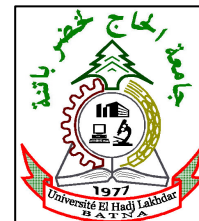


République Algérienne Démocratique et populaire

Ministère de l'Enseignement Supérieur et de la Recherche scientifique



UNIVERSITE DE BATNA
FACULTE DE TECHNOLOGIE
Département d'Electronique



Mémoire

Présenté en Vue de l'Obtention de Diplôme de

Magistère en Electronique

Option : Contrôle Industriel

Par : BERKANI Abdelhakim

Thème :

**Métaheuristique Hybride Réseaux de Neurones
Artificiels-PSO du Recuit Simulé pour la
Commande d'un Procédé Industriel Non-linéaire**

Membres de Jury Proposés:

- **Dr Khireddine Mohamed Salah**, Département d'Electronique, Université de Batna : Président.
- **Dr Arrouf Mohamed**, Département d'Electrotechnique de l'Université de Batna : Examineur.
- **Dr Abdessemed Yassine**, Département d'Electronique, Université de Batna : Rapporteur.
- **Dr Messai Abderraouf**, Département d'Electronique, Université de Constantine : Examineur.
- **Dr Athamena Noureddine**, Département d'Electronique, Université de Batna : Examineur.



2012/2013

Remerciements

Je tiens à remercier mon encadreur Monsieur Dr Abdessemed Yassine pour avoir voulu diriger ce travail. Sans ses conseils et sa rigueur ce travail n'aurait jamais vu le jour, réellement. Il m'a été un grand soutien.

Je tiens à remercier les membres de jury d'avoir accepté d'examiner ce modeste travail de magistère.

Je remercie aussi tout le corps enseignant et administratif du département d'électronique de Batna qui a contribué à ma formation universitaire.

Sans oublier tous mes amis.

Berkani Abdelhakim

Dédicaces

Je dédie ce modeste travail :

A mes très chers parents

A ma petite famille

A tous mes amis.

Et à tous ceux qui ont contribué de loin ou de près à ma formation

Mr: BERKANI Abdelhakim

Table des matières

Dédicace	i
Remerciements	ii
Table des matières	iii
Introduction générale	1
Chapitre I : Généralité sur les métaheuristiques	
I.1 Introduction	5
I.2 Vue générale	5
I.3 Notions fondamentales.....	6
I.3.1 Définition	6
I.3.2 Terminologies.....	7
I.3.3 Présentation des heuristiques.....	7
I.3.4 Présentation des métaheuristiques	8
I.3.5 Fonctionnement général des métaheuristiques	9
I.3.5.1 Diversification.....	9
I.3.5.2 Amélioration	10
I.3.5.3 Mise à jour des références	10
I.3.5.4 Génération de sous-ensemble.....	10
I.3.5.5 Combinaison Solution	10
I.4 Classification des métaheuristiques	12
I.4.1 Métaheuristique à parcours	12
I.4.2. Métaheuristique à population	12
I.4.3 Métaheuristiques à méthodes implicites.....	12
I.4.4 Métaheuristiques a méthodes explicites.....	12
I.5 Les méthodes métaheuristiques modernes	13
I.6 Le recuit simulé	14
I.6.1 Principe de base.....	14
I.6.2 Algorithme	15
I.6.3 Avantages	17
I.6.4 Inconvénients.....	17
I.6.5 Pseudocode.....	18
I.7 Algorithmes génétiques.....	19
I.7.1 Principe de base.....	19
I.7.2 Avantages.....	20
I.7.3 Inconvénients	21
I.8 La méthode recherche Tabou	22
I.8.1 Principe de la méthode.....	23
I.8.2 Principe de l'algorithme	24
I.8.3 Le critère d'arrêt.....	23
I.8.4 Avantages.....	25
I.8.5 Inconvénients	25
I.9 Conclusion.....	26

Chapitre II : Les réseaux de neurones artificiels

II.1 Introduction	29
II.2 Utilité des RNA	29
II.3 Modèle d'un neurone biologique	30
II.4 Modèle d'un neurone artificiel.....	31
II.4.1 Modélisation d'un neurone artificiel	31
II.4.2 Fonctionnement d'un RNA.....	32
II.4.2.1 Fonction d'activation	33
II.4.2.2 Fonction de combinaison.....	35
II.5 Les propriétés d'un réseau de neurones.....	35
II.5.1 La propriété d'approximation universelle.....	35
II.5.2 La propriété de parcimonie	35
II.6 Topologie des réseaux de neurones.....	36
II.6.1 Les réseaux statiques ou réseau à couche (FEED FORWARD)	36
II.6.2 Les réseaux dynamiques (récurrents)	38
II.7 Le Perceptron	39
II.7.1 Rétine	40
II.7.2 Couche de cellules d'association	40
II.7.3 Couche de cellule de décision	40
II.8 Perceptron simple	41
II.9 Réseaux bouclés	42
II.10 Identification et commande par réseaux de neurones.....	43
II.11 Apprentissage dans les réseaux de neurones.....	44
II.11.1 Identification par réseaux de neurones	44
II.11.2 Identification des processus par réseaux de neurones.	45
II.12 Avantages et Inconvénients des réseaux de neurones	45
II.7 Conclusion	46

Chapitre III : La commande des processus non-linéaires par les techniques RNA – PSO et SA

III.1 Introduction.....	48
III.2 Notions de base	48
III.3 Méthodologie de conception.....	49
III.3.1 Analyse du cahier des charges	49
III.3.2 Formulation du problème d'optimisation	50
III.3.3 Résolution du problème d'optimisation	51
III.3.4 Analyse et exploitation des résultats	52
III.4 Caractéristiques	53
III.4.1 Sensibilité et robustesse d'une méthode d'optimisation	53
III.4.2 Opérateurs de recherche fondamentaux	53
III.4.3 Mode de recherche de l'optimum	54
III.4.4 Ordre d'une méthode de résolution.....	54
III.5 Processus d'optimisation	55
III.5.1 Variables du problème.....	55
III.5.2 Espace de recherche	55
III.5.3 Fonction objectif	55

III.5.4 Problème d'optimisation.....	56
III.6 Optimisation par essaim de particules.....	56
III.6.1 Méthode de base.....	57
III.6.2 Formulation générale.....	58
III.6.3 Algorithme de base.....	59
III.6.4 Paramètres de l'algorithme	59
III.6.5 Principe.....	61
III.7 Commande des processus par réseaux de neurones.....	61
III.7.1 Apprentissage d'un contrôleur conventionnel	61
III.8 Algorithme de recuit simulé	62
III.9 Hybridation PSO-SA.....	64
III.9.1 La description d'une particule	64
III.9.2 Swarm particules initiale	64
III.9.3 Evaluer la fonction	65
III.9.4 L'algorithme SA	65
III.10 Conclusion	67
 Chapitre IV : Simulation de la commande du système de Lévitiation Magnétique	
IV.1 Introduction	69
IV.2 Etude du système de lévitation de la balle en acier	70
IV.3 Etude du système de lévitation magnétique du train Magelv.....	74
IV.3.1 Introduction	74
IV.3.2 HISTOIRE DU MAGLEV	74
IV.3.3 Fonctionnement du train à Lévitiation magnétique MAGLEV	75
IV.3.4 Les parties principales	76
IV.3.4 Le système de suspension	76
IV.3.6 Le système de contrôle	77
IV.3.7 Système de rétroaction	77
IV.3.8 Système de propulsion.....	77
IV.3.9 Le système d'orientation.....	78
IV.4 Le système de contrôle du train en lévitation magnétique	78
IV.4.1 Analyse du circuit électrique	78
IV.4.2 l'analyse par fonction de transfert.....	79
IV.4.3 Modélisation du système par la loi de physique	81
IV.5 Résultats de simulation sous Matlab-Simulink de la commande de Maglev.....	83
IV.5.1 Données du Système Expérimental	83
IV.4.2 Modèle simplifié de la commande sous simulink	84
IV.6 Résultats de simulation et interprétation	86
IV.7 Conclusion.....	91
Conclusion générale.....	93
Références bibliographiques	95
Annexe	98

INTRODUCTION GENERALE

Les métaheuristiques sont souvent inspirées à partir des systèmes naturels, qu'ils soient pris en physique (cas du recuit simulé), en biologie de l'évolution (cas des algorithmes génétique), ou encore en éthologie (cas des algorithmes de colonie de fourmis ou de l'optimisation par essaim de particules).

La majorité des problèmes d'extraction de connaissances peuvent s'exprimer comme des problèmes d'optimisation combinatoire. Or, de nombreux problèmes d'optimisation combinatoire sont difficiles, ne pourront donc pas être résolus de manière exacte dans un temps "raisonnable" puisque la capacité de calcul des machines évolue linéairement alors que le temps nécessaire à la résolution de ces problèmes évolue exponentiellement. Lorsqu'on s'attaque à des problèmes réels, il faut se résoudre à un compromis entre la qualité des solutions obtenues et le temps de calcul utilisé.

Les méthodes d'optimisation recherchent une solution, ou un ensemble de solutions, dans l'espace de recherche, qui satisfont l'ensemble des contraintes et qui minimisent, ou maximisent, la fonction objective. Parmi ces méthodes, les métaheuristiques sont des algorithmes génériques d'optimisation : leur but est de permettre la résolution d'une large gamme de problèmes différents, sans nécessiter de changements profonds dans l'algorithme. Elles forment une famille d'algorithmes visant à résoudre des problèmes d'optimisation difficile, pour lesquels on ne connaît pas de méthode classique plus efficace.

L'optimisation par essaim de particules (Particle Swarm Optimization) ou 'PSO', est issue d'une analogie avec les comportements collectifs de déplacements d'animaux. La métaphore a de plus été largement enrichie de notions de socio-psychologie. Chez certains groupes d'animaux, comme les bancs de poissons, on peut observer des dynamiques de déplacements relativement complexes, alors que les individus eux-mêmes n'ont accès qu'à des informations limitées, comme la position et la vitesse de leurs plus proches voisins.

A chaque itération, la particule se déplace. La socio-psychologie suggère que des individus se déplaçant sont influencés par leur comportement passé et par celui de leurs voisins. On tient donc compte dans la mise à jour de la position de chaque particule, de la direction de son mouvement, sa vitesse sa meilleure position et la meilleur position de ses voisins. [1]

La construction d'une métaheuristique hybride est motivée par la nécessité de parvenir à un compromis acceptable entre l'intensification et la diversification de recherche.

OBJECTIF DU MEMOIRE

Le présent mémoire est consacré à l'étude et la simulation d'une commande d'un système physique non-linéaire, à savoir un système de lévitation magnétique d'un train de locomotion qui est composé essentiellement composé d'un électro-aimant et d'un rail en fer. On fait d'abord une étude plus simple sur une balle en acier qui est mise en suspension dans l'air grâce à un champ magnétique créée dans une bobine parcourue par un courant (électro-aimant), par un contrôleur basé sur un réseaux de neurones artificiels (**RNA**) dont l'adaptation des poids suivant les conditions de travail se fait en utilisant conjointement les métaheuristicques PSO et celle du recuit simulé.

Cette commande est comparée à une régulation PID standard : élimination des dépassements, poursuite parfaite de la référence, élargissement de la plage de stabilité. La stratégie proposée est mise en œuvre en temps réel en utilisant l'environnement logiciel Matlab-Simulink, qui permet une conception rapide et modulaire et un changement en temps réel des paramètres et des consignes. [2] [3] [5]

ORGANISATION MEMOIRE

Notre mémoire est organisé comme suit :

Le premier chapitre est consacré à la présentation des métaheuristicques utilisés dans notre étude : Les techniques PSO et recuit simulé sont revues en montrant leurs

Intérêts actuels considérables actuels dans la commande des systèmes, surtout lorsqu'il est question d'optimiser un ou plusieurs paramètres d'un système de commande non-linéaire dont l'utilité industriel a été prouvée et qui est de plus en plus croissante.

Dans le deuxième chapitre, une présentation des réseaux de neurones artificiels (RNA) est effectuée en rappelant leur principe, les différents modèles neuronaux existants la technique de commande intelligente classique par les réseaux de neurones artificiels est présentée.

Le troisième chapitre présente les heuristiques de l'optimisation par essaim de particules 'PSO', les différentes stratégies de commande neuronale. Une stratégie de contrôle neuronale par anticipation est appliquée à un système non-linéaire.

Le quatrième chapitre est consacré simulation d'une commande d'un système industriel non-linéaire, à savoir un système de lévitation magnétique d'un train de locomotion. On utilise procède a fait d'abord une étude plus simple sur une balle en acier qui est mise en suspension dans un contrôleur basé sur un réseaux de neurones artificiels (RNA) dont l'adaptation des poids suivant les présences de perturbations en appliquant conjointement les métaheuristiques PSO et celle du recuit simule. Les résultats de simulations obtenus sont ensuite comparés à ceux d'une régulation PID standard : élimination des dépassements, poursuite parfaite de la référence, élargissement de la plage de stabilité. La stratégie proposée est mise en œuvre en temps réel en utilisant l'environnement logiciel **Matlab-Simulink**, qui permet une conception rapide et modulaire et un changement en temps réel des paramètres et des consignes.

Finalement le mémoire se termine par une conclusion générale dont laquelle on présente les perspectives et des suggestions pour améliorer dans le futur ce travail de recherche.

CHAPITRE I

GENERALITES SUR LES METAHEURISTIQUES

I.1 Introduction

Nous présentons dans ce chapitre le cadre générale des métaheuristiques.

Les premières métaheuristiques datent des années 1980, et bien qu'elles soient d'origine discrète, on peut les adapter à des problèmes continus. Elles sont utilisées généralement quand les méthodes classiques ont échoué, et sont d'une efficacité non garantie. Le terme métaheuristique est utilisé par opposition aux heuristiques particulières pour un problème donné. Les métaheuristiques peuvent être utilisées pour plusieurs types de problèmes, tandis qu'une heuristique est adaptée à un problème donné. Les métaheuristiques ont également comme caractéristiques communes leur caractère stochastique, ainsi que leur inspiration, une analogie avec d'autres domaines de recherche (la biologie, la physique, etc.).

Les métaheuristiques ne sont pas des méthodes figées; il n'y a pas de relation d'ordre quant à l'efficacité d'un algorithme ou d'un autre, cela dépend plutôt des paramètres utilisés, de l'application elle-même ou du problème.

I.2 Vue générale

Les ingénieurs, se heurtent quotidiennement, quel que soit leur secteur d'activité, à des problèmes d'optimisation. Il peut s'agir de minimiser un coût de production, d'optimiser le parcours d'un véhicule ou le rendement d'un portefeuille boursier, de rationaliser l'utilisation de ressources, d'améliorer les performances d'un circuit électronique, de fournir une aide à la décision à des managers, etc. Cet ouvrage présente une famille de techniques d'optimisation, appelées "métaheuristiques", adaptées à la résolution de problèmes pour lesquels il est difficile de trouver un optimum global ou de bons optimums locaux par des méthodes plus classiques.

I.3 Notions fondamentales

I.3.1 Définition

L'arrivée des métaheuristiques marque une réconciliation des deux domaines. En effet, celles-ci s'appliquent toutes sortes de problèmes combinatoires et elles peuvent s'adapter aussi aux problèmes continus.

Ces méthodes ont en commun, en outre, les caractéristiques suivantes:

Elles sont, au moins pour partie, stochastiques ; cette approche permet de faire face à l'explosion combinatoire des possibilités ; d'origine combinatoire, elles ont l'avantage, décisif dans le cas continu, d'être directes, c'est-à-dire qu'elles ne recourent pas au calcul, souvent problématique, des gradients de la fonction objectif ; elles sont inspirées par des analogies ; avec la physique (recuit simulé, diffusion simulé), avec la biologie (algorithmes génétiques, recherche tabou) ou avec l'éthologie (colonies de fourmis, essaims particulaires) ; elles partagent aussi les mêmes inconvénients : les difficultés de réglage des paramètres de la méthode, et le temps de calcul élevé [1].

Ces méthodes ne s'excluent pas mutuellement. En effet, dans l'état actuel de la recherche, il est le plus souvent impossible de prévoir avec certitude l'efficacité d'une méthode donnée, quand elle est appliquée à un problème donné. De plus, la tendance actuelle est l'émergence des méthodes hybrides, qui s'efforcent de tirer parti des avantages spécifiques des approches différentes en les combinant.

Le réglage et la comparaison des métaheuristiques sont souvent effectués empiriquement, en exploitant des jeux de fonctions analytiques de test, dont les minimums globaux et locaux sont connus.

Les métaheuristiques sont généralement des algorithmes stochastiques itératifs, qui progressent vers un optimum global, c'est-à-dire l'extremum global d'une fonction, par échantillonnage d'une fonction objectif. Elles se comportent comme des algorithmes de recherche, tentant d'apprendre les caractéristiques d'un problème afin d'en trouver une approximation de la meilleure solution (d'une manière proche des algorithmes d'approximation).

I.3.2 Terminologies

On parle de méta, du grec « au-delà » comprendre ici « à un plus haut niveau », heuristique, du grec qui signifie « trouver ». En effet, ces algorithmes se veulent des méthodes génériques pouvant optimiser une large gamme de problèmes différents, sans nécessiter de changements profonds dans l'algorithme employé.

I.3.3 Présentation des heuristiques

Par définitions les heuristique sont des règles utilisent les résultats passes et leurs analogies afin d'optimiser leurs recherches futures en examinant d'abord les cas les plus possible. Les heuristiques dispose d'une simplicité et d'une rapidité dans leur exécution que les algorithmes classique.

Bien que l'obtention d'une solution optimale ne soit pas garantie, l'utilisation d'une méthode heuristique offre de multiples avantages par rapport à une méthode exacte :

La recherche d'une solution optimale peut être totalement

Inappropriée dans certaines applications pratiques en raison de la dimension du problème, de la dynamique qui caractérise l'environnement de travail, du manque de précision dans la récolte des données, de la difficulté de formuler les contraintes en termes explicites ou de la présence d'objectifs contradictoires.

Lorsqu'elle est applicable, une méthode exacte est souvent beaucoup plus lente qu'une méthode heuristique, ce qui engendre des coûts informatiques supplémentaires et des difficultés au niveau du temps de réponse.

Les principes de recherche qui sont à la base d'une méthode heuristique sont en général plus accessibles aux utilisateurs non expérimentés. Le manque de transparence qui caractérise certaines Méthodes exactes Nécessite une intervention régulière de la part d'un spécialiste voire même du concepteur de la méthode.

Une méthode heuristique peut être facilement adaptée ou combinée avec d'autres types de méthodes. Cette flexibilité augmente considérablement les possibilités d'utilisation des méthodes heuristiques.

Une heuristique d'optimisation est une méthode approchée se voulant simple, rapide

et adaptée à un problème donne. Sa capacité à optimiser un problème avec un minimum d'informations est contrebalancée par le fait qu'elle n'offre aucune garantie quant à l'optimalité de la meilleure solution trouvée.

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

I.3.4 Présentation des métaheuristiques

Les métaheuristiques sont des algorithmes d'optimisation de type stochastique et progressant vers un optimum par échantillonnage d'une fonction objectif dont le but est la résolution de problèmes d'optimisation difficile.

Les métaheuristiques marquent une réconciliation de domaines: en effet, celles-ci s'appliquent à toutes sortes de problèmes discrets, et elles peuvent s'adapter aussi aux problèmes continus.

