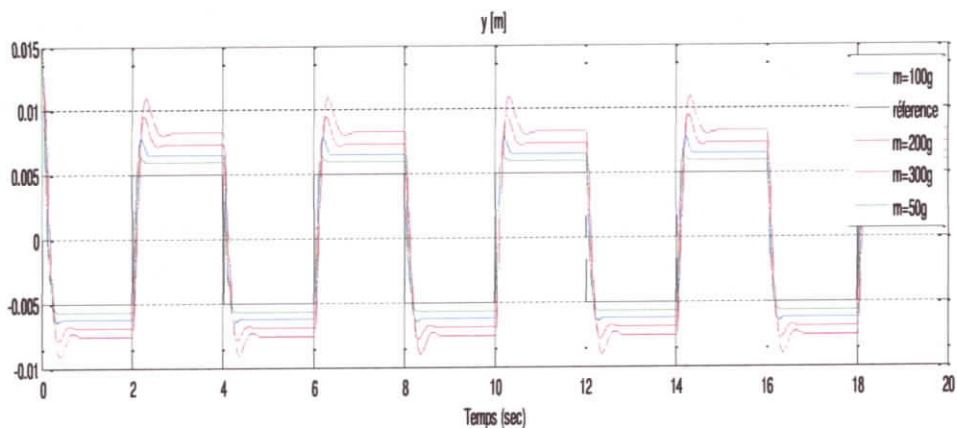


Figure III.12 : changement de valeur de masse de la balle

➤ **Changement de valeur de la résistance**

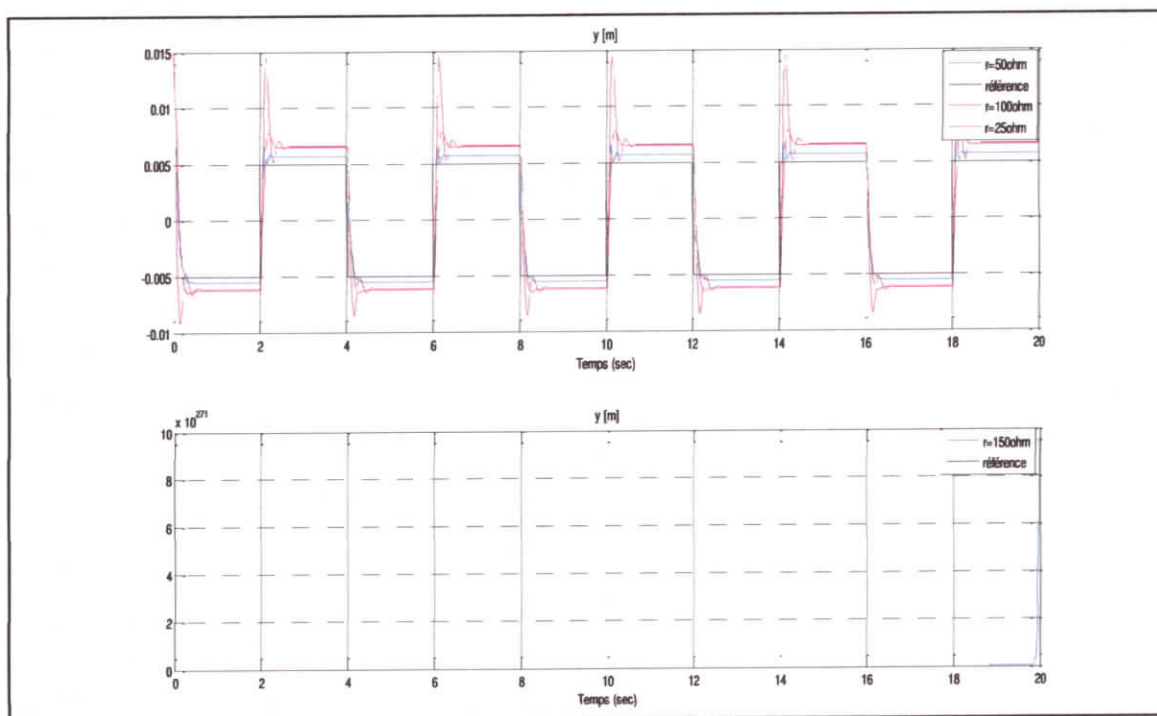


Figure III.13 : changement de valeur de la résistance

➤ **Changement de valeur de la bobine**

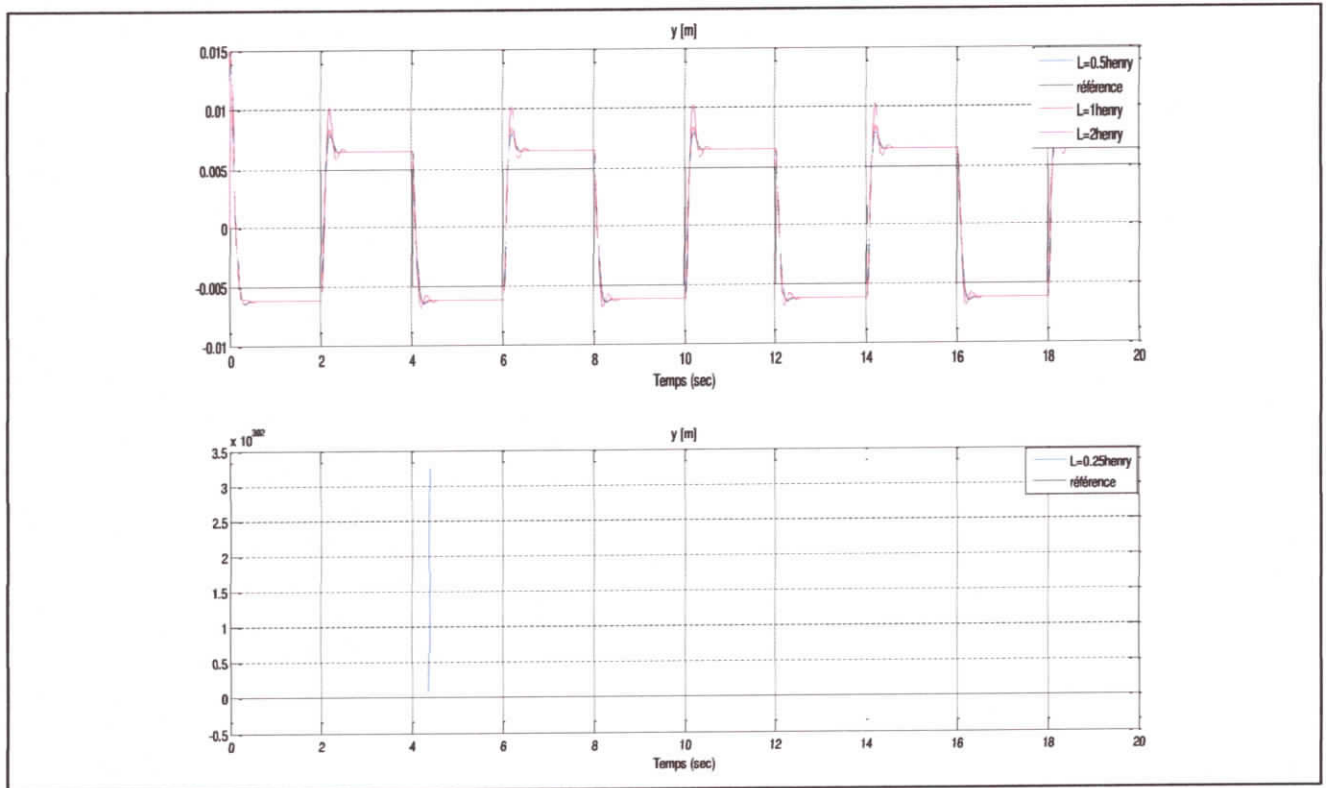


Figure III.14 : changement de valeur de la bobine

➤ **Changement de conditions initiales ($y(0)$)**

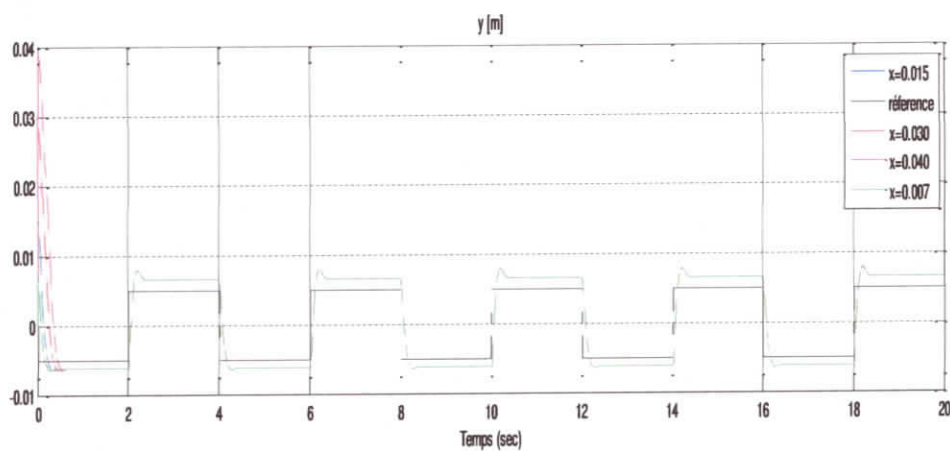


Figure III.15 : changement de la position initiale de la balle

En analysant les figures précédentes, nous pouvons conclure que :

- ✓ Le changement de la condition initiale et le poids de la balle n'influe pas sur la réponse du système, contrairement en changeant la valeur de la résistance ou la valeur de la bobine, le système entre dans une instabilité (divergence de la réponse) car leurs valeurs influent directement sur la tension (commande) qui contrôle la position de la balle.
- ✓ la stratégie proposée de conception accomplit, efficacement, les performances désirées.