Ces méthodes ont en commun, en outre, les caractéristiques suivantes :

- elles sont, au moins pour partie, stochastiques: cette approche permet de faire face à l'explosion combinatoire des possibilités;

- souvent d'origine discrète, elles ont l'avantage, décisif dans le cas continu, d'être directes, c'est-à-dire qu'elles ne recourent pas au calcul, souvent problématique, des gradients de la fonction objectif;

- Elles sont inspirées par des analogies: avec la physique (recuit simulé, diffusion simulée...), avec la biologie (algorithmes évolutionnaires, recherche tabou...) ou avec l'éthologie (colonies de fourmis...);

- Elles partagent aussi les mêmes inconvénients: les difficultés de réglage des paramètres de la méthode, et le temps de calcul élevé.

L'utilisation du mot métaheuristique signifie les algorithmes regroupent en réalité plusieurs heuristiques. Cette utilisation implique qu'on ne peut pas réellement classer les métaheuristiques dans la catégorie des algorithmes d'intelligence artificielle, puisqu'ils

sont principalement guidés par le hasard. Cependant ils sont souvent combinés à d'autres algorithmes afin d'accélérer la convergence en obligeant entre autre le métaheuristique à ne pas prendre en compte les solutions trop (extravagantes).

Une notion à bien cerner découlant de l'utilisation d'heuristique est l'ensemble des solutions possibles ou espace global. Car bien que les métaheuristiques fonctionnent plus ou moins de façon hasardeuse cette tare est compensée par la diminution de l'espace de travail local à chaque itération [1].

I.3.5 Fonctionnement général des métaheuristiques

Les métaheuristiques ne nécessitent pas une connaissance particulière sur les problèmes d'optimisation à résoudre. Il suffit d'associer une ou plusieurs variables à une ou plusieurs solutions (optimum).

Les habitudes de dispersion de recherche métaheuristiques reprennent les cinq composantes suivantes : [2]

- (1) la diversification,
- (2) d'amélioration,
- (3) Référence set mise à jour,
- (4) la génération de sous-ensemble, et la combinaison de solutions.

(5) L'idée de base derrière la méthode de recherche de dispersion est de combiner les bonnes solutions qui sont découverts par la métaheuristique.

Les cinq composantes de la métaheuristique sont définies comme suit:

I.3.5.1 Diversification:

La composante diversification cherche à générer un ensemble de solutions diversifiées qui est représentatif de la recherche l'espace. Dans l'approche présentée dans cet article, la méthode de diversification consiste à générer de la population aléatoirement.

I.3.5.2 Amélioration

Le composant amélioration commence par la recherche d'un vecteur de paramètres qui réduit le nouveau courant la somme carrée moyenne des erreurs (MSE), généralement en effectuant une recherche locale autour d'une bonne solution. Les procédures de recherche locale sont le problème de dépendance.

I.3.5.3 Mise à jour des références

L'ensemble de référence est composé d'un sous-ensemble solution soumise à un processus évolutif dans un tenter de déterminer la solution optimale. Dans cet article, l'ensemble de référence est composé d'un petit ensemble de meilleures solutions trouvé pour la MSE la somme carrée moyenne des erreurs. Il s'agit d'une référence élitiste que défini dans les meilleures solutions sont maintenues. Soit R la référence ensemble, organisé selon les valeurs décroissantes non-MSE

$$MSE(X) = \frac{\sum_{t=1}^n (Y(t) - \hat{Y}(t))^2}{n} \quad (I.1)$$

I.3.5.4 Génération de sous-ensemble

L'étape de génération de sous-ensemble fonctionne de combiner des solutions de R pour créer des nouveaux sous-ensembles. Il est de nombreuses façons de combiner ces solutions, mais les résultats généralement bons peuvent être obtenus même en combinant un petit nombre des solutions.

I.3.5.5 Combinaison Solution

L'étape de combinaison encourage la combinaison des solutions contenues dans chaque sous-ensemble.

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

L'optimisation multi objectif (dites aussi multicritère) [2], où il faut optimiser plusieurs objectifs contradictoires. La recherche vise alors non pas à trouver un optimum global, mais un ensemble d'optima formant la surface de compromis du problème.

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

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

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

La parallélisation, où l'on cherche à accélérer la vitesse de l'optimisation en répartissant la charge de calcul sur des unités fonctionnant de concert. Le problème revient alors à adapter les métaheuristiques pour qu'elle soit distribuée.

L'hybridation, qui vise à tirer parti des avantages respectifs de métaheuristiques différentes en les combinant [3].

Les métaheuristiques sont inspirées par des systèmes naturels et utilisées dans de nombreux domaines :

En physique (recuit simulé), en biologie de l'évolution (algorithmes évolutionnaires) en éthologie (algorithmes de colonie de fourmis) [1].

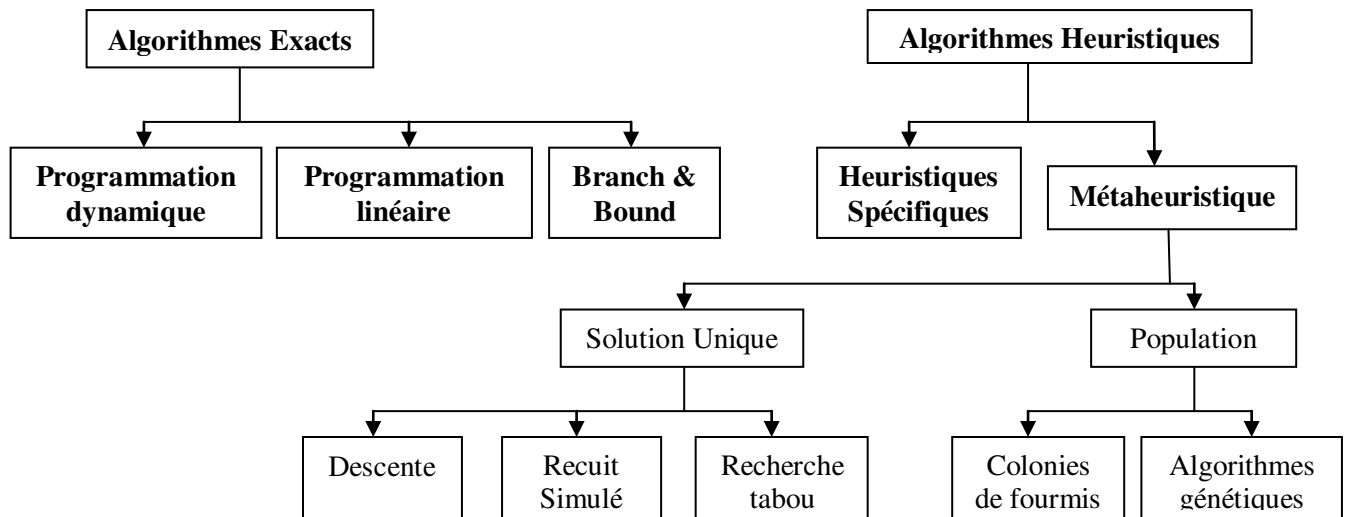


Figure I.1: Classification des méthodes d'optimisation combinatoire.

I.4 Classification des métaheuristiques

On peut classer les métaheuristiques selon leur fonctionnement :

I.4.1 Métaheuristique à parcours

Les métaheuristiques les plus classiques sont celles fondées sur la notion de parcours. Dans ce type d'algorithme, ce dernier fait évoluer une seule fonction objectif sur l'espace de recherche local à chaque itération puis la compare aux optimums. La compréhension de la notion de voisinage est alors nécessaire.

Les plus connues dans cette classe sont le recuit simulé et la recherche de tabous [1].

I.4.2. Métaheuristique à population

Dans cette famille, les métaheuristiques utilisent la notion de population :

Ils manipulent un ensemble de solutions en parallèle. Chaque élément de la population parcourt un certain nombre de solutions dans l'ensemble local. On peut les assimiler aux métaheuristiques à des méta-parcours.

Parmi les algorithmes inclus dans cette classification, on peut citer les algorithmes génétiques et les algorithmes de colonie de fourmis. [1]

I.4.3 Métaheuristiques à méthodes implicites

Lors de l'utilisation des méthodes implicites, avec les algorithmes génétiques, la distribution de probabilité n'est pas connue ou n'est pas utilisée. Le choix de l'échantillonnage entre deux itérations ne suit pas une loi donnée, mais est fonction de règles locales. [3]

I.4.4 Métaheuristiques à méthodes explicites

Ces méthodes utilisent une distribution de probabilité choisie à chaque itération. C'est le cas des algorithmes à estimation de distribution, comme leur nom l'indique, estiment à chacune de leur itération et via une distribution de probabilité l'espace de recherche local optimal [3].

I.5 Les méthodes métaheuristiques modernes

Ces sont méthodes métaheuristiques modernes les plus connues sont :

- Les algorithmes évolutionnistes, parmi lesquels :
 - les stratégies d'évolution,
 - les algorithmes génétiques,
 - les algorithmes à évolution différentielle,
 - les algorithmes à estimation de distribution,
 - les systèmes immunitaires artificiels,
 - la recombinaison de chemin (Path relinking en anglais)
- le recuit simulé,
- les algorithmes de colonies de fourmis,
- Les algorithmes d'optimisation par essais particuliers,
- la recherche avec tabous,
- la méthode GRASP.

Il existe un très grand nombre d'autres métaheuristiques, plus ou moins connues :

- l'algorithme du kangourou,
- la méthode de Fletcher et Powell,
- la méthode du bruitage,
- la tunnelisation stochastique,
- l'escalade de collines à recommencements aléatoires,
- la méthode de l'entropie croisée.

Nous allons nous intéresser aux principales méthodes métaheuristiques modernes :

I.6 Le recuit simulé

I.6.1 Principe de base

Le recuit simulé est une métaheuristique probabiliste permettant d'approximer l'optimum global d'une fonction objectif. Cette technique est souvent utilisée lorsque le calcul de la solution optimale exacte demanderait un temps de calcul trop important, le recuit simulé est alors utilisé pour trouver une solution approchée dans un temps raisonnable.

Le nom et le principe de cette métaheuristique sont inspirés directement de la technique du recuit en métallurgie. Le recuit permet de diminuer les défauts d'un matériau en augmentant sa température puis en contrôlant le refroidissement par un apport externe de chaleur. Un refroidissement naturel ne permet pas d'atteindre la configuration optimale des atomes pour obtenir la meilleure stabilité. De manière analogue, pour le recuit simulé, nous parlerons de l'énergie du système, nous cherchons à atteindre la configuration du système qui possède l'énergie la plus faible.

A chaque itération du recuit simulé, la solution actuelle est remplacée par une autre solution issue du voisinage de la solution initiale. Cette nouvelle solution est acceptée avec une probabilité dépendant à la fois de la différence entre les valeurs de cout des deux solutions et d'un paramètre T appelé température. Cette température décroît progressivement au cours du temps selon une loi déterminée préalablement. Il y a donc toujours une probabilité de dégrader la solution lors d'une itération. Cette stratégie permet d'éviter de rester coincer dans un minimum local.

Le recuit simulé est une technique générique qui peut être appliquée aussi bien à des problèmes discrets que des problèmes continus. Généralement, le recuit simulé est plus souvent utilisé dans des cas discrets, on peut par exemple citer également le problème du voyageur de commerce ou encore le problème du flow shop [6].

1.6.2 Algorithme

A chaque itération de l'algorithme, le recuit simulé examine un état voisin s' de l'état courants et détermine de façon stochastique si le système doit effectuer la transition vers cet état ou rester dans la configuration actuelle.

Ce processus amène progressivement le système vers une configuration de plus faible énergie. Cette étape est répétée jusqu'à que le système soit figé sur la même solution pendant plusieurs itérations ou lorsque qu'une limite d'itérations a été dépassée.

Le voisinage d'un état est généré en effectuant des modifications élémentaires de la configuration du système. Pour obtenir de bons résultats il est indispensable que ces modifications soient les plus simples possibles, les deux configurations doivent rester proches. Par exemple dans le cas du voyageur de commerce, une modification élémentaire est la permutation de deux villes adjacentes. L'exploration du voisinage est l'étape la plus importante du recuit, elle doit se faire de façon optimisée.

La probabilité de transition depuis un état s vers un état s' est donnée par une fonction d'acceptation P Avec T la température du système, et l'énergie de l'état initial été l'énergie de l'état voisin. Cette fonction renvoie alors la probabilité de transition vers cet état. Généralement, si la nouvelle solution est meilleure, la probabilité de transition est 1, mais ce n'est pas toujours le cas selon les versions de l'algorithme. Si l'énergie de l'état voisin est plus forte, on utilise la règle de Metropolis-Hastings, la probabilité de transition est alors:

$$\Delta E = e^{-T} \quad (I.2)$$

ΔE est la différence d'énergie entre les deux états. Initialement T est très élevé donc l'algorithme va souvent dégrader la solution. Au fil des itérations, T décroît et les dégradations importantes ne sont plus acceptées.

Procédure recuit simulé (solution initiale s)

Poser $T \leftarrow T_0$

Répéter :

Choisir aléatoirement 's' appartenant à N(s)

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

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

Mettre à jour T

Jusqu'à ce que le critère de terminaison soit satisfait

Fin

Et voici l'interprétation de son fonctionnement :

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

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

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

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

- 1) Choisir, aléatoirement, une solution initiale x du système à optimiser et évaluer la valeur de la fonction objectif $f = f(x)$;
- 2) Choisir une température initiale "élevée" T .
- 3) Perturber cette solution pour obtenir une nouvelle solution $x' = x + \Delta x$;
- 4) Calculer $\Delta f = f(x') - f(x)$;
- 5) Accepter ou refuser la solution x' , en appliquant une certaine "règle d'acceptation"
- 6) Sauver le meilleur point rencontré ;
- 7) Si l'"équilibre thermodynamique" du système à la température T est atteint,
Alors abaisser légèrement la température T ;
Sinon Aller à l'étape 3) ;
- 8) Si le "système est figé" (par exemple, la température T est inférieure à une température seuil voisine de 0),
Alors Aller à l'étape 9) ;
Sinon Aller à l'étape 3) ;
- 9) Solution = meilleur point trouvé ; Arrêt du programme.

Figure I.2 Algorithme du recuit simulé

I.6.3 Avantages

- Très simple et très rapide à mettre en place.
- Convergence vers un optimum global, la prédiction de la future à partir du présent ne nécessite pas la connaissance du passé. Plus clairement cette métaheuristique ne nécessite pas de mémoire (passé) afin de trouver les espaces de recherche locaux suivants (futur). Ce la signifie également que contrairement à d'autres métaheuristiques.
- Le recuit simulé peut trouver la meilleure solution si on le laisse chercher indéfiniment.

I.6.4 Inconvénients

- Non utilisation de mémoire bride.
- Il faut déterminer les paramètres à la main : température initiale, modification élémentaire en testant divers valeurs.

I.6.5 Applications

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

I.6.6 Pseudocode

Voici le pseudo code de la version la plus simple du recuit simulé:

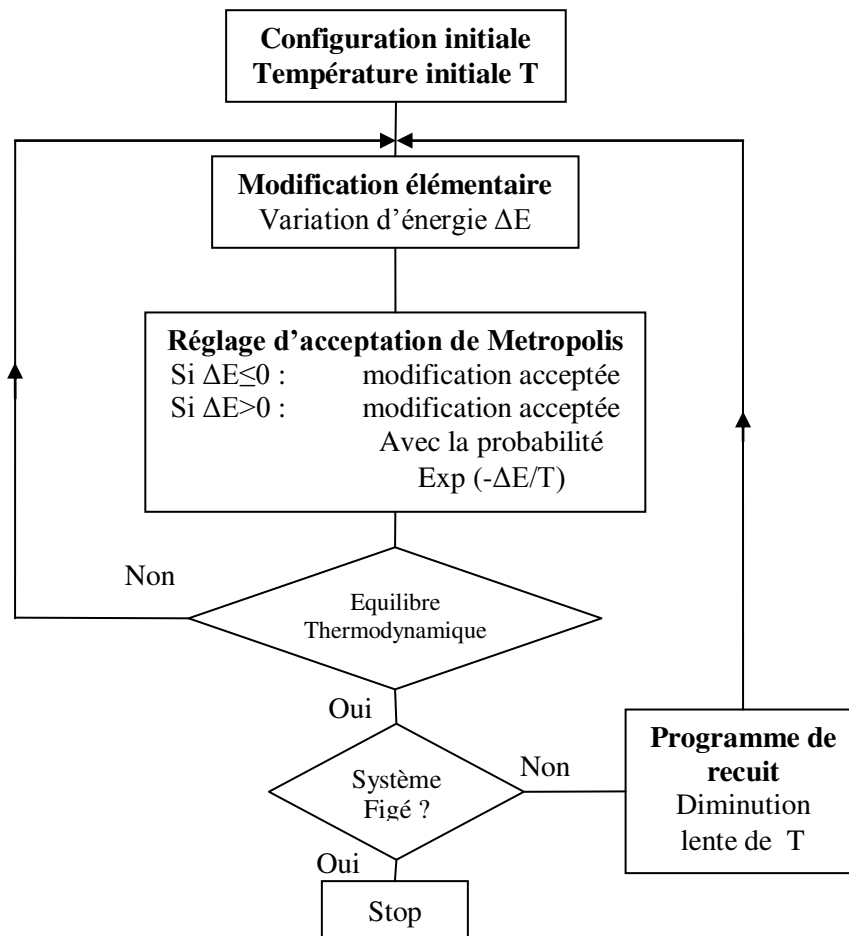


Figure I.3 : Organigramme de l'algorithme du recuit

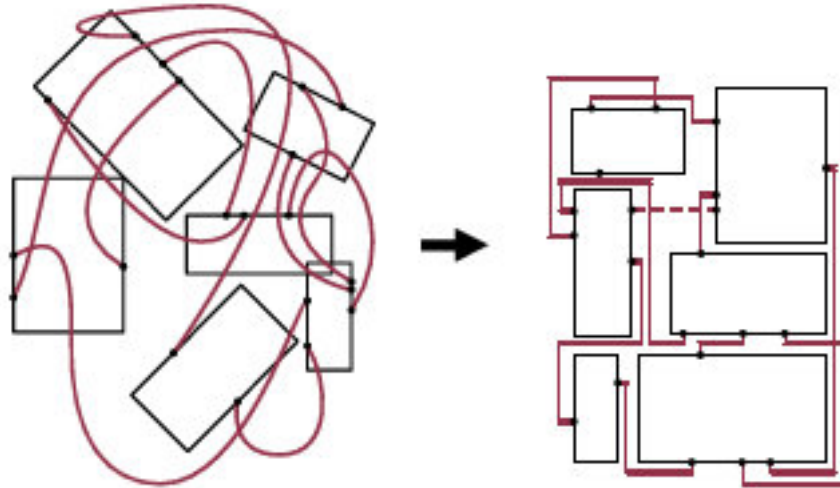


Figure I.4 a) désordre

Figure I.4 b) ordre

Figure I.4 : Recuit simulé appliqué au placement de composants électroniques

I.7 Algorithmes génétiques

Les algorithmes génétiques (AG) sont des techniques de recherche inspirées par l'évolution biologique des espèces. Introduits par J.H. Holland au début des années 1970, ils ont d'abord suscité un intérêt limité, du fait de leur important coût d'exécution. Ils connaissent, depuis les années 1990, un développement considérable, notamment suite à l'apparition des architectures massivement parallèles, qui exploitent leur (parallélisme intrinsèque) [1].