III.4.3 Optimisation du contrôleur flou par la méthode de recuit simulé :

III.4.3.1 Organigramme générale du recuit simulé

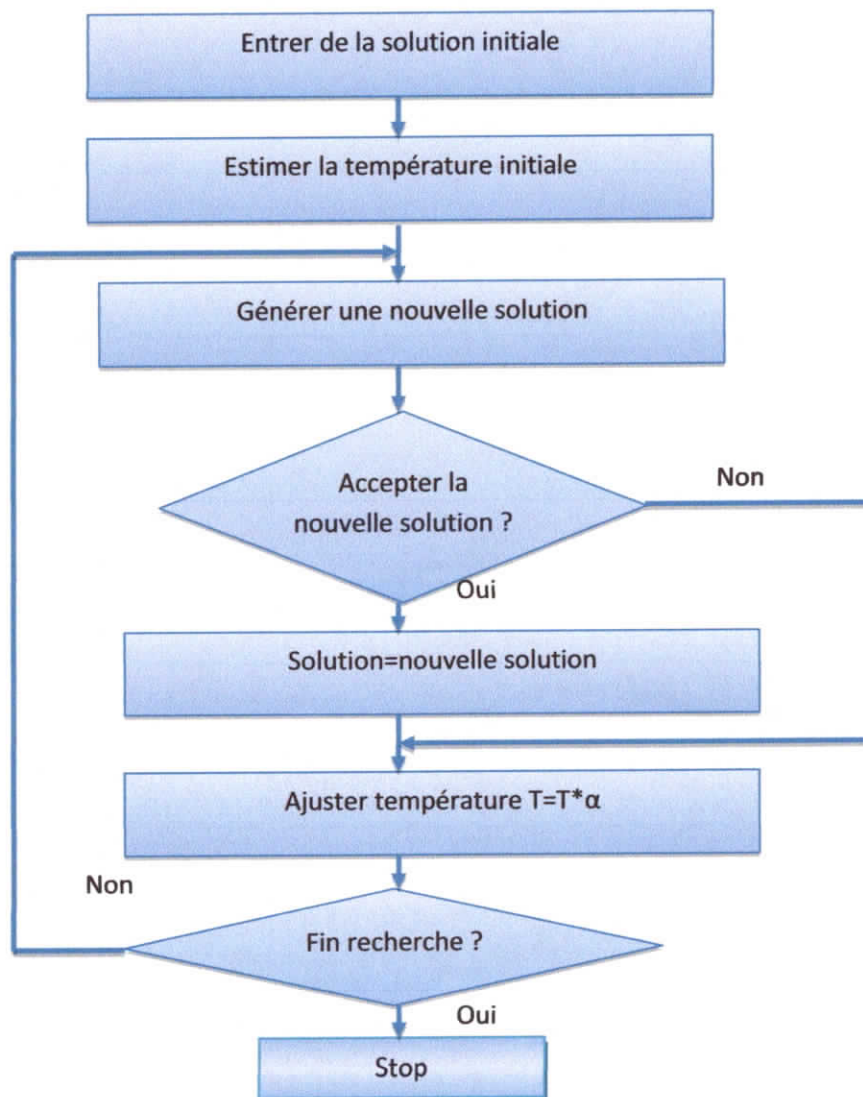


Figure III.16 Organigramme général du recuit simulé utilisé pour l'optimisation

III.4.3.2 Application à la commande de la position de la balle magnétique

Le contrôleur flou initial est un contrôleur flou (CF) type Takagi-Sugeno d'ordre zéro, à neuf règles.

L'objectif de l'optimisation est de trouver :

- Les valeurs modales (les sommets) des fonctions d'appartenance triangulaires respectivement a_1, a_2, a_3 , pour l'erreur et b_1, b_2, b_3 , pour la variation de l'erreur, tout en respectant la contrainte $\begin{cases} a_1 < a_2 < a_3 \\ b_1 < b_2 < b_3 \end{cases}$.
- Les valeurs des conclusions des règles respectivement s_1, s_2, s_3 , tout en respectant la contrainte $(s_1 < s_2 < s_3)$.

Les paramètres introduits au recuit simulé sont :

- La Température initiale : 100 °c.
- Le nombre d'itérations : 100.
- Le paramètre $\alpha = 0.99$.