I.7.1 Principe de base

Le principe d'un AG se décrit simplement, dans le cas de la minimisation d'une fonction f . Un ensemble de N points, qui peuvent être choisis au hasard, constitue la population initiale ; chaque individu x de la population possède une certaine compétence, qui mesure son degré d'adaptation à l'objectif visé, x est d'autant plus compétent que $f(x)$ est petit. Un AG consiste à faire évoluer progressivement, par générations successives, la composition de cette population, en maintenant sa taille constante, d'une génération à la suivante, la (compétence) de la population doit globalement s'améliorer ; un tel résultat est obtenu en mimant les deux principaux mécanismes qui régissent l'évolution des êtres vivants : la sélection naturelle (qui détermine quels membres d'une population survivent et se reproduisent) et la reproduction (qui assure le brassage et la recombinaison des

gènes parentaux, pour former des descendants aux potentialités nouvelles).

En pratique, chaque individu est codé par une chaîne de caractères de longueur donnée (de même qu'un chromosome est formé d'une chaîne de gènes).

Le passage d'une génération la suivante se déroule en deux phases : une phase de reproduction et une phase de remplacement. La phase de reproduction consiste à appliquer des opérateurs, dits génétiques, sur les individus de la population courante, pour engendrer de nouveaux individus ; les opérateurs les plus utilisés sont le croisement, qui produit deux descendants à partir de deux parents, et la mutation, qui produit un nouvel individu à partir d'un seul individu. Les individus de la population prenant part à la reproduction sont préalablement sélectionnés, en respectant le principe suivant : plus un individu est compétent, plus sa probabilité de sélection est élevée. La phase de remplacement consiste ensuite à choisir les membres de la nouvelle génération : on peut, par exemple, remplacer les plus mauvais individus (au sens de la fonction objectif) de la population courante par les meilleurs individus produits (en nombre égal). L'algorithme est interrompu après un nombre donné de générations.

Nous avons représenté en figure suivante le principe d'un algorithme génétique. [2]

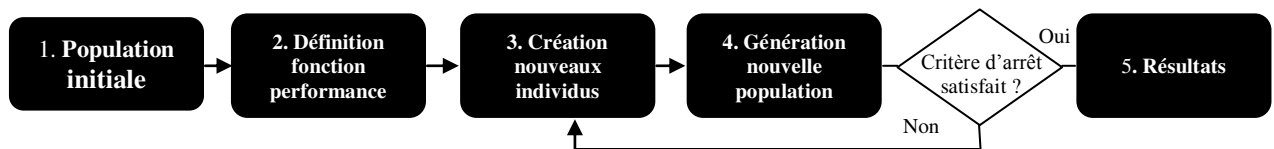


Figure I.5 : étapes de l'algorithme génétique

I.7.2 Avantages

- Aucune hypothèse à faire sur l'espace de recherche.
- Nombreuses méthodes disponibles.
- Solutions intermédiaires interprétables.
- Adaptation rapide à de nouveaux environnements.
- Coévolution (tournoi), parallélisme et distribution aisés.
- Les représentations facilitent la compréhension.

I.7.3 Inconvénients

- Aucune garantis de solution optimale en un temps fini.
- Initialisation de plusieurs paramètres, choix des méthodes importantes.
- Coût d'exécution important [1].

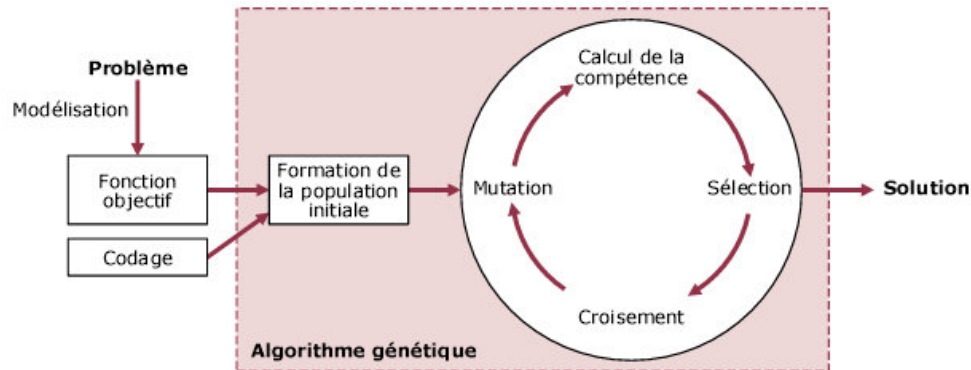


Figure I.6 : Résolution d'un problème par algorithme génétique

Voici l'algorithme génétique de base [19].

<u>Début</u>
1 : Générer une population aléatoire de n chromosomes
2 : Evaluer la fitness des chromosomes avec la fonction $f(x)$
3 : Répéter
4 : Calcule la fonction fitness $f(x)$, pour tout chromosome x
5 : Appliquer l'opération de sélection
6 : Appliquer l'opération de croisement avec une probabilité PC
7 : Appliquer l'opération de mutation avec une probabilité PM
8 : Ajouter les nouveaux chromosomes à la nouvelle population
9 : Calcule la fonction fitness $f(x)$, pour tout chromosome x

10 : Appliquer l'opération de remplacement
11 : <u>Jusqu'a</u> la satisfaction des conditions de terminaison
<u>Fin</u>

Figure I.7 Algorithme génétique de base

L'algorithme génétique comporte trois phases distinctes [22].

- La production de la population d'individus la mieux adaptée pour contribuer a la reproduction de la génération suivante (version artificielle de la sélection naturelle) elle peut être mise en œuvre sous plusieurs formes algorithmique;
- la phase de reproduction, qui exploite essentiellement les operateurs de croisement et de mutation ;
- la stratégie de remplacement des populations parent et enfant par la génération suivante. Elle pourra être mise en œuvre sous plusieurs formes.

I.8 La méthode recherche Tabou

La recherche Tabou a été appliquée pour résoudre un problème d'affectation de fréquences (Dahl et al, 1995). L'algorithme est exécuté avec un seul jeu de poids qui est fixé suivant les priorités des objectifs

Tabou, un sujet qu'il est préférable de ne pas aborder si l'on veut respecter les codes de la société.

La recherche Tabou dans le domaine de la recherche opérationnelle

La recherche Tabou est une méthode d'optimisation mathématique de la famille des techniques de recherche locale présentée pour la première fois en 1986, et elle est devenue très classique en optimisation combinatoire [22].

Elle se distingue des méthodes de recherche locale simples par l'introduction de la notion d'historique dans la politique d'exploration des solutions pour diriger au mieux la recherche dans l'espace. Cette méthode s'est révélée particulièrement efficace et a été

appliquée avec succès à de nombreux problèmes difficiles.

I.8.1 Principe de la méthode

L'idée de départ

Se déplacer de solution en solution (en visitant éventuellement des solutions moins bonnes) en s'interdisant de revenir à une solution déjà rencontrée.

- A chaque itération, on examine $V(i)$ et nous allons sur la meilleure solution i' même si le coup remonte ($F(i') > F(i)$).

Donc : La recherche Tabou ne s'arrête pas au premier optimum trouvé.

- Le danger serait alors de revenir à i immédiatement, puisque i est meilleure que i' . Pour éviter de tourner ainsi en rond, on crée une liste T qui mémorise les dernières solutions visitées et qui interdit tout déplacement vers une solution de cette liste. Cette liste T est appelée liste Tabou.
- On conserve en cours de route la meilleure solution trouvée .
- On stoppe dès que le critère de fin est vérifié.

I.8.2 Principe de l'algorithme

```

Générer une solution initiale S de manière aléatoire
 $S^* \leftarrow S ; C^* \leftarrow F(S) / S^*$  est la meilleure solution rencontrée,  $C^*$  est son coût
et F la
    fonction objectif
Ajouter S à la liste Tabou ;  $K \leftarrow 0$ 
Répéter tant qu'un critère de fin n'est pas vérifié
    Choisir parmi le voisinage de  $S_K$ ,  $V(S_K)$ , le mouvement qui minimise F et
qui n'appartient pas à la liste Tabou, meilleur ( $S_K$ )
     $S_{K+1} \leftarrow \text{meilleur}(S_K)$ 
Si la liste Tabou est pleine alors
        Remplacer le dernier élément de la liste Tabou par  $S_{K+1}$ 
Si non
        Ajouter  $S_{K+1}$  à la liste Tabou
Fin si
Si ( $C(S_{K+1}) < C^*$ ) alors
         $S^* \leftarrow S_{K+1}, C^* \leftarrow C(S_{K+1})$ 
Fin si
Fin d'algorithme

```

Figure I.8 Algorithme de la méthode recherche Tabou

Lorsque la mémoire est pleine, elle est gérée comme une liste circulaire en FIFO (First In First Out)

On élimine le plus vieux point Tabou et on insère la nouvelle solution. La taille de la mémoire permet de ne pas saturer rapidement les ressources disponibles pour la recherche et permet de surcroît d'adapter facilement la méthode à un espace de recherche dynamique.

I.8.3 Le critère d'arrêt

Le critère d'arrêt sert à déterminer le moment où l'on considère que la solution trouvée est d'assez bonne qualité pour être recevable. On peut par exemple :

- fixer un nombre maximum d'itérations
- après un nombre fixe d'étapes n'ayant pas amélioré la solution s.
- fixer un temps limite après lequel la recherche doit s'arrêter.

A partir de cet algorithme initial, certaines adaptations ont été élaborées. Ces améliorations ont été introduites afin de pallier à des problèmes constatés dans l'analyse de l'exploration de l'espace de recherche.

I.8.4 Avantages

- Grande efficacité.
- Fonctionnement simple à comprendre.

I.8.5 Inconvénients

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

Voici l'organigramme simple de l'algorithme tabou.

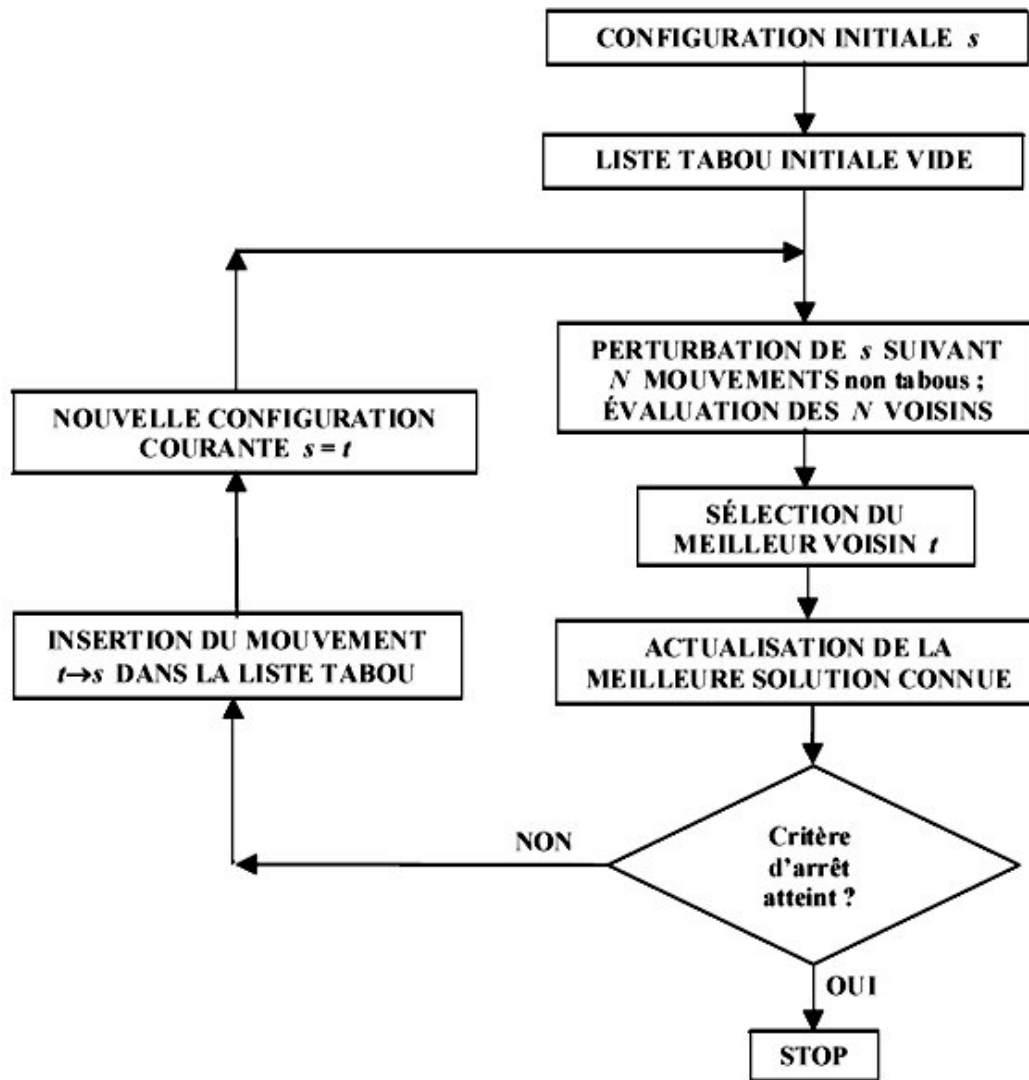


Figure I.9: Organigramme de l'algorithme tabou simple

I.9 Conclusion

Dans ce chapitre, nous avons décrit les méthodes d'optimisation basées sur les métaheuristiques en les divisant en deux classes : les méthodes à solution unique et les méthodes à population de solutions.

Nous avons pu constater au fur et à mesure d'un court état de l'art pour chaque méthode, que leur utilisation en extraction de connaissances est relativement répandue pour les différentes tâches que nous désirons étudier.

Dans l'étude citées dans ce chapitre, sur la fonction d'évaluation d'une solution qui représente, en effet, souvent, la partie la plus sensible de ces méthodes mais aussi sur le codage qui va directement influencer, comme nous l'avons vu dans ce chapitre, sur le type de résultats attendus. Cette première réflexion amène automatiquement à une étape d'étude et de conception d'opérateurs spécifique.

CHAPITRE II

LES RESEAUX DE NEURONES ARTIFICIELS

II.1. Introduction

Les réseaux de neurones artificiels ou ‘**RNA**’ formels ont été développés dans les années 80 pour résoudre des problèmes d’ingénierie: estimation, approximation des fonctions, prédiction, élimination d’échos, égalisation des canaux, etc....

Les réseaux de neurones sont des modèles mathématiques et informatiques, des assemblages d’unités de calcul appelés neurones formels, et dont l’inspiration originelle était le fonctionnement cérébral de l’être humain

En 1986, Rumelhart, Hinton et Williams publient, l’algorithme de la rétro-propagation de l’erreur qui permet d’optimiser les paramètres d’un réseau de neurones à plusieurs couches.

Actuellement, les chercheurs s’intéressent beaucoup plus à l’exploitation des structures d’identification et de contrôle mono et multi variable de la commande adaptative avec les réseaux de neurones [4].

II.2 Utilité des RNA

Les réseaux de neurones, en tant que système capable d’apprendre, mettent en œuvre le principe de l’induction, c’est à dire l’apprentissage par l’expérience. Par confrontation avec des situations ponctuelles, ils infèrent un système de décision intégré dont le caractère générique est fonction du nombre de cas d’apprentissages rencontrés et de leur complexité par rapport à la complexité du problème à résoudre.

Grâce à leur capacité de classification et de généralisation, les RN sont aussi utilisés dans des problèmes de nature statistique, tels que la classification automatique de codes postaux ou la prise de décision concernant un achat boursier en fonction de l’évolution des cours.

Il est à noter que le réseau de neurones ne fournit pas toujours de règle exploitable par un humain. Le réseau reste souvent une boîte noire qui fournit une réponse quand on lui présente une donnée, mais le réseau ne fournit pas de justification facile à interpréter.

II.3 Modèle d'un neurone biologique

Cette section décrit un modèle très grossier des neurones biologiques qui servi à la mise en place des premiers neurones formels. La figure II.1 représente un neurone biologique [4] [12].

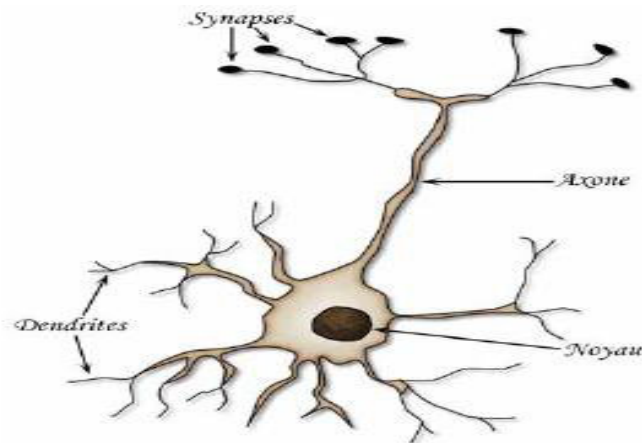


Figure II.1 Neurone biologique

Le neurone est une cellule nerveuse elle se compose d'un corps cellulaire appelé « soma » qui contient le noyau (ou se déroule les activités cellulaires vitales) de prolongement appelé « neurite », ces dernières sont de deux types, les dendrites qui servent de canaux d'entrées et l'axone, unique qui est le canal de sortie.

Du point de vu fonctionnel, il faut considérer le neurone comme une entité polarisée, c'est-à-dire que l'information ne se transmet que dans un seul sens : des dendrites vers l'axone. Le neurone va donc recevoir des informations, venant d'autres neurones, grâce à ses dendrites. Il va ensuite y avoir sommation, au niveau du corps cellulaire, de toutes ces informations et via un potentiel d'action (un signal électrique) le résultat de l'analyse va transiter le long de l'axone jusqu'aux terminaisons synaptiques. A cet endroit, lors de l'arrivée du signal, des vésicules synaptiques vont venir fusionner avec la membrane cellulaire, ce qui va permettre la libération des neurotransmetteurs (médiateurs chimiques) dans la fente synaptique. Le signal électrique ne pouvant pas passer la synapse (dans le cas d'une synapse chimique), les neurotransmetteurs permettent donc le passage des informations, d'un neurone à un autre.

Les synapses possèdent une sorte de «mémoire» qui leur permet d'ajuster leur fonctionnement. En fonction de leur «histoire», c'est-à-dire de leur activation répétée ou non entre deux neurones, les connexions synaptiques vont donc se modifier.

Ainsi, la synapse va faciliter ou non le passage des influx nerveux. Cette plasticité est à l'origine des mécanismes d'apprentissage.

II.4 Modèle d'un neurone artificiel

Le neurone formel (figure II.2) est une modélisation mathématique qui reprend les principes du fonctionnement du neurone biologique, en particulier la sommation des entrées. Sachant qu'au niveau biologique, les synapses n'ont pas toutes la même «valeur» (les connexions entre les neurones étant plus ou moins fortes), les auteurs ont donc créé un algorithme qui pondère la somme de ses entrées par des poids synaptiques (coefficients de pondération). En général, un neurone formel est un élément de traitement possédant n entrées $x_1, x_2, \dots, x_i, \dots, x_n$ (Qui sont les entrées externes ou les sorties des autres neurones) et une ou plusieurs sorties. Son traitement consiste à effectuer à sa sortie y_j le résultat d'une fonction de seuillage f (dite aussi la fonction d'activation) de la somme pondérée.

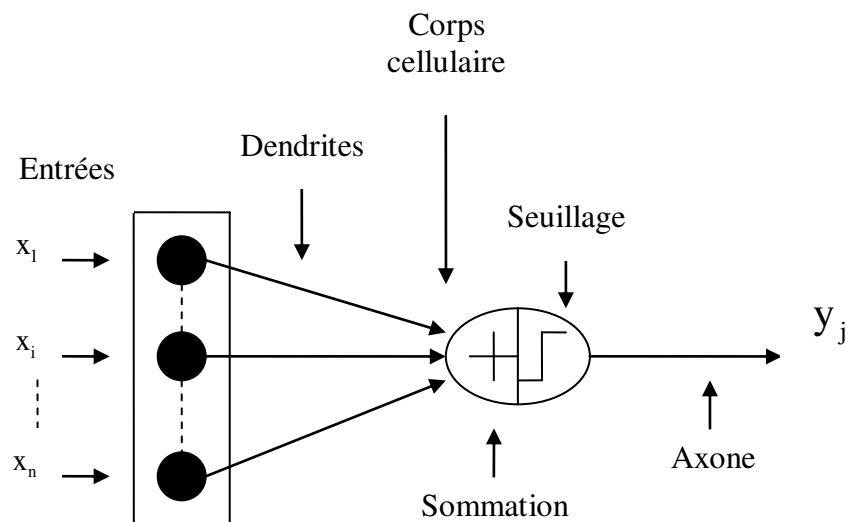


Figure II.2 Le neurone artificiel

II.4.1 Modélisation d'un neurone artificiel

La modélisation consiste à rassembler les connaissances que l'on a du comportement dynamique du processus, par une analyse physique des phénomènes mis en jeu, et une analyse des données expérimentales. Ces analyses conduisent à la définition des grandeurs caractérisant le processus, c'est-à-dire ses entrées, ses variables d'état et ses sorties.

La modélisation consiste à mettre en œuvre un système de réseau de neurones sous un aspect non pas biologique on aura une correspondance pour chaque élément composant le neurone biologique, donc une modélisation pour chacun d'entre eux (figure II.3).

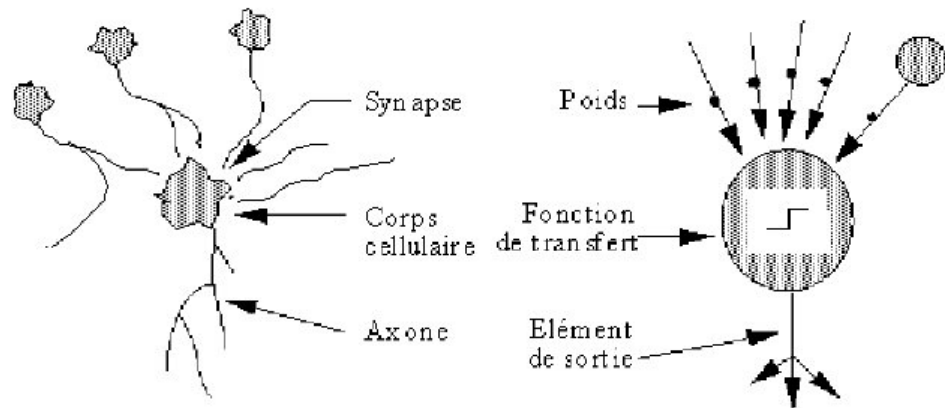


Figure II.3 : Analogie entre le neurone biologique et le neurone artificiel

On pourra résumer cette modélisation par le tableau II.1, qui montre la transition entre le neurone biologique et le neurone artificiel. [4][13]

Neurone biologique	Neurone artificiel
Dendrite	Signal d'entrée
Synapses	Poids de connexions
Somma	Fonction d'activation
Axones	Signal de sortie

Tableau II.1 : Analogie entre le neurone biologique et le neurone artificiel

II.4.2 Fonctionnement d'un RNA

Le neurone calcule la somme de ses entrées puis cette valeur passe à travers la fonction d'activation pour produire sa sortie.

En règle générale, le calcul de la valeur de cette fonction peut se décomposer en deux étapes :

Une combinaison linéaire des entrées :

$$v = w_0 + \sum_{i=1}^n w_i x_i \quad (\text{II.1})$$

Les w_i sont appelés poids synaptiques, w_0 est appelé biais. Le biais peut être considéré comme la pondération de l'entrée 0 fixée à 1. v est appelé potentiel du neurone. f : est la fonction d'activation du neurone.

La sortie du neurone est :

$$y = f(v) = f\left(\sum_{i=0}^n w_i x_i\right) \quad (\text{II.2})$$

II.4.2.1 Fonction d'activation

La fonction d'activation (ou fonction de seuillage, ou encore fonction de transfert) sert à introduire une non linéarité dans le fonctionnement du neurone. Les fonctions de seuillage présentent généralement trois intervalles :

- en dessous du seuil, le neurone est non actif (souvent dans ce cas, sa sortie vaut 0 ou -1) ;
- aux alentours du seuil, une phase de transition ;
- au-dessus du seuil, le neurone est actif (souvent dans ce cas, sa sortie vaut 1).

Dans sa première version, le neurone formel était implémenté avec une fonction à seuil, mais de nombreuses versions existent. Ainsi le neurone de McCulloch et Pitts a été généralisé de différentes manières, en choisissant d'autres fonctions d'activations, comme les fonctions de neurones énumérées au tableau II.4. Les trois fonctions les plus utilisées sont les fonctions « seuil », « linéaires » et « sigmoïdes ». [10]

Le choix de la fonction d'activation dépend de l'application. S'il faut avoir des sorties binaires c'est la première fonction que l'on choisit habituellement. Une entrée spéciale est pratiquement toujours introduite pour chaque neurone. Cette entrée, normalement appelée biais, sert pour déplacer le pas de la fonction d'activation sur l'axe S.

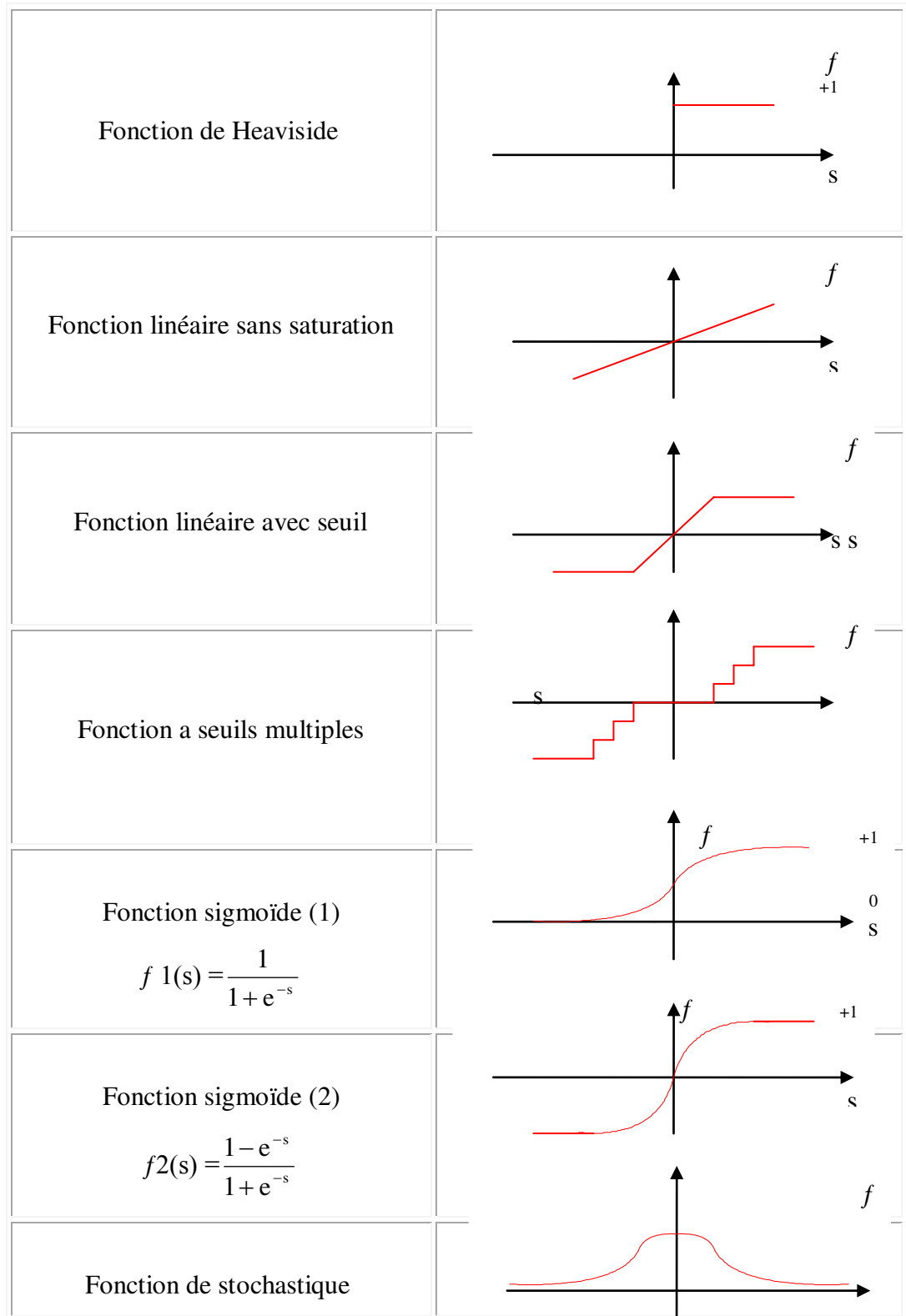


Figure II.4 : Les modèles de fonctions d'activation.

II.4.2.2. Fonction de combinaison

Considérons un neurone quelconque. Il reçoit des neurones en amont un certain nombre de valeurs via ses connexions synaptiques, et il produit une certaine valeur en utilisant une fonction de combinaison. Cette fonction peut donc être formalisée comme étant une fonction vecteur à scalaire, notamment :

- Les réseaux de type MLP (Multi Layer Perceptron) calculent une combinaison linéaire des entrées, c'est-à-dire que la fonction de combinaison renvoie le produit scalaire entre le vecteur des entrées et le vecteur des poids synaptiques.

- Les réseaux de type RBF (Radial Basis Function) calculent la distance entre les entrées, c'est-à-dire que la fonction de combinaison renvoie la norme euclidienne du vecteur issu de la différence vectorielle entre les vecteurs d'entrées. [14]

II.5 Les propriétés d'un réseau de neurones

II.5.1 La propriété d'approximation universelle

La propriété d'approximation universelle a été démontrée par [Cybenko, 1989] et [Funahashi, 1989] et peut s'énoncer de la façon suivante :

Toute fonction bornée suffisamment régulière peut être approchée uniformément, avec une précision arbitraire, dans un domaine fini de l'espace de ses variables, par un réseau de neurones comportant une couche de neurones cachées en nombre fini, possédant tous la même fonction d'activation, et un neurone de sortie linéaire. Comme le montre ce théorème, le nombre de neurones cachés doit être choisi convenablement pour obtenir la précision voulue.

II.5.2. La propriété de parcimonie

Lorsqu'on cherche à modéliser un processus à partir des données, on s'efforce toujours d'obtenir les résultats les plus satisfaisants possibles avec un nombre minimum de paramètres ajustables. Dans cette optique, [Hornik et al.1994] ont montré que [14] :

Si le résultat de l'approximation (c'est-à-dire la sortie du réseau de neurones) est une fonction non linéaire des paramètres ajustables, elle est plus parcimonieuse que si elle est une fonction linéaire de ces paramètres. De plus, pour des réseaux de neurones à fonction d'activation sigmoïdale, l'erreur commise dans l'approximation varie comme l'inverse du

nombre de neurones cachés, et elle est indépendante du nombre de variables de la fonction à approcher. Par conséquent, pour une précision donnée, donc pour un nombre de neurones cachés donné, le nombre de paramètres du réseau est proportionnel au nombre de variables de la fonction à approcher.

Ce résultat s'applique aux réseaux de neurones à fonction d'activation sigmoïdale puisque la sortie de ces neurones n'est pas linéaire par rapports aux poids synaptiques. Cette propriété montre l'intérêt des réseaux de neurones par rapport à d'autres approximateurs comme les polynômes dont la sortie est une fonction linéaire des paramètres ajustables : pour un même nombre d'entrées, le nombre de paramètres ajustables à déterminer est plus faible pour un réseau de neurones que pour un polynôme.

II.6 Topologie des réseaux de neurones

On peut classer les réseaux de neurones en deux grandes catégories, selon la dépendance de l'évolution de ceux-ci en fonction explicite du temps.

II.6.1. Les réseaux statiques ou réseau à couche (FEED FORWARD)

C'est le cas de réseaux statiques, ou le temps n'est pas un paramètre significatif. En d'autres termes, la modification d'entrée n'entraîne que la modification stable de la sortie, mais elle n'entraîne pas le retour de l'information de cette entrée.

Les réseaux statiques (FEED FORWARD) sont des réseaux à couches, constitués d'une couche d'entrée, une couche de sortie et entre les deux au moins une couche composée de nombreux éléments de traitements non linéaires, appelée couche cachée.

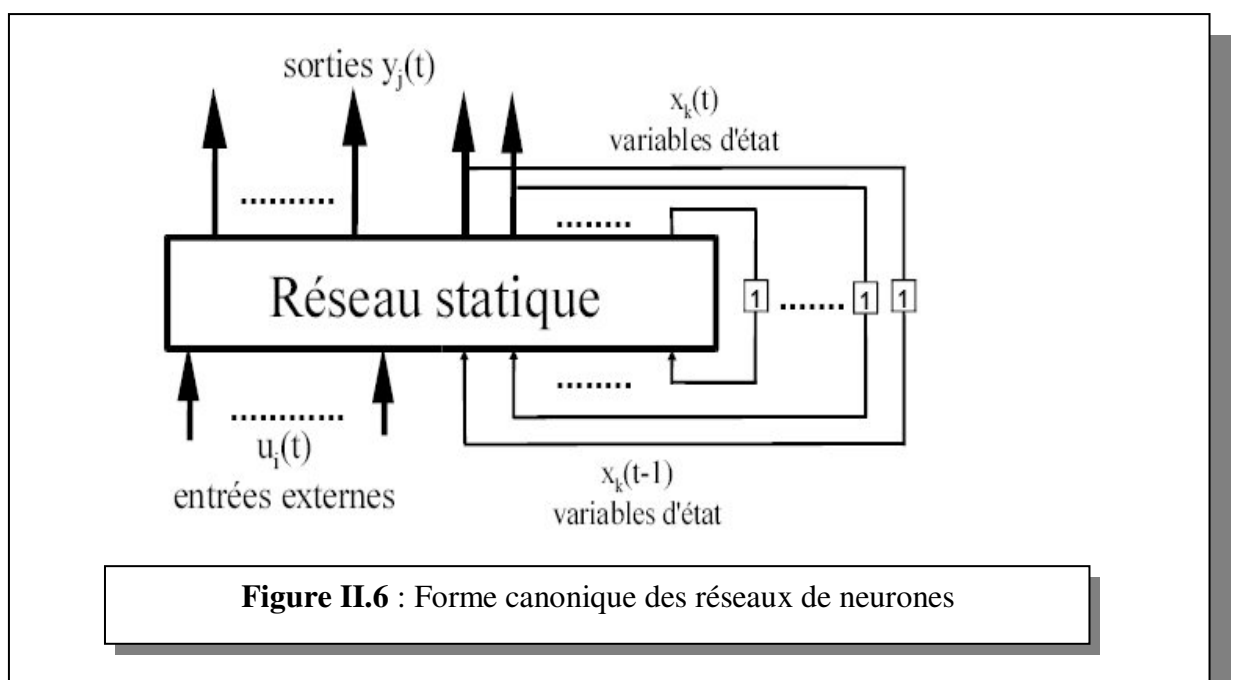
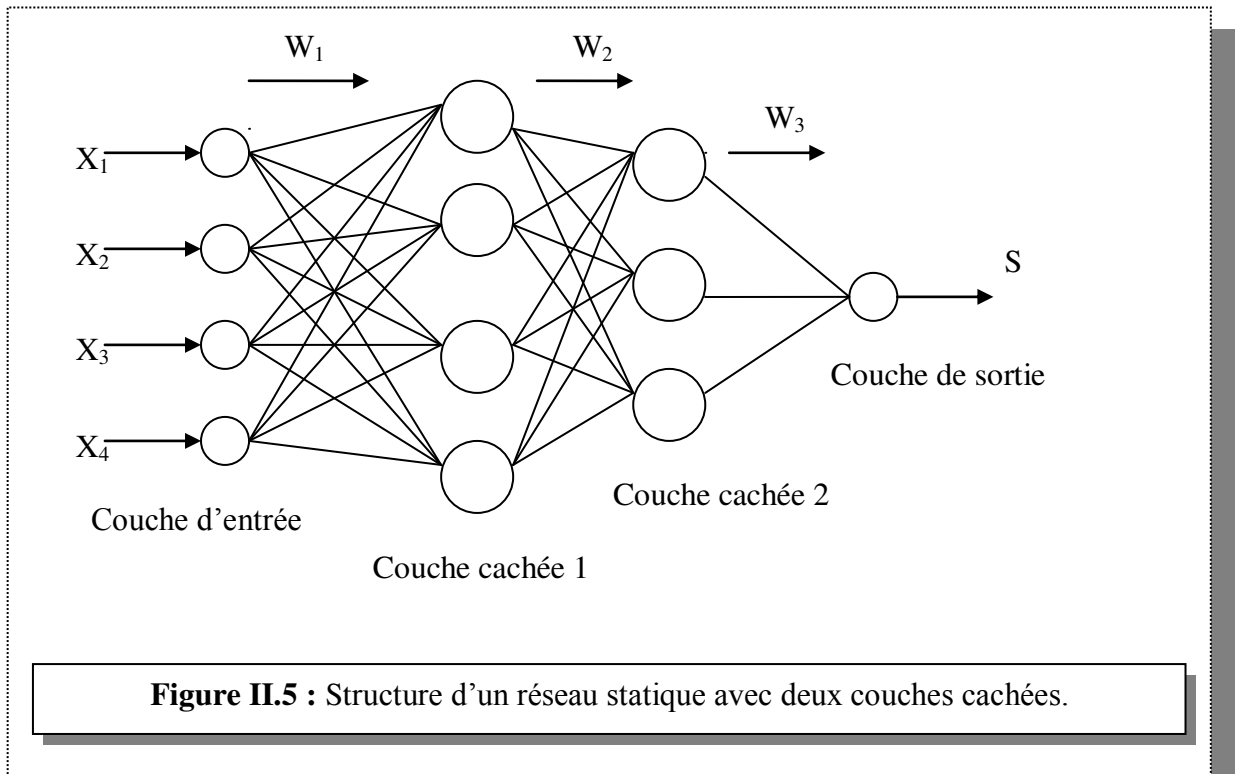
Les signaux des entrées se propagent de la première couche à la couche de sortie en passant par les couches cachées, Il n'y a pas des communications entre les unités de la même couche, d'où le nom de feedforwarde. Les liens dirigés connectant les neurones sont appelés les interconnexions [4].