A. Résultat de simulation

L'évolution de la fonction objective est donnée par la figure (III.17). A la fin d'exécution du programme, l'erreur quadratique moyenne est égale à : $2.70 \cdot 10^{-5}$

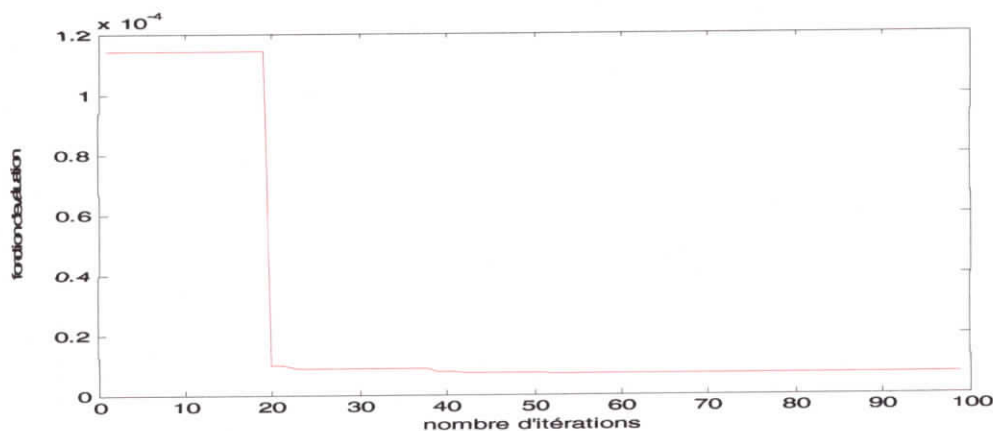


Figure III.17 : Evolution de la fonction d'évaluation

Les tableaux donnent les paramètres optimaux obtenus par le recuit simulé après 100 itérations :

Les points modaux des fonctions d'appartenance			
e	-0.7745 (a_1)	0.1828 (a_2)	0.7972 (a_3)
Δe	-0.5782 (b_1)	-0.4252 (b_2)	0.6260 (b_3)

Tableau III.5 Paramètres des prémisses optimaux obtenus par le recuit simulé

Les conclusions des règles		
N	Z	P
-0.6543	-0.0783	0.6641

Tableau III.6 paramètres des conclusions optimaux obtenus par le recuit simulé

La figure (III.18) représente les formes des fonctions d'appartenance des entrées et de la sortie du contrôleur flou trouvées après optimisation :

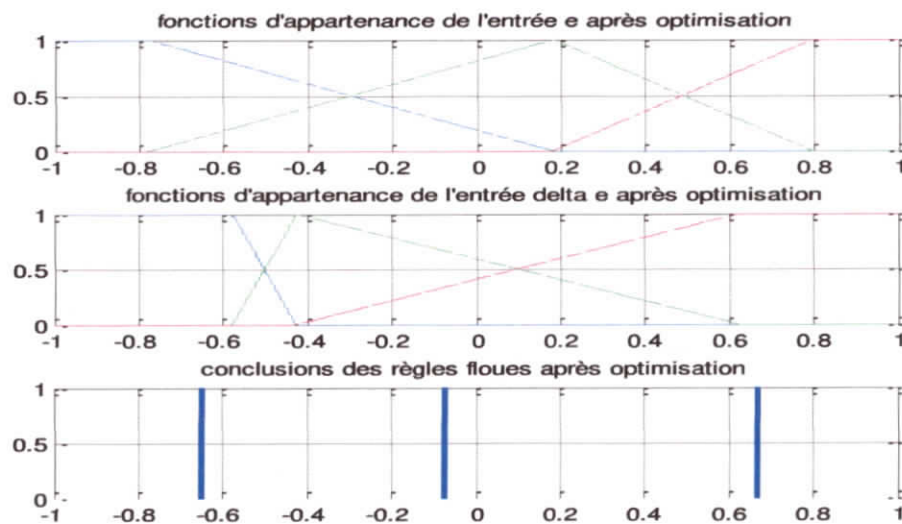


Figure III.18 Disposition et formes des fonctions d'appartenance des prémisses des conclusions après optimisation