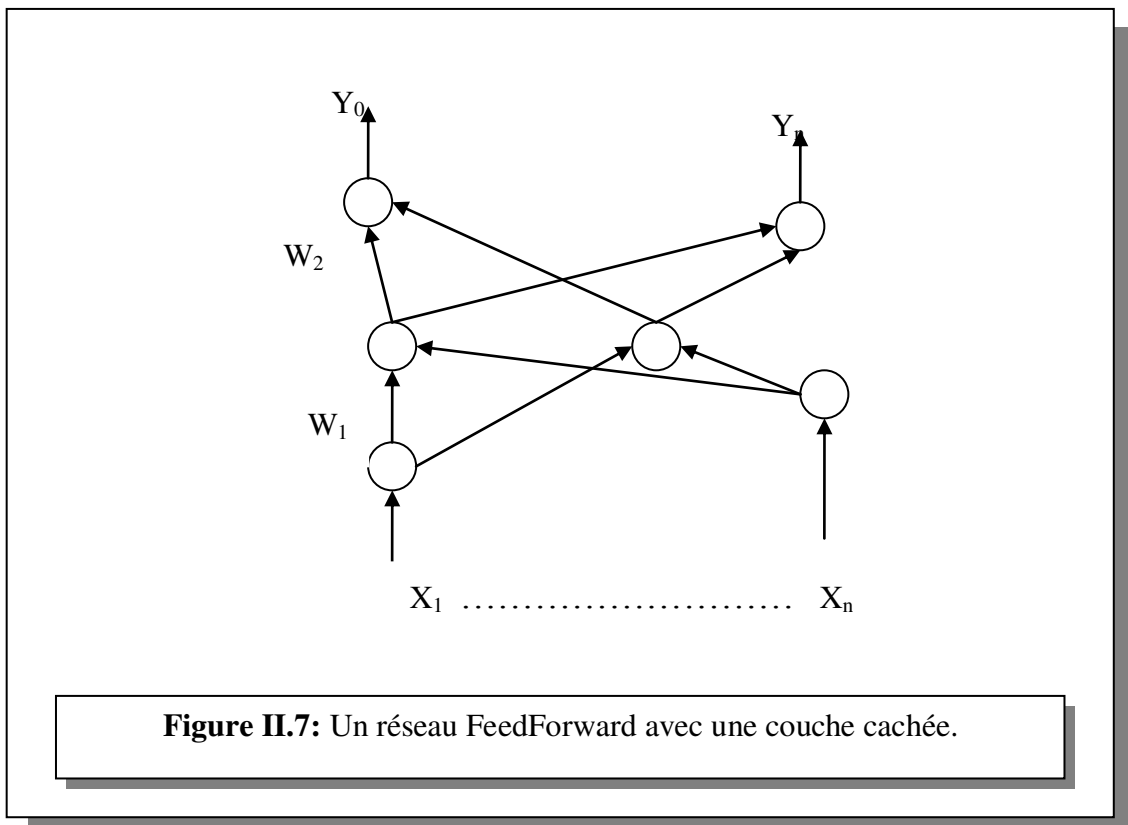
On distingue des réseaux à deux couches tel que le perceptron et *l'adaline* (adaptive linear neurone) qui sont caractérisés par :

- la simplicité de réglage d'apprentissage.
- la facilité de détermination de l'influence d'un neurone d'entrée sur l'erreur d'un neurone de sortie d'en déduire les modifications à apporter au lien qui les relie.

- La limitation au calcul de fonction très simple.

Ces réseaux ne pouvaient résoudre que des problèmes simples de classification. Pour des problèmes complexes, une solution consiste à organiser le réseau en plusieurs couches [4].



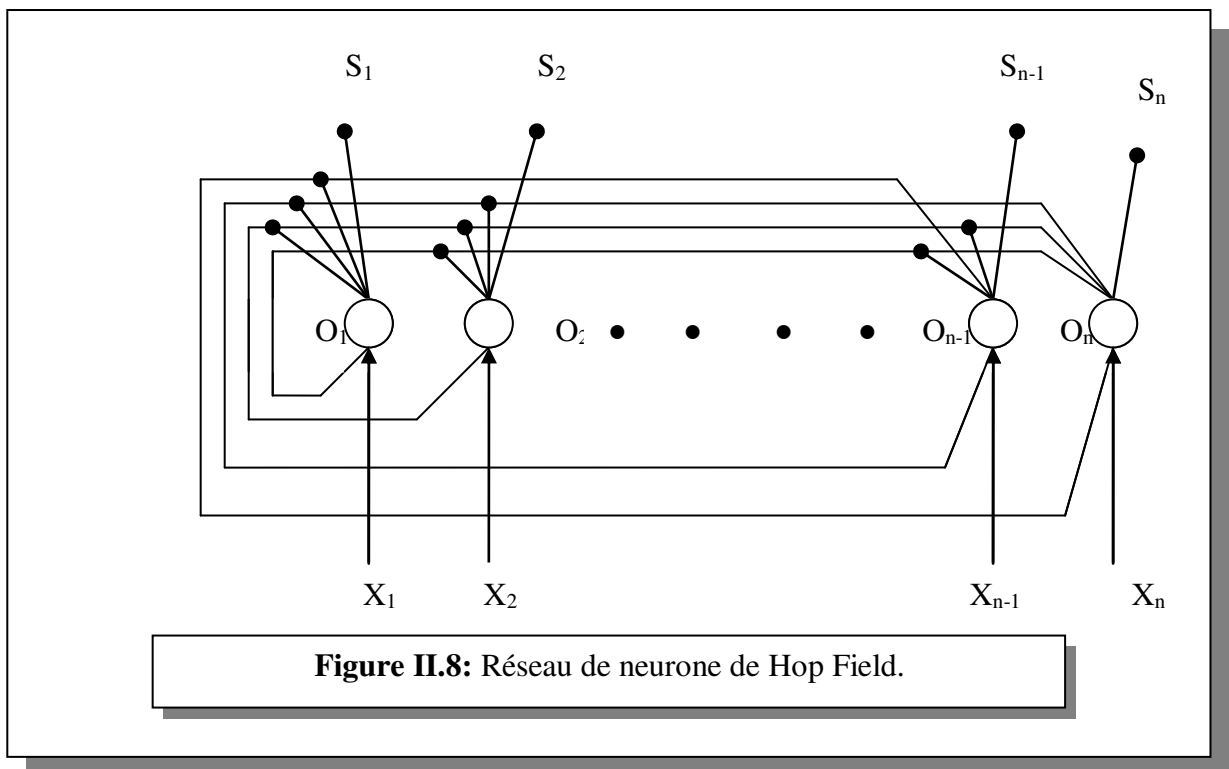


II.6.2. Les réseaux dynamiques (récurrents)

Comme leurs noms l'indique, contiennent des rebouclages partiels ou totaux entre neurones, ils représentent donc une évolution dépendante du temps.

Il faut bien distinguer la dépendance théorique, pour laquelle l'état du réseau à un certain instant dépend de son état à l'instant ou aux instants précédents, du temps nécessaire à obtenir une réponse, dans le cas d'une réalisation matérielle ou d'une simulation sur ordinateur.

La théorie de ces réseaux fut publiée par HOPFIELD dans les années 1980 et sont souvent appelées « **HOPFIELD nets** ». Ils sont principalement utilisés pour l'implémentation des mémoires associatives « BAM : bidirectionnel Associative Memory » [12].



Le perceptron multicouche ordinaire ou la carte auto organisatrice sont des réseaux statiques. Par contre, le réseau du HOPFIELD ou le perceptron avec rebouclage est des réseaux dynamiques.

Les critères motivant les choix d'un type de réseau sont la simplicité de mise en œuvre et l'efficacité des algorithmes d'adaptation appelés à répondre aux performances désirées du système, quelle que soit sa complexité [10].

L'opérateur non linéaire réalisé par un réseau, bouclé ou non, dépend des valeurs des coefficients de pondération du réseau.

Pour qu'un réseau effectue une tâche donnée, il faut donc ajuster la valeur des coefficients. Une tâche est définie par un ensemble d'exemples, ou couples (valeurs des entrées et valeurs des sorties désirées correspondantes), tels les couples (forme classe) en classification, ou les couples (commande sortie mesurée du processus) en modélisation ces couples constituent l'ensemble d'apprentissage.

II.7. Le Perceptron

A la fin des années **50**, la neurobiologie ne disposait pas encore des techniques modernes de l'électrochimie pour étudier les structures du cerveau. L'idée de certains chercheurs fut alors

d'utilisé les modélisations des neurones et de l'évolution des synapses pour simuler des réseaux de neurones. Ceux-ci pourraient exhiber des comportements proches de ceux du cerveau et ainsi fournir des explications sur le fonctionnement de celui-ci.

Le premier modèle fut présenté par ROSEBLATT c'était le perceptron ce modèle a pour propriétés [15].

- D'être spécifié en termes suffisamment précis pour permettre le test des performances annoncées.
- D'être suffisamment complexe pour que l'on puisse espérer des comportements intéressants.
- D'être suffisamment simple pour que ses performances puissent être prédites et analysées.
- Enfin et surtout d'être en accord avec les faits biologiques

Un perceptron comprend trois éléments principaux :

II.7.1. Rétine

Elle est constituée de cellules, sur lesquelles s'inscrit le stimulus.

II.7.2. Couche de cellules d'association

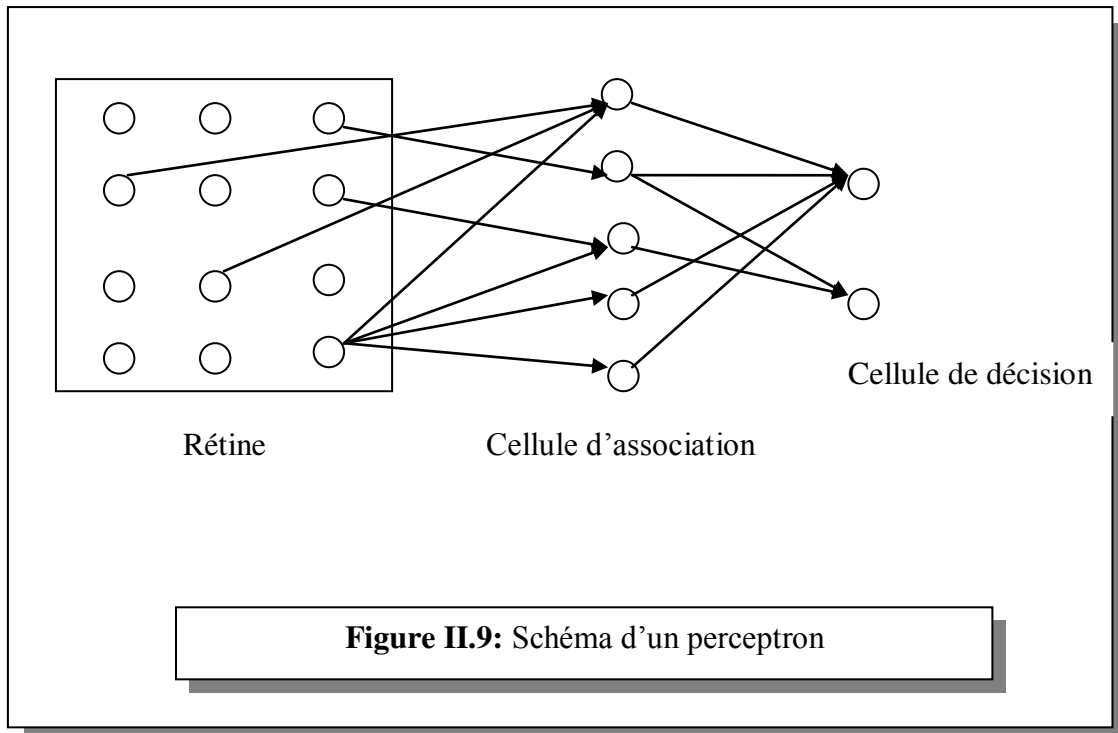
Chacune de ces cellules peut être connectée à des cellules de la rétine, et à d'autres cellules de décision. Elle fait la somme des impulsions qui lui viennent et des cellules aux quelles elle est connectée.

II.7.3. Couche de cellule de décision

Ces cellules fonctionnent comme les cellules d'association, elles reçoivent leurs entrées des cellules d'association ou d'autres cellules de décision. Elles représentent la sortie du perceptron. Le sens des connexions entre les cellules d'associations et les cellules de décisions est bi directionnel, ce qui un 'Feedback' de la sortie sur le réseau.

Remarque

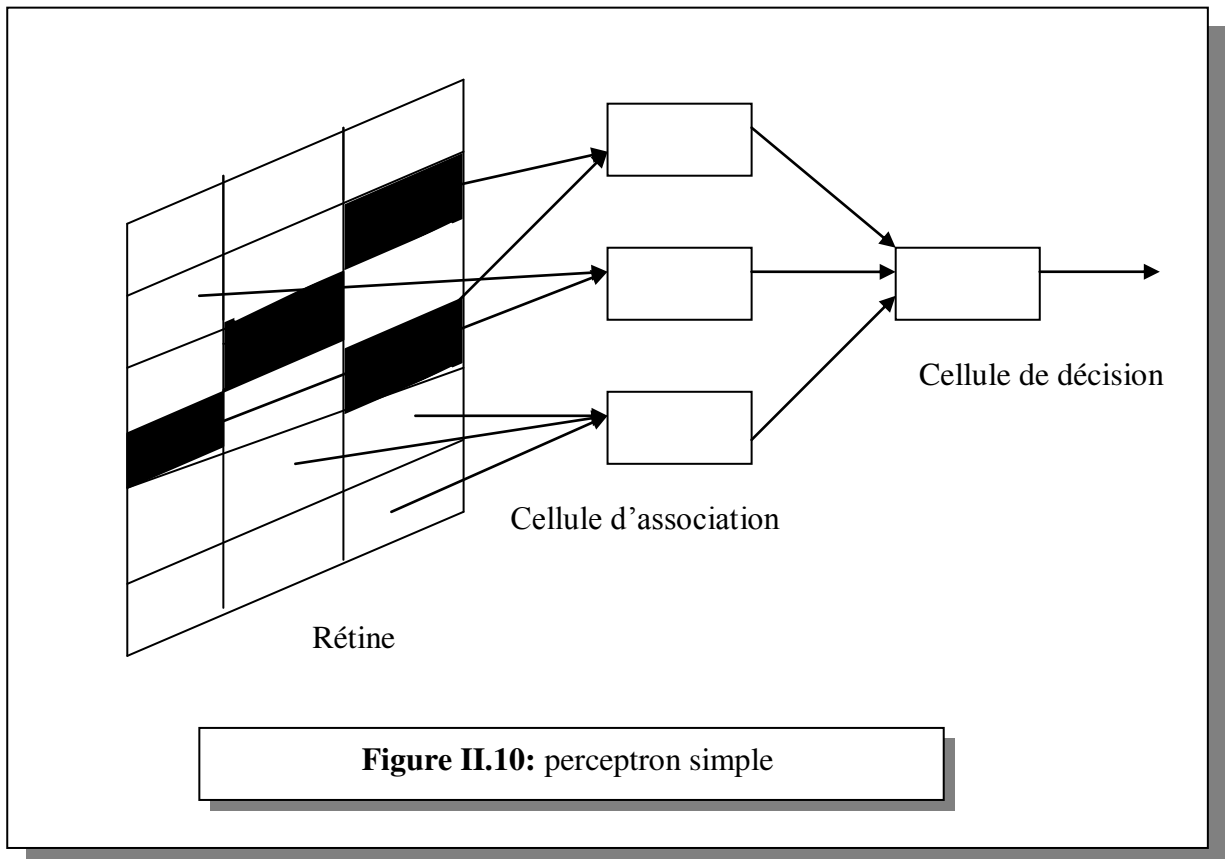
Les neurones constituant le perceptron sont des réseaux linéaires à seuil.



II.8 Perceptron simple :

Pour faciliter l'étude des comportements du perceptron, il est courant d'utiliser un perceptron d'une forme plus simple que celle définie par le modèle générale. Dans ce cas [19]:

- Les valeurs de sorties de toutes les cellules (rétine, association, décision) sont binaires en 0 et 1.
- Les fonctions réalisées par les cellules d'associations sont des fonctions booléennes et les cellules de décision sont des automates à seuil.
- Il n'existe pas de connexion entre deux cellules d'association, ni entre deux cellules de décisions.
- Les connexions entre cellules d'association et cellules de décision sont à sens unique association décision.



Un tel perceptron va être utilisé pour essayer de reconnaître des formes qui lui seront présentées sur sa rétine. Si l'on n'utilise qu'une cellule de décision, celle-ci devra répondre 1 lorsque la forme qui lui sera présentée correspond à une forme qu'on veut lui faire reconnaître et 0 dans tous les autres cas.

L'apprentissage du perceptron va se faire suivant le perceptron de HEED, c'est-à-dire en modifiant ses connexions [15].

II.9. Réseaux bouclés

Un réseau dynamique ou récurrent possède la même structure qu'un réseau multicouche muni de rétroactions. Les connexions rétroactives peuvent exister entre tous les neurones du réseau sans distinction, ou seulement entre certains neurones (les neurones de la couche de sortie et les neurones de la couche d'entrée ou les neurones de la même couche par exemple). La figure II.11 montre deux exemples de réseaux récurrents. Le premier est un simple multicouche qui utilise un vecteur d'entrée qui contient les copies des activations de la couche de sortie du réseau

et le deuxième est un réseau à mémoire se distingue du premier par la présence des unités mémoires

II.10. Identification et commande par réseaux de neurones

La commande de processus au moyen des réseaux de neurones, a connu un très grand essor cette dernière décennie. La vision de la biologie a pris, dans le domaine de la théorie des systèmes dynamiques, une tournure particulièrement intéressante. Elle a ouvert des perspectives de compréhension à la fois plus larges et plus méfiantes. Dans le but d'élargir son champ d'application, la théorie de contrôle ressent actuellement un besoin d'intégrer de nouveaux concepts regroupés sous le terme de commande intelligente.

L'objectif est d'introduire de nouveaux mécanismes permettant une commande plus simple, capable de s'adapter à des variations de l'environnement et démontrant des capacités d'apprentissage, telles les efforts et les interventions de l'home, tant dans les phases de conception que de conduite proprement dite, en soient significativement réduits.

La plupart des commandes utilisant un réseau de neurones en tant que contrôleur se distinguent par une étape d'identification et une étape de contrôle. L'identification consiste à élaborer un modèle neuronal qui est une estimation du processus à commander et cela au moyen d'une phase d'apprentissage. Celle-ci peut être soit préalable (hors ligne), ou bien elle peut se faire intégralement en ligne. La commande utilise les connaissances acquises pendant la phase d'identification et/ou de l'apprentissage en ligne pour élaborer des signaux de commande. Un apprentissage en ligne, pendant la commande du système, est intéressant si des perturbations viennent affecter le processus ou son environnement [21].

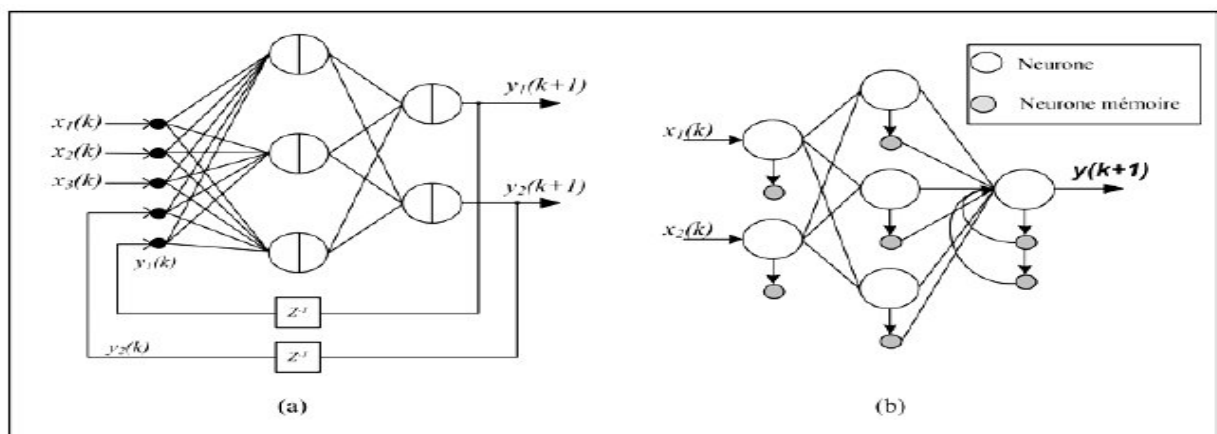


Figure II.11: Réseaux de neurones récurrent

II.11. Apprentissage dans les réseaux de neurones

L'apprentissage dans le contexte des réseaux de neurones, est le processus de modification des poids de connexions (y compris les biais) ou plus rarement du nombre de couches et de neurones (Man et Halang, 1997), afin d'adapter le traitement effectué par le réseau à une tâche particulière.

On distingue trois familles d'apprentissage :

– apprentissage supervisé: Dans ce cas, un superviseur(ou expert humain) fournit une valeur ou un vecteur de sortie (appelé cible ou sortie désirée), que le réseau de neurones doit associer à un vecteur d'entrée. L'apprentissage consiste dans ce cas à ajuster les paramètres du réseau afin de minimiser l'erreur entre la sortie désirée et la sortie réelle du réseau.

– Apprentissage semi-supervisé: ou apprentissage par renforcement: Ce mode d'apprentissage, suppose qu'un comportement de référence n'est pas possible, mais en revanche, il est possible d'obtenir des indications qualitatives (vrai, faux, ...) sur les performances du réseau.

– Apprentissage non supervisé: Dans cet apprentissage, les données ne contiennent pas d'information sur la sortie désirée. Il n'y a pas de superviseur. La tâche du réseau consiste, par exemple dans ce cas, à créer des regroupements de données selon des propriétés communes (classification).

II.11.1 Identification par réseaux de neurones

L'identification des systèmes linéaires est actuellement un domaine bien maîtrisé par l'automaticien. Cependant, de nombreux processus réels sont complexes et présentent des dynamiques non linéaires. De plus, sur ces mêmes systèmes, les connaissances des phénomènes physico-chimiques mises en jeu ne sont pas forcément mesurées. Le recours à des modèles de type "boite noire" devient nécessaire et l'identification du modèle est réalisée autour d'un ensemble de données entrées/sorties. Un réseau de neurone, qualifié de boîte noire, présente un outil incontournable pour l'identification de tels systèmes.

La figure II.12 montre le schéma général d'identification neuronale. Sur cette figure, le réseau de neurones est utilisé en parallèle avec un processus de type boîte noire à identifier.

La sortie y du processus est comparée à la sortie \hat{y} du réseau de neurones puis l'erreur

$e_i = y - \hat{y}$ est utilisée par un algorithme d'apprentissage approprié (exemple la rétropropagation) pour ajuster les paramètres du réseau neuronal [18].

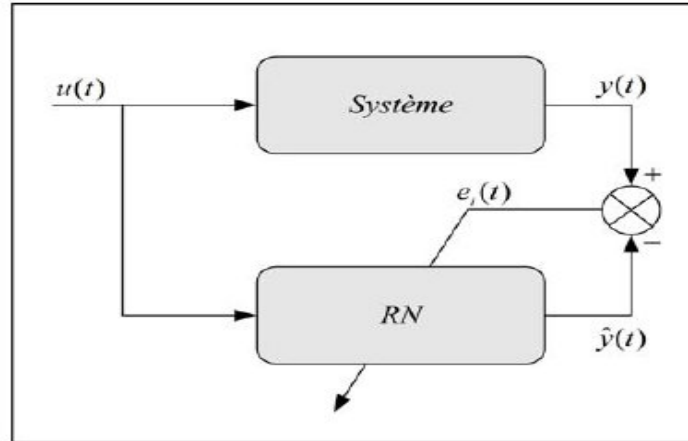


Figure II.12: Schéma de principe d'identification par réseau de neurones

II.11.2 Identification des processus par réseaux de neurones

Le principe de l'identification par réseau neuronal consiste à substituer aux modèles paramétriques classiques des modèles neuronaux, c'est-à-dire proposer un modèle établissant une relation entre son entrée et sa sortie et à déterminer, à partir du couple des signaux d'entrée-sortie, le comportement du modèle. Deux raisons importantes nous motivent [21] :

- Prédire le comportement d'un système pour différentes conditions de fonctionnement ;
- Elaborer une loi de commande à appliquer au processus, pour qu'il réalise l'objectif assigné.

II.12 Avantages et Inconvénients des réseaux de neurones

II.12.1 Avantage des réseaux de neurones

- Capacité de représenter n'importe quelle fonction, linéaire ou pas, simple ou complexe ;
- Faculté d'apprentissage à partir d'exemples représentatifs, par "rétro-propagation des erreurs". L'apprentissage (ou construction du modèle) est automatique ;
- Résistance au bruit ou au manque de fiabilité des données;
- Simple à manier, beaucoup moins de travail personnel à fournir que dans l'analyse statistique classique. Aucune compétence en mathématiques, informatique statistique requise ;

-Comportement moins mauvais en cas de faible quantité de données;

-Pour l'utilisateur novice, l'idée d'apprentissage est plus simple à comprendre que les complexités des statistiques multi-variables.

II.12.2. Inconvénients des réseaux de neurones

-L'absence de méthode systématique permettant de définir la meilleure topologie du réseau et le nombre de neurones à placer dans la (ou les) couche(s) cachée(s);

-Le choix des valeurs initiales des poids du réseau et le réglage du pas d'apprentissage, qui jouent un rôle important dans la vitesse de convergence;

-Le problème du sur-apprentissage (apprentissage au détriment de la généralisation);

-La connaissance acquise par un réseau de neurone est codée par les valeurs des poids synaptiques, les réseaux de neurones sont donc des boîtes noires où les connaissances sont intelligibles pour l'utilisateur.

II.13 Conclusion :

Dans ce chapitre, nous avons exposé les réseaux de neurones, en spécifiant leurs définitions, les concepts de base de cette technique, l'apprentissage des réseaux de neurones, et ses différents domaines d'utilisation.

CHAPITRE III

La Commande Des Processus Non-Linéaires Par Les Techniques RNA – PSO ET SA

III.1 Introduction

Prévision a toujours été un défi crucial pour les organisations qui jouent un rôle important dans la prise de nombreuses décisions cruciales. Beaucoup d'efforts ont été consacrés au cours des dernières décennies pour développer et améliorer les modèles de prévision de séries chronologiques. Dans ces modèles, la plupart des chercheurs ont supposé une relation linéaire entre les valeurs passées de la variable prévisions. Bien que l'hypothèse linéaire, il est plus facile de manipuler les modèles mathématiques, il peut conduire à une représentation inadéquate de nombreux modèles du monde réel dans lequel relation non linéaire est répandue. L'optimisation est un sujet central en recherche opérationnelle, un grand nombre de problèmes d'aide à la décision pouvant être décrits sous la forme de problèmes d'optimisation.

Les problèmes d'identification, l'apprentissage supervisé de réseaux de neurones ou encore la recherche du plus court chemin est, par exemple, des problèmes d'optimisation.

III.2 Notions de base

- **Une solution:** est une affectation de toutes les variables du problème.
- **Une solution optimale:** est une solution de coût minimal (ou maximal).
- **Un mouvement :** est une opération élémentaire permettant de passer d'une solution a une solution voisine (exemple : changer la valeur d'une variable, échanger deux variables).
- **Le voisinage :** d'une solution est l'ensemble des solutions voisines, c'est-à-dire l'ensemble des solutions accessibles par un mouvement (et un seul).

- **Un essai** : est une succession de mouvements.
- **Une recherche locale** : est une succession d'essais.

III.3 Méthodologie de conception

L'optimisation est souvent réduite aux techniques de résolution mathématiques auxquelles sont attribués par la suite les échecs rencontrés. Or, comme pour la plupart des problèmes que l'ingénieur est amené à résoudre, l'optimisation doit faire l'objet d'une démarche qui comporte quatre phases récapitulées dans la figure I.1. Les phases peuvent s'enchaîner séquentiellement mais les itérations et les retours sont bien souvent indispensables.

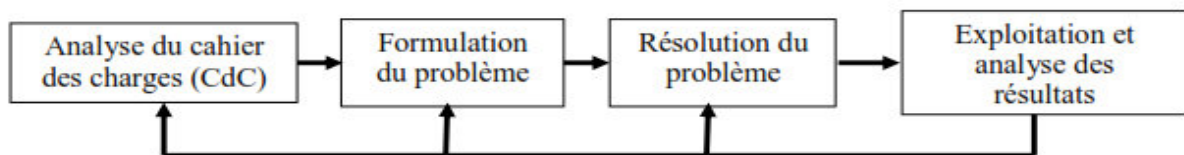


Figure. III.1.Démarche de résolution d'un problème de conception

III.3.1 Analyse du cahier des charges

Le cahier de charge, définit en amont, exprime les besoins des utilisateurs en termes de fonction de service et contraintes à satisfaire. Des exemples de cahier des charges pour divers dispositifs électrotechnique sont présentés dans cette partie au paragraphe I.6. Par exemple, il précise le rendement, l'encombrement, la maintenance et la robustesse de la chaîne de traction électrique d'un moteur synchrone à aimants permanents à commutation électronique.

A l'état initial, il est décidé que l'objet à concevoir doit assurer certaines fonctions dans un environnement donné : les fonctions de service. Il doit également satisfaire certaines exigences qui conditionnent son adoption par l'utilisateur final : les contraintes.

En général, l'objet à concevoir interagit avec son environnement. Un ou plusieurs phénomènes physiques sont nécessaires pour décrire ces interactions et constituent les modèles physiques comportementaux de l'objet.

Pour évaluer les performances de l'objet et vérifier qu'il répond aux exigences imposées, il est nécessaire de définir un ensemble de réponses de contrôle ainsi qu'un ensemble de facteurs de lesquels il est possible d'agir. Facteurs et réponses sont liés à la nature et au comportement de l'objet ainsi qu'à la réaction de l'environnement.

La phase de rédaction du cahier des charges impose une caractérisation rigoureuse du dispositif à concevoir et n'est pas traitée ici. Elle peut s'appuyer sur des méthodologies et technique de management.

III.3.2 Formulation du problème d'optimisation

Cette phase consiste à traduire le problème de conception, décrit par le cahier des charges, en un problème mathématique équivalent .C'est l'étape la plus délicate du processus de conception car, là aussi, la formulation d'un problème n'est jamais unique, en particulier la définition des fonctions caractérisant les performances du système.

Elle consiste à définir de façon précise [19] :

1. La fonction objective.
2. Les paramètres de conception.
3. Les éventuelles contraintes liées à la fabrication ou à l'utilisation du dispositif et donc exprimées dans le cahier des charges.
4. Les contraintes ajoutées par le concepteur.

La fonction objectif est une des réponses de l'objet qui définit l'objectif à atteindre et peut être de deux nature : un cout à minimiser (cout de fabrication, consommation, cout d'exploitation, durée de développement) ou une performance à maximiser (profit, rendement, facteur de transmission).Son choix conditionne la définition du problème d'optimisation et inclus les moyens qui en permettent le calcul, c'est-à-dire la modélisation retenue pour l'objet.

Dans le cas d'un objectif unique, le choix de cette fonction est évident. Par exemple, dans le cas ou le but est de trouver les caractéristiques d'un dispositif produisant des performances dont les valeurs sont spécifiées, la fonction objectif peut prendre comme expression l'écart entre les performances et les spécifications.

Cependant, les problèmes d'optimisation doivent souvent satisfaire des objectifs multiples dont certains sont concurrents. Plusieurs façons de traiter ces problèmes sont analysées dans ce chapitre.

Les paramètres ou variable de conception sont des facteurs contrôlés qui permettent d'influencer les performances. Ils peuvent être de nature diverses : dimensions géométriques, propriétés des matériaux, choix structurels, etc. Ils peuvent être quantitatifs ou qualitatifs, continus ou discrets. Le choix et le nombre des paramètres conditionnent aussi la définition du problème d'optimisation. Il peut être intéressant de faire varier un grand nombre de facteurs afin d'augmenter l'espace recherche mais le processus d'optimisation sera alors plus long.

Des contraintes peuvent être ajoutées par le concepteur pour avoir, par exemple, une forme géométrique convenable, s'assurer de la validité de la modélisation retenue et de son bon fonctionnement, etc.

La formulation de problème d'optimisation est fondamentale dans le processus de conception parce qu'elle conditionne le succès des étapes suivantes. Elle n'est pas facile à aborder car le choix des variables de conception n'est jamais unique et les moyens de calcul actuels ne peuvent en gérer qu'un nombre limité.

III.3.3 Résolution du problème d'optimisation

La recherche de l'optimum d'un problème est réalisée à l'aide de méthodes d'optimisation qui seront présentées dans la deuxième partie.

Certaines de ces méthodes sont dites déterministes car elles conduisent, pour une solution initiale donnée, toujours au même résultat final. Pour trouver l'optimum, elles s'appuient sur une direction de recherche qui peut être fournie par les dérivées de la fonction objective.

Ces méthodes ont la réputation d'être efficaces lorsque la solution initiale est proche de l'optimum recherché. Cette particularité constitue un inconvénient majeur dans le cas d'une fonction objectif possédant plusieurs optimums. Elles peuvent, en effet, converger vers un optimum local.

Les méthodes stochastiques, contrairement à la plupart des méthodes déterministes, ne nécessitent ni point de départ, ni à la connaissance du gradient de la fonction objectif pour atteindre la solution optimale. Elles s'appuient sur des mécanismes de transition probabilistes et aléatoire qui explorent efficacement l'espace de recherche et convergent vers l'optimum global. Leur nature aléatoire implique que plusieurs exécutions successives de ces méthodes conduisent à des résultats différents pour une même initialisation du problème d'optimisation. Cependant, elles demandent un nombre important d'évaluation de la fonction objectif en comparaison avec les méthodes déterministes exploitant la dérivée de la fonction objectif

III.3.4 Analyse et exploitation des résultats

Une fois le problème résolu, il est impératif d'évaluer la qualité de la solution et en cas d'échecs de s'interroger sur les choix adoptés lors des phases précédentes. On attribue souvent l'échec de l'optimisation à la méthode de recherche employée pour la localisation de l'optimum ou à la sensibilité des paramètres de cette méthode alors que le problème est peut être mal formulé.

Un cahier des charges peut être non faisable à cause de contraintes trop sévères ou parce que la fonction objectif n'est pas pertinente. Par exemple, la réduction des oscillations de couple doit être faite en ajoutant une contrainte pour maintenir le couple moyen. Il se peut que le problème soit mal formulé par manque de contraintes qui n'apparaissent pas explicitement dans le cahier des charges. Il est alors nécessaire de les ajouter. Ce cas s'est présenté dans les exemples traités au paragraphe III.1. Le choix de méthode de résolution peut être effectivement erroné et il est nécessaire de s'assurer de l'adéquation entre la méthode de résolution et le modèle retenu. En effet, si le modèle n'est pas continu et différentiable ou si l'expression de la fonction objectif n'est pas explicite, il est imprudent d'utiliser les méthodes déterministes du premier ordre, c'est-à-dire qui utilisent les dérivées premières de la fonction objectif.

III.4.Caractéristiques

III.4.1.Sensibilité et robustesse d'une méthode d'optimisation

La méthode d'optimisation est conditionnée par des paramètres de contrôle et des conditions initiales (valeurs initiale des initiales des variables de conception, valeurs paramètres de contrôle,..).Elle peut être caractérisée selon le modèle de la boite noire illustré en figure III.2

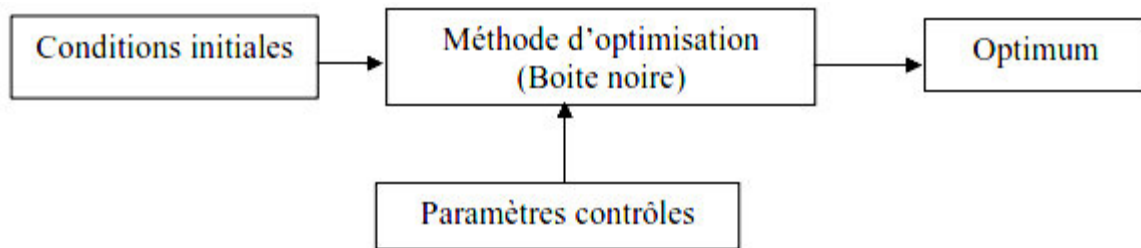


Figure III .2 Méthodes d'optimisation (Boite noire)

L'efficacité d'une méthode d'optimisation est liée à la sensibilité et à la robustesse par rapport aux paramètres de contrôle et aux conditions initiales. Lorsque les variables de conception doivent prendre une valeur bien précise pour que la méthode de résolution converge vers l'optimum d'une fonction donnée, la méthode est dite sensible aux conditions initiales .Une méthode d'optimisation est robuste si pour une même valeur des paramètres de contrôle et des conditions initiales, elle est capable de trouver l'optimum de fonctions très différentes [12].

Une méthode parfaite devrait être totalement insensible aux conditions initiales et aux variables de conception et converger vers l'optimum quelles que soient la fonction objective et les contraintes.

III.4.2.Opérateurs de recherche fondamentaux

La recherche de l'optimum d'une fonction est généralement réalisée à l'aide de deux opérateurs fondamentaux : l'exploration et l'exploitation.

L'exploration permet une localisation imprécise de l'optimum global alors que l'exploitation affine cette solution en augmentant la précision de l'optimum.

Le succès et l'efficacité d'une technique de résolution dépendent la plupart du temps d'un compromis entre l'exploration et l'exploitation. Certaines méthodes toutefois n'utilisent qu'un seul de ces opérateurs pour parvenir à l'optimum. Ainsi, les méthodes déterministes, exploitant les dérivées de la fonction objectif et des contraintes pour atteindre rapidement et précisément le minimum local le plus proche du point de départ, privilégient l'exploitation au détriment de l'exploration.

Tout algorithme d'optimisation doit utiliser ces deux stratégies pour trouver l'optimum global : l'exploiter la connaissance acquise aux points déjà visités et ainsi trouver des points meilleurs. Ces deux exigences peuvent paraître contradictoires mais un bon algorithme de recherche doit trouver le bon compromis entre les deux. Une recherche purement aléatoire est bonne pour l'exploration mais pas pour l'exploitation alors que la recherche dans le voisinage est une bonne méthode d'exploitation mais pas d'exploration.

III.4.3. Mode de recherche de l'optimum

Lorsque l'évolution de la méthode de résolution est prévisible et ne laisse aucune place au hasard, celle-ci est qualifiée de déterministe. En revanche les méthodes dites stochastiques s'appuient sur des mécanismes de transition probabiliste qui peuvent conduire à des résultats différents pour des conditions initiales et des paramètres de contrôle identiques.

Les méthodes déterministes sont qualifiées de méthodes locales, c'est-à-dire qu'elles convergent vers un optimum dépendant uniquement des points de départ, qu'il soit local ou global. A l'opposé, les techniques stochastiques sont reconnues comme des méthodes globales qui permettent de localiser l'optimum global.

III.4.4 Ordre d'une méthode de résolution

Les méthodes de résolution peuvent être classées à partir de leur ordre selon qu'elles nécessitent ou non le calcul des dérivées de la fonction objectif et des fonctions contraintes par rapport aux paramètres. Une méthode est dite d'ordre zéro si elle utilise uniquement la connaissance de la fonction elle-même. Elle est d'ordre un si elle requiert

le calcul des dérivée premières et d'ordre deux s'il lui faut aussi accéder aux dérivée secondes.