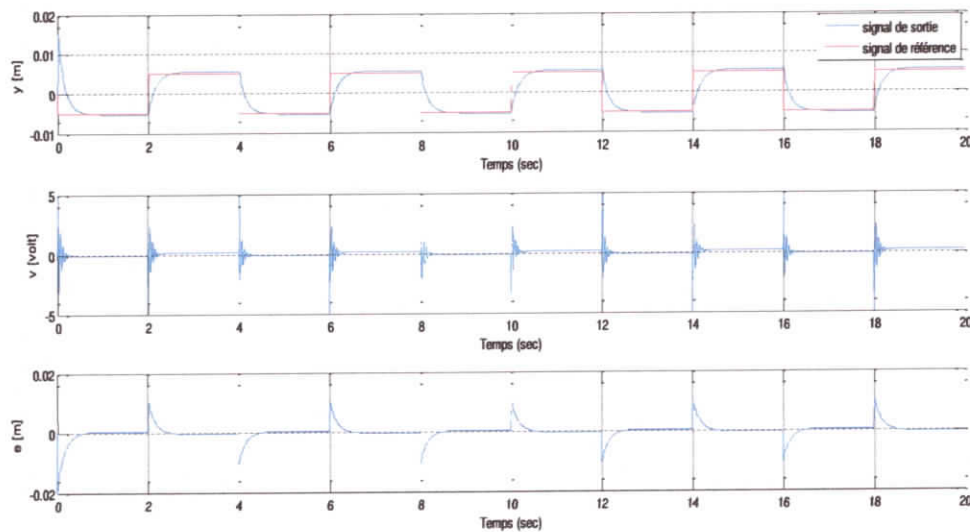


Figure III.19 Evolution de la position, la commande et l'erreur en fonction du temps

B. Interprétation des résultats

La figure (III.19) montre, respectivement, l'évolution de la position, la commande appliquée et l'erreur pour le système nominal contrôlé. Ces résultats montrent que le système est convergé vers la référence (trajectoire désirée) avec un temps de réponse moins d'une seconde en garantissant la stabilité et la précision.

C. Test de robustesse

Pour tester la réaction du contrôleur flou aux changements des paramètres du système, nous avons procédé à une variation de 0% à 300% des paramètres : M , R , L , y_0 du système. Les résultats présentés respectivement sur les figures (III.20), (III.21), (III.22), (III.23).

➤ **Changement de valeur de masse de la balle**

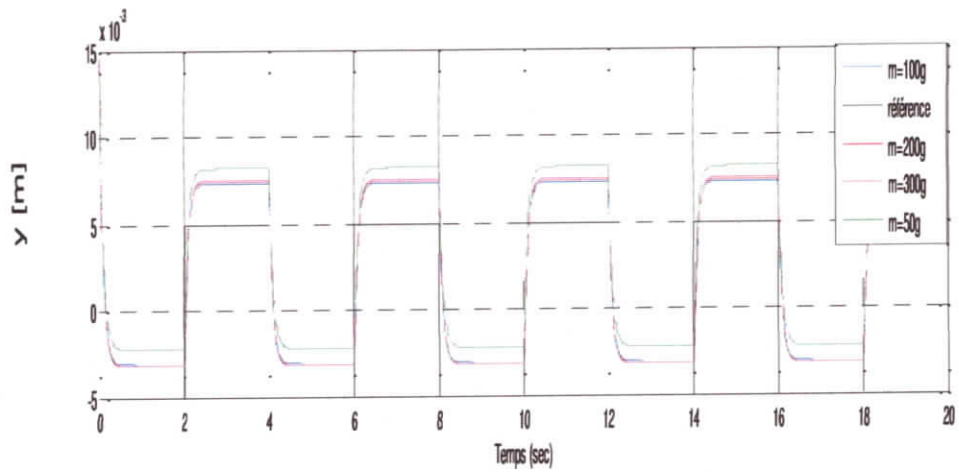


Figure III.20 : changement de valeur de masse de la balle

➤ **Changement de valeur de la résistance**

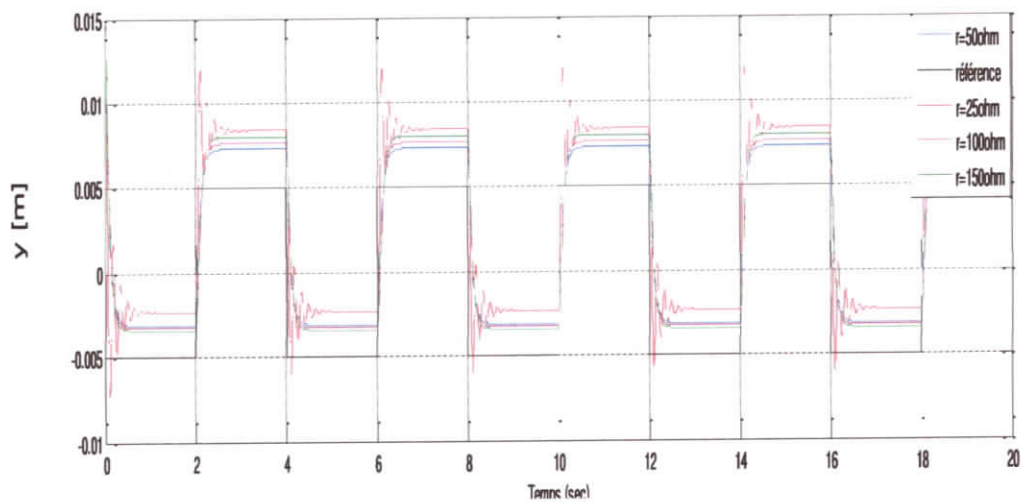


Figure III.21: changement de valeur de la résistance

➤ **Changement de valeur de la bobine (L)**

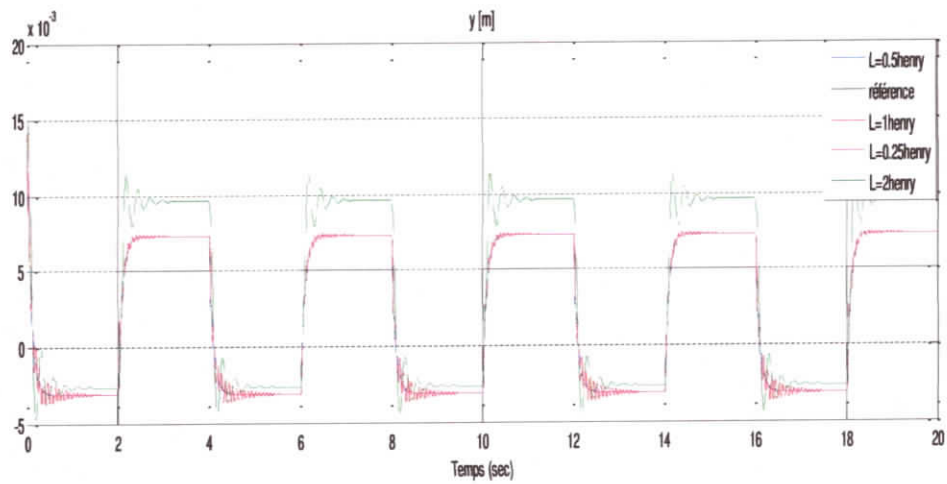


Figure III.22 : changement de valeur de la bobine

➤ **Changement de conditions initiales**

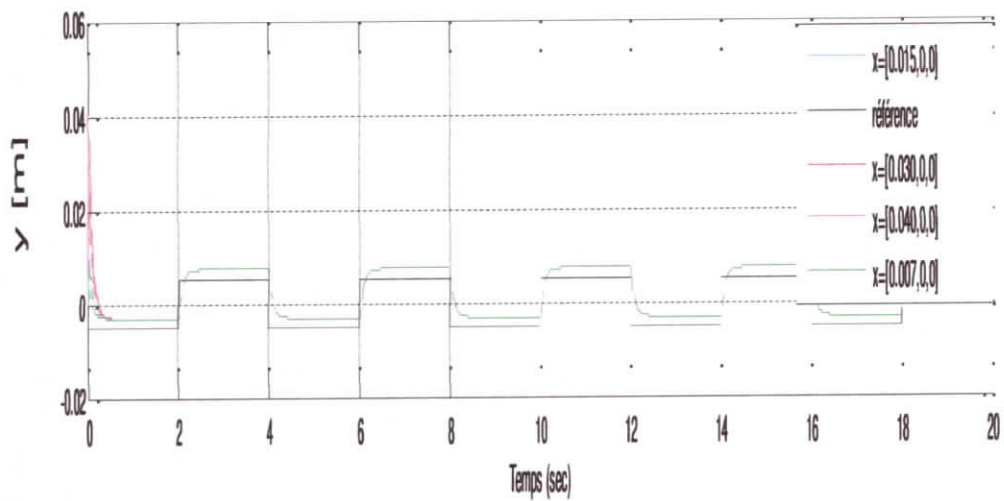


Figure III.23: changement de conditions initiales

D'après les figures, on peut conclure que :

- ✓ le changement de la masse et les conditions initiales n'influent pas sur les réponses du système. Par contre le changement de valeur de la résistance et de la bobine se traduit par une légère divergence du système par rapport à la position de référence. En effet le changement de valeur de la résistance et de la bobine influe très peu sur la tension et donc sur la position de la balle. Le contrôleur optimisé par le recuit simulé a pu éliminer le problème de divergence qu'on a vu lors l'utilisation de la recherche taboue.
- ✓ la stratégie proposée de conception accomplit, efficacement, les performances désirées.

III.5 Conclusion

Après l'utilisation des deux méthodes de recherche locales: la recherche taboue et le recuit simulé pour l'optimisation du contrôleur flou d'un système de lévitation magnétique; on peut constater que le recuit simulé donne de meilleurs résultats en termes de convergence et de stabilité de système. En effet cette méthode peut contrôler les paramètres du contrôleur lors du changement de la valeur de la résistance et de la bobine magnétique et garantie une optimisation du contrôleur flou pour la stabilisation de la balle dans un minimum temps de réponse et une bonne précision.



Conclusion Générale

Conclusion Générale

Le travail présenté dans ce mémoire concerne la commande 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 le recuit simulé et la recherche tabou.