Les méthodes d'ordre zéro sont en général peu précises et convergent plus lentement vers l'optimum. En revanche, elles offrent l'avantage d'éviter le calcul du gradient, ce qui est intéressant lorsque la fonction n'est pas différentiable ou que le calcul de son gradient représente un coût important. C'est notamment le cas des méthodes qui ne donnent que le gradient aux points finis.

Les méthodes d'ordre un permettent d'accélérer la localisation de l'optimum, puisque le gradient donne l'information sur la direction de l'amélioration. Par contre elles sont applicables seulement aux problèmes où les fonctions objectif et contraintes sont continuellement différentiables.

III.5 Processus d'optimisation

III.5.1 Variables du problème

C'est à l'utilisateur de définir les variables du problème. Il peut avoir intérêt à faire varier un grand nombre de paramètres pour augmenter les degrés de liberté de l'algorithme afin de découvrir des solutions nouvelles. Autrement, s'il a une vue suffisamment précise de ce qu'il veut obtenir, il peut limiter le nombre de variables à ceux qui sont essentielles. [25].

III.5.2 Espace de recherche

L'espace de recherche est infini, seule la population initiale est confinée dans un espace fini. Mais dans le cas des algorithmes de type Monte Carlo et génétique, il est généralement nécessaire de définir un espace de recherche fini. Cette limitation de recherche n'est généralement pas problématique. En effet, ce n'est pas des raisons technologiques ou informatiques, les variables sont en général naturellement limitées. [25]

III.5.3 Fonction objectif

Un algorithme d'optimisation nécessite généralement la définition d'une fonction rendant compte de la pertinence des solutions potentielles, à partir de la grandeur à

optimiser. C'est la fonction d'adaptation f (ou *fitness function*). L'algorithme convergera vers un optimum de cette fonction, quelle que soit sa définition. La fonction f doit donc exprimer le plus fidèlement possible le désir de l'utilisateur sous forme mathématique.

III.5.4 Problème d'optimisation

Un problème d'optimisation au sens générale est défini par un ensemble de solutions possibles S , dont la qualité peut être décrite par une fonction objectif f .

On cherche alors à trouver la solution S^* possédant la meilleure qualité $f(S^*)$ par la suite on peut chercher à minimiser ou maximiser $f(S)$.

Un problème d'optimisation peut présente des contraintes d'égalité ou d'inégalité, si $f(S)$ change avec le temps ou encore multi-objectif si plusieurs fonction objectifs doivent être optimisées.

Il existe des méthodes déterministes permettant de résoudre certains problèmes en temps fini ;

Ces méthodes nécessitent généralement un certain nombre de caractéristique de la fonction objectif, comme la stricte convexité, la continuité ou encore la dérivabilité. On peut citer comme exemple la méthode programmation linéaire quadratique ou dynamique, la méthode du gradient, la méthode de Newton, ...etc. [24].

III.6 Optimisation par essaim de particules

Essaim de particules a été l'origine en 1995 par Kennedy et Eberhart (1995), après avoir étudié le comportement social des oiseaux. Selon à ce que les scientifiques ont découvert, pour chercher de la nourriture, chaque membre d'un groupe d'oiseaux qui détermine sa vitesse en fonction de leur expérience personnelle ainsi que l'information obtenue grâce à l'interaction avec d'autres membres du troupeau. Cette idée a été le principe de base pour optimisation par essaim de particules (PSO). Chaque oiseau, appelé particules, vole dans l'espace des solutions du problème d'optimisation la recherche de la meilleure solution, et donc sa position représente une solution potentielle au problème.

En essaim de particules la terminologie, les solutions disponibles dans chaque itération sont appelés l'essaim "ce qui équivaut à la population" en génétique algorithmes.

III.6.1 Méthode de base

La procédure de la norme PSO se résume comme suit [15]:

Étape 1: Initialiser une population de particules à des positions aléatoires et des vitesses, où chaque particule contient D variables.

Étape 2: chaque particule et sa valeur objective égal à son courant la position et la valeur objective, et soit g Évaluer les valeurs objectives de toutes les particules. Soit p et sa valeur objective être égale à la position et l'objectif valeur de la meilleure particule initiale.

Étape 3: Mise à jour de la vitesse et de la position de chaque particule.

Étape 4: Évaluer les valeurs objectives de toutes les particules.

Étape 5: Pour chaque particule, de comparer sa valeur actuelle objectif. Si la valeur actuelle vaut mieux, puis mettre à jour p avec la valeur objective de ses p meilleur et sa valeur objective avec la position actuelle et la valeur objective.

Étape 6: Déterminer la meilleure particule de l'essaim en cours avec la meilleure valeur objective. Si la valeur objective est meilleure que la valeur objective de g, Mise à jour g et son objective valeur de la position et de la valeur objective de la courante meilleure particule.

Étape 7: Si un critère d'arrêt est remplie, veuillez indiquer g de sortie et ses valeur objective, sinon, passez à l'étape 3.

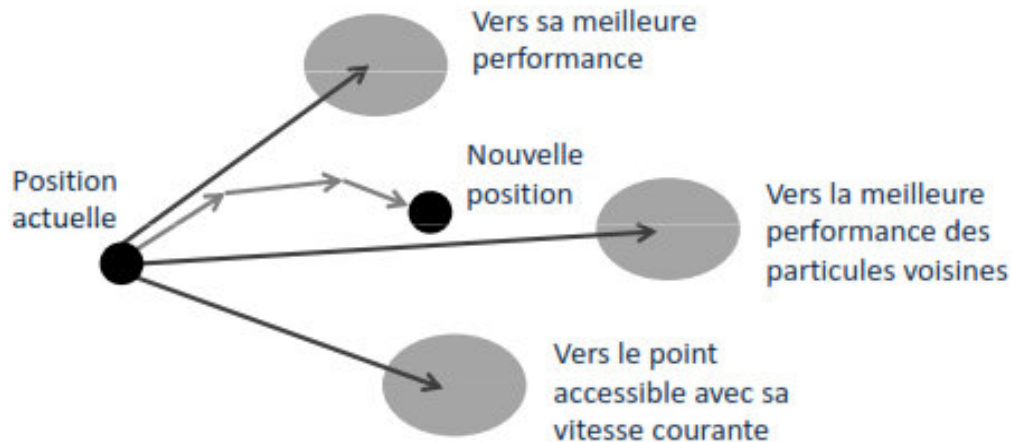


Figure III.3 Le schéma de principe du déplacement d'une particule [1].

Pour réaliser son prochain mouvement, chaque particule combine trois tendances : suivre sa vitesse propre, revenir vers sa meilleure performance, aller vers la meilleure performance de ses informatrices.

III.6.2 Formulation générale

La formulation générale de ce comportement est la suivante [1] [5] [6]

$$v(t + 1) = \lambda_1 v(t) + \lambda_2 (p_i(t) - x(t)) + \lambda_3 (p_g(t) - x(t)) \quad (\text{III.1})$$

$$x(t + 1) = x(t) + v(t + 1) \quad (\text{III.2})$$

Ou t est le temps, x la position de la particule, v sa vitesse, p_i sa meilleure position atteinte, p_g le meilleur des meilleures positions atteintes dans son voisinage et $\lambda_1, \lambda_2, \lambda_3$ les trois coefficients de confiance pondérant les trois directions possibles, sont choisis à chaque pas de temps au hasard dans un intervalle donné.

III.6.3 Algorithme de base

[Initialisation]

Initialiser aléatoirement la population

[Traitement]

Répéter

 Pour chaque particule

 Evaluer la valeur de fitness

 Modifier la vitesse en utilisant l'équation (1)

 Puis déplacer en utilisant l'Equation (2)

 Fin pour

Jusqu'à ce que (le processus converge)

Algorithme III.1 algorithme de base de l'OEP

L'algorithme s'exécute tant qu'un critère de convergence n'a pas été atteint. Cela peut être [11] :

 Un nombre fixe d'itération ;

 En fonction de la fitness ;

 Lorsque la variation de vitesse est proche de 0.

III.6.4 Paramètres de l'algorithme

L'algorithme OEP comprend plusieurs paramètres de réglage qui permettent d'agir sur le compromis exploration – exploitation.

L'exploration est la capacité de tester différentes régions de l'espace à la recherche de bonnes solutions candidates.

L'exploitation est la capacité de concentrer la recherche autour des solutions prometteuses afin de s'approcher le plus possible de l'optimum [5].

Parmi ces paramètres, on peut citer

1. Le nombre de particules ;
2. Les valeurs des coefficients λ ;

3. La taille et la définition du voisinage ;
4. La vitesse maximale.

Voisinage

Le voisinage constitue la structure du réseau social. Les particules à l'intérieur d'un voisinage communiquent entre-elles. Différents voisinages ont été étudiés [5].

- Topologie en étoile : le réseau social est complète, chaque particule est attirée vers la meilleure particule et communique avec les autres.
- Topologie en anneau : chaque particule communique avec n voisines immédiates. Chaque particule tend à se déplacer vers le meilleur dans son voisinage local.
- Topologie en rayon : une particule « centrale » est connectée à toutes les autres. Seule cette particule centrale ajuste sa position vers la meilleure, si cela provoque une amélioration l'information est propagée aux autres.

Vitesse maximale

Pour éviter que les particules se déplacent trop rapidement d'une région à une autre dans l'espace de recherche, on fixe une vitesse maximale (n'est pas obligatoire).

De nombreux problèmes, très variés, peuvent se mettre sous la forme d'une fonction numérique dont on doit rechercher le minimum. Dès lors, une technique comme l'optimisation par essaim particulaire (OEP) permet de les traiter de manière uniforme, que l'espace de recherche soit discret, continu ou mixte, avec des performances souvent honorables et parfois excellentes.

L'OEP (PSO) est une "métaheuristique" d'optimisation, qui simule le comportement social d'animaux évoluant en essaims. L'algorithme reprend les principes fondamentaux de l'OEP, tout en s'affranchissant d'un des principaux défauts des métaheuristicues : le réglage, demandé à l'utilisateur, des paramètres de l'algorithme. L'algorithme PSO exploite à cet effet des règles. L'algorithme PSO présente le risque d'une convergence trop rapide, qui peut se traduire par un piégeage de la méthode dans un optimum local. Pour remédier à ce problème, des chercheurs ont introduit une stratégie de déplacement des particules, qui repose sur un algorithme à estimation de distribution.

Cette technique permet de renforcer le caractère explorateur et d'éviter les convergences prématurées. Cette stratégie de déplacement des particules améliore sensiblement les performances de l'algorithme PSO

III.6.5 Principe

Le principe de l'OEP s'est éloigné du comportement (trop complexe) des animaux, pour ne conserver qu'une modélisation basées sur des agents simples, appelés particules. Un essaim de particules, qui sont des solutions potentielles au problème d'optimisation, « survole » l'espace recherche, à la recherche de l'optimum global. Le déplacement d'une particule est influencé par trois composantes :

- Une composante d'inertie : la particule tend à suivre sa direction courante de déplacement ;
- Une composante cognitive : la particule tend à se fier à sa propre expérience et, ainsi, à se diriger vers le meilleur site par lequel elle est déjà passée ;
- Une composante sociale : la particule tend à se fier à l'expérience de ses congénères et, ainsi, à se diriger vers le meilleur site déjà atteint collectivement par l'essaim.

III.7 Commande des processus par réseaux de neurones

La littérature scientifique fait mention de différentes architectures de commande. Les plus simples se basent sur l'apprentissage d'un contrôleur conventionnel déjà existant, d'autres opèrent un apprentissage hors-ligne du modèle inverse du processus ou d'un modèle de référence et enfin, d'autres travaillent complètement en ligne [13][14].

III.7.1 Apprentissage d'un contrôleur conventionnel

Un réseau de neurones peut reproduire le comportement d'un contrôleur conventionnel déjà existant (PI, PID, RST, ...) grâce à ses facultés d'apprentissage et d'approximation. Il suffit de le soumettre à un apprentissage hors ligne pendant une phase

d'identification directe en considérant que le contrôleur est lui-même un processus. La figure III.3 montre le principe de l'identification directe d'un contrôleur conventionnel.

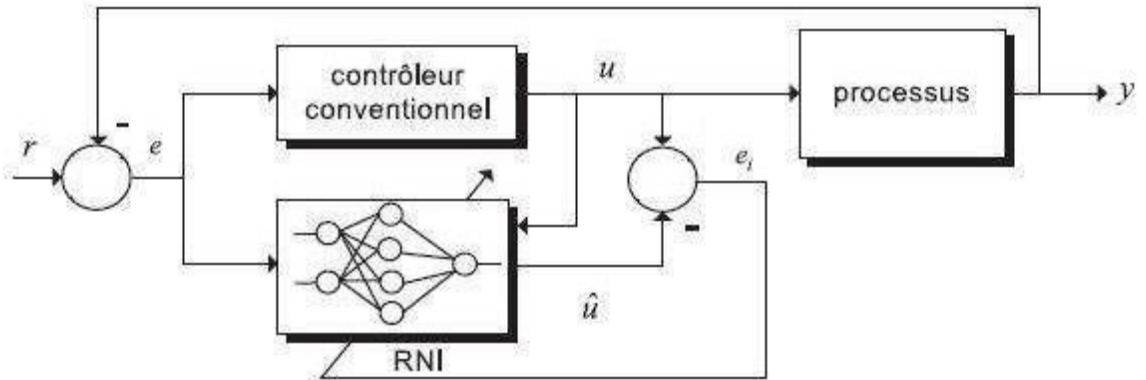


Figure III.4: Schéma d'identification directe d'un contrôleur conventionnel avec un RNI

Le but de cette architecture n'est pas de perfectionner les performances du contrôleur conventionnel déjà existant, mais de s'affranchir des contraintes d'implémentations matérielles que peuvent nécessiter certains régulateurs. La méthode de régulation de type RST par exemple est reconnue pour ses bonnes performances en commande mais elle pose de sérieux problèmes en intégration numérique.

Dans ce travail, nous avons conçu un algorithme hybride (PSO-SA) qui combine la capacité d'exploration de PSO avec la capacité d'exploitation de la SA, et qui est capable de prévenir convergence prématurée. Par rapport à d'autre algorithme [30] sur les fonctions de référence bien connus et pour le problème de la réduction de la consommation d'énergie et stabilité dans les systèmes, PSO-SA effectue bien en termes de précision, vitesse de convergence, de stabilité et de robustesse à l'avenir, nous allons également comparer les performances.

III.8 Commande par le recuit simulé.

SA est une probabiliste variante de la méthode de recherche locale, qui peut, à la différence de PSO, s'échapper des optima locaux. SA est basée sur une analogie prise de la thermodynamique, qui est le suivant [19]:

Afin de développer un cristal, on commence par chauffer le matériau jusqu'à ce qu'il atteigne sa fusion Etat. Ensuite, on réduit la température de cette masse en fusion cristalline progressivement, jusqu'à ce que la structure cristalline soit formée. Une norme Procédure SA débute par la génération d'une solution initiale à aléatoire.

Au début, un petit changement aléatoire est fait dans la solution courante. Ensuite, la valeur de la fonction objectif de la nouvelle solution s_n est calculée et comparée à celle de la solution en cours. Un déplacement est effectué à la nouvelle solution si il a une meilleure valeur ou si la fonction de probabilité en œuvre dans SA a une valeur supérieure à un nombre généré de façon aléatoire. Sinon, une nouvelle solution est générée et évalués. La probabilité d'acceptation d'une nouvelle solution est donnée comme suit [30]:

$$\text{Si } f(S1) - f(Sc) < 0 \dots\dots\dots \dots\dots (III. 3)$$

$$P = \begin{cases} 1 \\ \exp\left(\frac{-|f(S_n) - f(S_c)|}{T}\right) \dots\dots\dots \dots\dots \dots\dots (III. 4) \end{cases}$$

Le calcul de cette probabilité repose sur un paramètre T, qui est considéré comme la température, car il joue un semblable rôle que la température dans le processus physique de recuit. A éviter de se faire piéger à un point minimum local, le taux de la réduction doit être lente. Dans notre problème, nous utilisons la suivant la méthode pour réduire la température: $\gamma = 0,99$ [30].

$$T_{i+1} = \gamma T_i \dots\dots\dots \dots\dots (III.5)$$

Ainsi, au début de la plupart des mouvements SA aggravation peut être accepté, mais à la fin seulement les améliorer sont susceptibles d'être autorisé, ce qui peut aider à la procédure de sauter hors d'un local minimum. L'algorithme peut être résilié après un

certain la fraction volumique de la structure a été atteinte ou après un prédéfinie exécution.

III.9 Hybridation PSO-SA

Cette section présente un nouveau modèle hybride PSO-SA algorithme qui combine les avantages des deux PSO (qui a une mondiale forte capacité de recherche) et SA (qui a une forte capacité de recherche locale). D'autres applications de PSO et hybride SA algorithme peut être trouvé.

L'approche hybride permet d'utiliser pleinement l'exploration la capacité à la fois de PSO et SA et compense les faiblesses de chaque. Par conséquent, grâce à l'application de la SA aux PSO, l'algorithme proposé est capable d'échapper à un local optimale. Toutefois, si SA est appliquée aux PSO à chaque itération, le coût de calcul augmente fortement et au même moment où la capacité de convergence rapide des PSO peuvent être affaiblis. Afin d'intégrer de manière flexible avec PSO SA, SA est appliquée PSO à toutes les itérations K si aucune amélioration de meilleure solution ne se produisent.

Par conséquent, l'approche hybride PSO-SA est capable de maintenir une convergence rapide (la plupart du temps) grâce aux opérations, et d'échapper à un optimum local avec l'aide de la SA. Afin de permettre PSO sauter d'un local optimale, SA est appliquée à la meilleure solution dans l'essai.

L'hybride PSO-SA algorithme fonctionne comme illustré dans Algorithme, où l'on a ce qui suit.

III.9.1 La description d'une particule.

Chaque particule (solution) $X \in S$ est représenté par ses composantes $n > 0$, c'est-à-dire $X = (x_1, x_2, \dots, X)$, Où $i = 1, 2, \dots, n$ et n représente la dimension du problème d'optimisation à résoudre.

III.9.2 Swarm (Particules) initiale.

Swarm initiale correspond à la population de particules qui vont évoluer. Chaque particule x est initialisée avec la valeur aléatoire uniforme entre les limites inférieure et supérieure de l'intervalle définissant le problème d'optimisation [30].

III.9.3 Evaluer la fonction.

Évaluer la fonction (ou fitness) dans PSO-SA algorithme est typiquement l'objectif fonction que nous voulons minimiser dans le problème.

Il sert pour chaque solution à tester pour convenance à l'environnement en cours d'examen.

III.9.4 L'algorithme SA.