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 des méthodes d'optimisation des caractéristiques d'un contrôleur flou basée sur l'utilisation du recuit simulé et de recherche tabou.

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.

Le deuxième chapitre met l'accent sur les différentes méthodes méta heuristiques, parmi lesquelles, on a présenté la méthode stochastique la plus prometteuse : le recuit simulé et la recherche tabou. On a décrit l'algorithme de calcul du recuit simulé et de recherche tabou qui sont base sur la notion de voisinage, Pour trouver l'optimum global d'un problème donné.

Le dernier chapitre est divisé en deux parties : Dans la première partie, une commande de lévitation magnétique par logique floue a été synthétisée. Dans la deuxième partie, les deux approches de recherche locale ont été testées et validées pour la stabilisation d'une balle magnétique.

Les deux stratégies de commande optimisée par les deux algorithmes RT et RS en poursuite de trajectoire, a effectivement assuré la stabilité et la robustesse du système commandé.

Dans le cadre de développement futur de ce travail, il est intéressant d'élaborer une loi de commande optimisée par ces techniques d'optimisation pour la commande multi variables des systèmes dynamiques en prenant en considération :

- ☞ Les non linéarités,
- ☞ Le couplage croisé dans le cas des systèmes multi variables,
- ☞ Les perturbations externes qui peuvent affectées le système.



Bibliographie

BIBLIOGRAPHIE

- [1] J.P.Corriou, « Commande des Procédés », édition Lavoisier, Paris 1996.
- [2] N.Talbi, « Conception des Systèmes D'Inférence Floue par des Approches Hybrides : Application pour la Commande et la Modélisation des Systèmes Non linéaires », Doctorat en Electronique, Université de Constantine 1, 2014.
- [3] H.Bouraoui, « Optimisation d'un Contrôleur Flou par les Algorithmes Génétiques : Application à Commande d'un Pendule Inversé », Master II en Electronique, Université de Jijel, 2010.
- [4] S.Ambapour, « Théorie des Ensembles Flous : Application à La Mesure De La Pauvreté Au Congo », Bureau D'Application Des Méthodes Statistiques Et Informatiques, 2009.
- [5] E.H.Mamdani, « Application of fuzzy Algorithms for Control of Simple Dynamic Plant », Proceedings of IEEE, vol 121, n°12, pp 1585-1588, 1974.
- [6] H.B.Verbruggen, H.J Ziimmermann, R.Babůska, « *Fuzzy algorithms for control* », International Séries in Intelligent Technologies, Kluwer Academic publisher, 1999.
- [7] A.F.Megri, « Programmation Dynamique Floue Appliquée à la Commande des Systèmes non Linéaires », Thèse De Doctorat En Electronique, Université Mentouri De Constantine, 2008.
- [8] N.Bentoubal, « Etude de L'Optimisation D'un Contrôleur Flou par Le Recuit Simulé : Application à La Commande D'un Processus dynamique », Master II en Electronique, Université de Jijel, 2011.
- [9] C.H.Papadimitriou, K.Steiglitz, *Combinatorial optimization – algorithms and complexity*. Prentice Hall, 1982.
- [10] A.Layeb, Utilisation des Approches d'Optimisation Combinatoire pour La Vérification des Applications Temps Réel. Thèse de Doctorat, Université Mentouri de Constantine 2010.

BIBLIOGRAPHIE

- [11] V.Angel, « La Rugosité des paysages : une théorie pour la difficulté des problèmes d'optimisation combinatoire relativement aux métaheuristiques », Thèse De Doctorat De L'université de Paris-Sud, Orsay, 1998.
- [12] Wei WU. Synthèse d'un Contrôleur flou par Algorithme Génétique : Application au réglage dynamique des paramètres d'un système, thèse de doctorat de l'Université de Lille1, 1998.
- [13] D.Corne, M.Dorigo and F.Glover, « New Ideas in Optimization », *McGraw-Hill*, 1999.
- [14] M.E.Marmion, « Recherche Locale et Optimisation Combinatoire : De l'analyse structurelle d'un problème à la conception d'algorithmes efficaces », Thèse de Doctorat en l'informatique, Université lille1, 2011.
- [15] S.Le Digabel, « Introduction aux Métaheuristiques », Ecole Polytechnique De Montréal, 2014.
- [16] S.M.Douiri, S.Elbernoussi, H.Lakhabab, « Cours des Méthodes de Résolution Exactes Heuristiques et Métaheuristiques », Master Codes, Cryptographie et Sécurité de L'information, Université Mohammed V, Faculté des Sciences de Rabat.
- [17] H.Omessaad: Contribution au développement de méthodes d'optimisation Stochastiques application à la conception des dispositifs électrotechniques, Thèse de Doctorat, Université De Lille France 2003.
- [18] A.Bendahman, « Le Recuit Simulé », Master II RFIA En Informatique, Université Des Sciences et de la Technologie D'Oran, 2011.
- [19] : E.Aarts, J.Korst, « Similated Annealing Boltzmann machines », John Wiley&Sons, New York, 1990.
- [20] : N.Metropolis, A.Rosebluth, M.Rosebluth, A.Teller, E.Teller, "Equation of state calculations by fast computing machines", *Journal Chem. Phys.*, Vol. 27, No. 1087, pp.1092, 1953.
- [21] : Tierney, L.Markov chains for exploring posterior distributions, « *The annals of Statistics* », Vol.22, No.4, pp 1701-1762.1994.

BIBLIOGRAPHIE

- [22] : D.Vanderbilt, S.G.Louie, « A Monte Carlo Simulated Annealing Approach to optimisation over Continuous Variables », *Journal of computational, Physics*, 56, PP. 259-271, 1984.
- [23] S.M.Sait et H.Youssef; *iterative computer algorithms with applications in engineering: solving combinatorial optimization problems*, IEEE computers society, 1999.
- [24] F.Glover, « Tabu search - part I », *ORSA Journal on Computing*, 1(3), 1989, pp. 190–206.
- [25] N.Bouchikhi, « La Recherche Tabou », Master(RFIA) En Informatique, Université Des Sciences Et De La Technologie D'ORAN Mohamed Boudiaf USTO, 2013.
- [26] Article Lapetoule de J.Ayas & M.André Viau « la recherche tabou ».
- [27] S.Meghriche, F.Bounaas, « Conception de la Base de Règles Floues en Utilisant L'Algorithme d'optimisation Par Essaim Particulaire », Master II en Electronique, Université de Jijel, 2015.
- [28] K.M.Passino, S.Yurkovich, "Fuzzy control", edition Wesley Longman, pp. 365-388, 1998.
- [29] T.Takagi, M.Sugeno, "Fuzzy identification of systemes and Its applications to modeling and control", *IEEE Transactions on systems Man and Cybernetics*, vol 15, No 1, pp 116-132,1985.
- [30] P.Borne, J.Rozinoer, J.Y.Dieulot, L.Dubois, « Introduction à la Commande Floue », éditions TECHNIP, pp 2-24, 1998.
- [31] B.Bouchon-Meunier, « La logique floue », collection : « que sais-je ? », Presse Universitaire de France, 1993.
- [32] H.Buhler, « Réglage par logique floue », Presses polytechniques et universitaires ROMANDES, pp 32-65, 1994.
- [33] C.C.Lee, "Fuzzy logic in control systems : fuzzy logic controller. Part 1", *IEE Transaction on systems Man and Cybernetics*, vol 20, pp 404-418. 1990.
- [34] C.C.Lee, "Fuzzy logic in control systems : fuzzy logic controller. Part 2", *IEE Transaction on systems Man and Cybernetics, Part B*, vol 20, pp 419-435. 1990.

BIBLIOGRAPHIE

- [35] A.Chatouane, « Optimisation d'un système d'inférence floue par la recherche Tabou », Master II en Electronique, université de Jijel, 2013.
- [36] J.Godjevac, « Idées nettes sur la logique flou », presses polytechniques et universitaires romandes.1999
- [37] E.Lutton, « Darwinisme artificiel : une vue d'ensemble », INRIA-Rocquencourt-Equipe COMPLEX-FRACTALES, B.P.105, 78153 LE CHESNAY Cedex, France, Fév.2004.
- [38] Jantzen, J. Foundations of Fuzzy Control. *John Wiley & Sons*, 2007.
- [39] D.Mokeddem, « Contrôle Flou des Processus Biotechnologiques à Base d'Algorithmes Génétiques », Thèse de Doctorat En sciences en Electronique, Université Ferhat Abbas de Setif, 2011.
- [40] C.C.Ribeiro, N.Maculan (Eds.), Applications of combinatorial optimization. *Annals of Operations Research* 50, 1994.
- [41] S.Lin, B.W.Kernighan, An efficient heuristic for the traveling-salesman problem. *Operations Research* 21 : 498-516, 1973.
- [42] C.R.Reeves (Ed.) *Modern heuristic techniques for combinatorial problems*, Blackwell Scientific Publications, Oxford, 1993.
- [43] E.H.L.Aarts, J.K. Lenstra (Eds.), *Local search in combinatorial optimization*, John Wiley & Sons, 1997.
- [44] G.Laporte, I.H.Osman, *Metaheuristics in combinatorial optimization*, *Annals of Operations Research* 63, J.C. Baltzer Science Publishers, Basel, Switzerland, 1996.
- [45] I.H.Osman, J.P.Kelly (Eds.), *Meta-heuristics: theory and applications*, Kluwers Academic Publishers, Boston, 1996.