Si aucune meilleure amélioration de la solution globale produire au cours des itérations K dernières, puis cela signifie que l'algorithme est piégé dans un local optimal point. Pour échapper à partir optimum local, nous appliquons l'algorithme de SA de la meilleure solution globale. Le rendement de SA dépend de la définition de plusieurs paramètres de commande.

```

(1)  $iter \leftarrow 0$ ,  $cpt$  initialise la taille  $swarm\_size$ 
(2)  $stop\_criterion \leftarrow$  nombre maximum d'évaluations de la fonction ou
(3) while Not  $stop\_criterion$  do
(4)   for chaque particule  $i \leftarrow 1$  to  $swarm\_size$  do
(5)     Evaluer (particule(i)) if la valeur de remise en forme est meilleure que la meilleure
valeur de remise en forme( $cbest$ ) dans l'histoire
(6)       Mettez à jour la valeur actuelle comme nouvelle  $cbest$ 
(7)     end
(8)   end
(9) Choisissez la particule avec la meilleure valeur de fitness à proximité

(10) for chaque particule  $i \leftarrow 1$  to  $swarm\_size$  do
(11) Mettre à jour la vitesse des particules selon l'équation(3)
(12) Appliquer les limites de vitesse
(13) Mettre à jour la position des particules selon l'équation(4)
(14) Appliquer les limites de particules
(15) end
(16) if il n'y a pas d'amélioration de la meilleure solution globale then
(17)    $cpt \leftarrow cpt+1$ 
(18) end
(19) Mise à jour globale meilleure solution
(20)  $cpt \leftarrow 0$ 
(21) if  $cpt = \mathcal{K}$  then
(22)    $cpt \leftarrow 0$ 
(23) Appliquer SA meilleure solution globale
(24)    $iter_{SA} \leftarrow 0$  initialiser T conformément à l'équation(5)
(25)      $current\_solution \leftarrow global\_best\_solution$ 
(26)      $current\_cost \leftarrow$  Evaluer  $current\_solution$ 
(27)     while Not  $SA\_stop\_criterion$  do
(28)       while inner-loop  $stop\_criterion$  do
(29)          $Neighbor \leftarrow$  Générer  $current\_solution$ 
(30)          $Neighbor\_cost \leftarrow$  Evaluer  $Neighbor$ 
(31)         if  $Accept(current\_cost, Neighbor\_cost, T)$  then
(32)            $current\_solution \leftarrow Neighbor$ 
(33)            $current\_cost \leftarrow Neighbor\_cost$ 
(34)         end
(35)          $iter_{SA} \leftarrow iter_{SA} + 1$ 
(36)         Mettre à jour ( $global\_best\_solution$ )
(37)       end
(38)       Mettre à jour (T) conformément à l'équation(4)
(39)       Mettre à jour ( $SA\_stop\_criterion$ )
(40)     end
(41)   end
(42)    $iter \leftarrow iter + 1$  Mettre à jour ( $stop\_criterion$ )

```

Algorithme III.2 Algorithme de l'hybride PSO-SA [30]

III.10 Conclusion

Afin d'examiner l'adaptation de l'une des métaheuristiques hybride modernes qui est l'optimisation par essaim de particules et le recuit simulé et les réseaux de neurones aux problèmes d'optimisation pour la commande d'un procédé non linéaire, nous appliquerons dans le chapitre suivant, ces méthodes sur le problème de lévitation magnétique.

CHAPITRE IV

Simulation de la Commande d'un Système De Lévitation Magnétique

IV.1 Introduction

La période actuelle est caractérisée par de multiples exigences de la réalisation de moyens de transport possédant des systèmes mécaniques de qualité élevée, de grandes vitesses et assurant confort et fiabilité de fonctionnement. Ils sont gérés par des systèmes de contrôle intelligents faisant appel à des technologies industrielles diverses. Les développements actuels concernant la commande dans la lévitation magnétique sont le transport, les paliers magnétiques, les actionneurs spéciaux, les organes hépatiques, etc....

Non seulement les paramètres d'un contrôleur doivent être estimés d'une manière optimale pour un régime de fonctionnement donné d'un procédé tel qu'un système de lévitation électromagnétique (MLS) qui est train Maglev [28], qui est un système mécatronique fortement non-linéaire, mais ils doivent assurer une optimalité globale (métaheuristique). Dans notre étude, nous avons utilisé une métaheuristique hybride qui a la capacité d'estimer la valeur optimale globale des paramètres (poids du réseau de neurones artificiel) du contrôleur du modèle théorique de Maglev. La commande hybride [29] étudiée est définie par deux approches. La première est l'utilisation de la PSO (optimisation par essaim de particules) pour l'optimisation des paramètres d'un régulateur PID classique (l'hybridation d'une PSO avec le recuit simulé 'SA' , combinant les avantages de ces deux composants individuels, a été étudiée mais l'approche a été envisagée. La deuxième approche est l'optimisation des paramètres (poids) d'un contrôleur neuronale de type MLP par la PSO afin de commander d'une manière optimale le procédé. Des simulation de la commande de ce train à lévitation magnétique est développée On présente d'abord le modèle simple du système de lévitation électroaimant-balle en acier qui donne une idée de base sur sa commande puis on effectue la simulation du contrôle pour le système de lévitation magnétique plus complexe (le train Maglev) à partir d'un modèle de référence déduit à partir d'équations physiques.

IV.2 Etude du système de lévitation de la balle en acier

La lévitation électromagnétique représente une question de contrôle classique sujet pour lequel diverses solutions ont été développées. Beaucoup d'entre eux ont convergé vers une linéarisation par retour de tension de commande [28], Cette méthode nécessite un modèle mathématique très précis pour le système analysé. Différentes solutions pour un contrôle avancé ont été envisagées, chacune d'entre elles décrivent la complexité du problème étudié en raison des multiples effets non-linéaires et le rôle essentiel de ces effets dans la modélisation et la simulation du système.

Le système à lévitation magnétique (SLM) qu'on considère pour une étude de base préliminaire est constitué d'une balle en acier suspendue dans un champ magnétique et dont le contrôle de la lévitation se fait par la grandeur du courant dans la bobine. La figure IV.1 ci-dessous montre le banc d'essai du système.

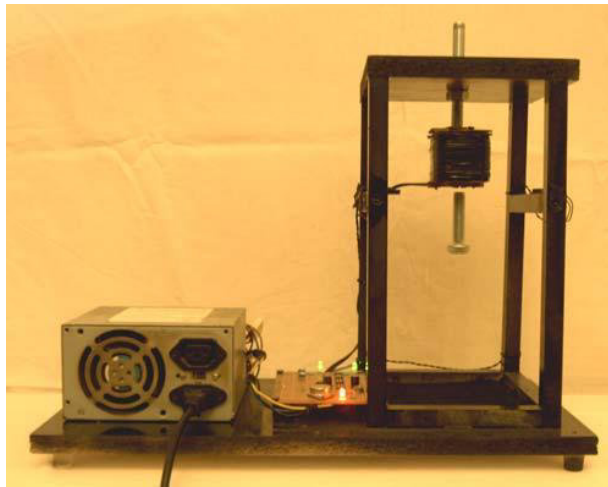


Figure IV.1 : Montage d'un système de sustentation magnétique d'une balle en acier

Ce système mécatronique est composé de sous-systèmes suivants:

- * L'actionneur électromagnétique représenté par la bobine 1 (une bobine à noyau ferromagnétique);
- * Le capteur de position (optique ou à rayons infrarouges) qui permet de mesurer la position de la bille métallique par rapport à la bobine;
- * Les circuits électriques pour l'alimentation, l'amplification et la commande e contrôle, etc.

La balle ferromagnétique possède deux degrés de liberté. L'analyse n'envisage que le mouvement de translation effectué dans le plan vertical, tout en négligeant la rotation de la balle autour de son propre axe. Le but du système conçu consiste à maintenir la balle à un niveau de référence qui est prédéfini.

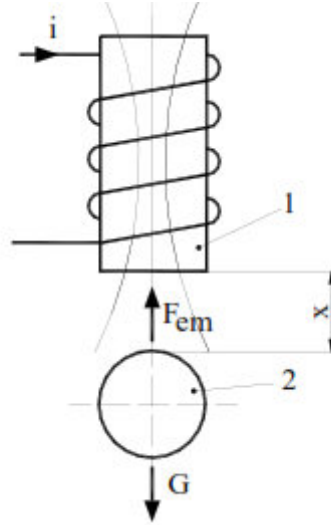


Figure IV.2 : Le système de Lévitation magnétique de la balle en acier

La balle est sous l'influence de deux forces:

- Le champ de gravité "G",
- La force de sustentation électromagnétique FEM produite par le champ électromagnétique généré dans la bobine.

On peut définir l'équilibre fondé sur les lois connues de base.

Le modèle mathématique du système de lévitation électromagnétique

On peut construire le modèle mathématique du système de sustentation en écrivant équations différentielles appropriées conformément à la mécanique typique et électrique principes. La façon dont les composants sont appréciés dans le mode approche peut conduire à des alternatives plus simples ou plus complexes.

La formule de l'équilibre énergétique au sein du système est le suivant:

$$dW_e = dW_{mec} + dW_t + dW_m \quad (4.1)$$

Le procédé de mise en lévitation magnétique proposé présente une technique très simple en principe, mais qui peut s'avérer délicate en pratique lorsque l'on prend en considération des perturbations externes agissant sur le système.

L'objectif principal de ce projet est de maintenir une charge pour que celle-ci reste en lévitation en se basant sur un modèle complet du système de lévitation ; en déterminant les paramètres qui caractérisent chaque élément qui constitue la plateforme de lévitation magnétique.

Le but alors de ce travail est de concevoir un compensateur à avance de phase afin de rendre le système plus performant au niveau de sa stabilité.

Pour simplifier la compréhension de la lévitation magnétique on présente dans la figure IV.3 le principe de la lévitation magnétique d'une balle en acier dans un champ magnétique.

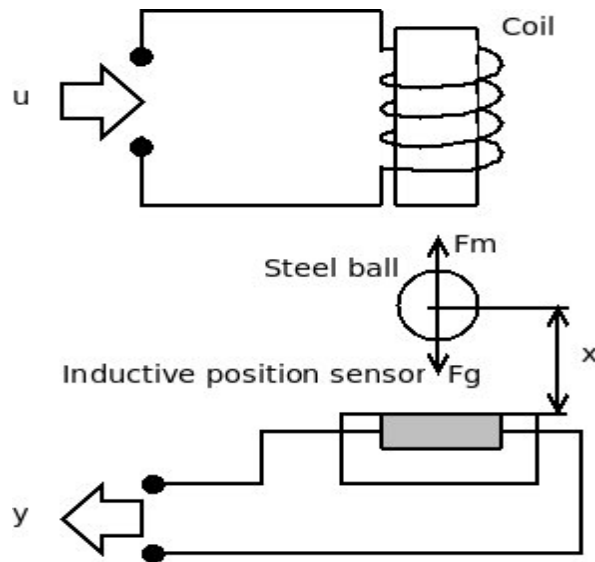


Figure. IV 3: Principe simplifié de la lévitation magnétique

Les équations du mouvement de la balle sont basées sur l'équilibre de toutes les forces, c'est-à-dire la force de gravité F_g , la force électromagnétique F_m et la force d'accélération F_a

$$F_a = F_m - F_g \quad (4.2)$$

$$F_g = mg \quad (4.3)$$

$$F_m = \frac{i^2 k_c}{(x - x_0)^2} \quad (4.4)$$

$$F_a = m \ddot{x} \quad (4.5)$$

Où :

i – courant dans la bobine [A]

x - position de la balle [m]

x_0 – position offset de la bobine [m]

k_c : constante de la bobine

m : masse de la balle [kg]

g : constante de gravité universelle [ms^{-2}]

Ce système peut être commandé en utilisant un contrôleur linéaire, par exemple le régulateur PID dont les paramètres sont ajustés par la méthode de Ziegler-Nichols, ou par un contrôleur non-linéaire (basé par exemple sur les systèmes de commande avec contraintes comme la commande optimale, etc...). Cependant, dans les deux cas, il est nécessaire de commencer avec l'identification de ce système non-linéaire instable en boucle ouverte (figure IV.4).

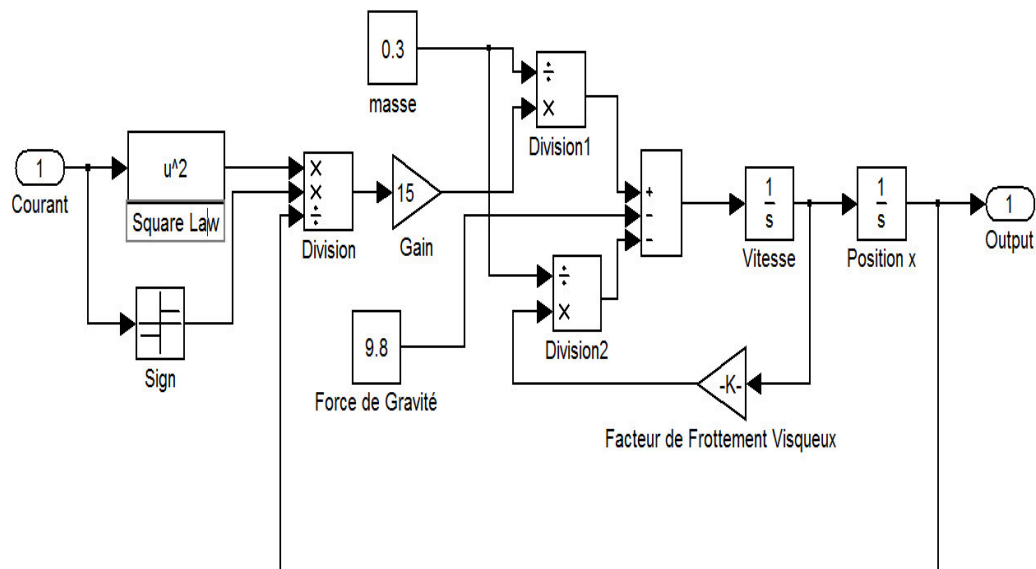


Figure IV.4 : Modèle d'un système de Lévitation Magnétique de la balle en acier Sous Matlab/Simulink

IV.3. Etude du système de lévitation magnétique du train Maglev

IV.3.1. Introduction La lévitation magnétique d'un train à grande vitesse qui est à support sans frottements utilise en contre-partie des moteurs à faible énergie. Le système étudié consiste en un électro-aimant qui est attiré vers un objet qui est fabriqué à partir d'un matériau électromagnétique. Le but de la commande est de maintenir l'entrefer entre le matériau et l'électro-aimant constant en contrôlant le courant dans la bobine et contrôler la distance de déplacement horizontal. La figure IV.6 montre le schéma physique du procédé non-linéaire étudié.

Lévitation magnétique est définie comme équilibre d'un corps sans contact dans la surface du solide par le résultat d'un champ magnétique. Trains de travail basé sur cette technologie est appelé comme trains à sustentation magnétique. Récemment, notre porte sont à rendre le système précis de la vitesse et de haute voyager si toutes les attentions en matière de systèmes à lévitation magnétique. Dans la dernière décennie le Japon et la Chine a développé le système Maglev pour les transports publics. Son principal avantage est son chemin sans frottement. La base système en boucle ouverte à sustentation magnétique est non linéaire de sorte qu'ils ont développé une plate-forme un excellent contrôle pour le contrôle et l'évaluation.

IV.3.2. HISTOIRE DU MAGLEV



Figure IV.5 : Le train Maglev

- L'histoire du Maglev [33] a commencé en 1962, avec les premières recherches sur les moteurs linéaires. Mais ce n'est que dans le courant des années 70 que tout débute réellement : l'étude sur la lévitation à l'aide de supraconducteurs commence formellement.
- En fait dès 1972, le mode de propulsion LSM " Linear Synchronous Motor " est expérimenté sur le modèle LSM200, un modèle de train qui utilise les supraconducteurs.

- On a développé également le prototype ML100 qui a été expérimenté afin de tester la propulsion par les LIMs qui signifie " Linear Induction Motors ".
- Ces deux expériences effectuées en lévitation furent d'un grand succès. C'est grâce à ces grandes avancées qu'est créé, en avril 1977, le premier centre de recherche sur les trains à lévitation magnétique à Miyazaki.
- A l'aube des années 80, le modèle ML-500 réussit à atteindre la vitesse vertigineuse de 517 km/h ! Les ingénieurs japonais décident alors de commencer à partir de 1981 des expérimentations avec wagons. Il sera d'abord exécuter des tests à deux wagons, puis à trois : avec trois wagons, le modèle atteint la vitesse de 352.4 km/h.
- Les années 90 seront essentiellement basées sur la sécurité, avec la construction du Centre Yamanashi qui a comme objectif principal l'élaboration d'un Maglev parfaitement sécurisé avant l'an 2000.
- En 1991, alors que les tests sur la lévitation latérale (par les murs du côté) commencent, un premier incident survient : un incendie éclate au niveau du pneu de secours du Maglev. Ce sinistre va beaucoup endommager le train et bloquer les expérimentations.
- Trois ans passent et un nouveau modèle, le MLU-002, entre en phase de test. Ce nouveau prototype a un caractère spécial, car il est le fruit du bilan de tout le travail déjà réalisé sur le Maglev. En effet, il réunit les principes testés dans le passé telle la propulsion LSM et LIM et également la lévitation latérale.

IV.3.3 Fonctionnement du train à Lévitation magnétique MAGLEV

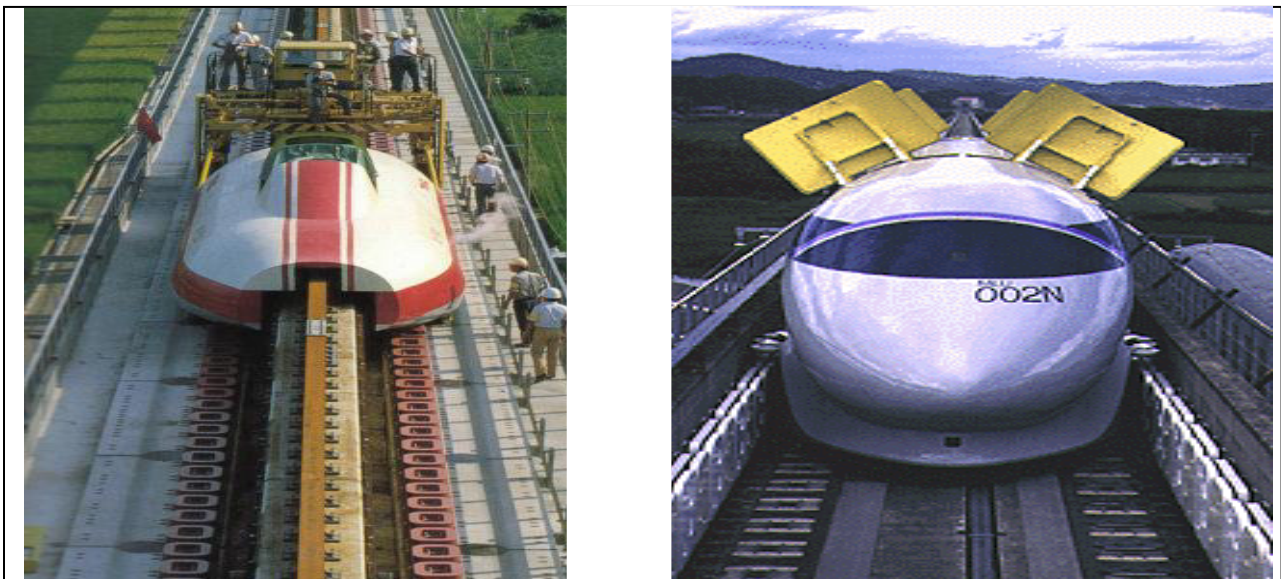


Figure IV.6 : Le système de sustentation magnétique et de freinage du train Maglev

Le train Maglev à lévitation magnétique est propulsé grâce à l'électromagnétisme. Les ingénieurs japonais dans la figure IV.6 utilisent la méthode de la sustentation magnétique. Le Maglev circule sur un rail en forme de U équipé de trois couches de bobines alimentées en électricité par des sous-stations fixées le long de la ligne.

Le système de freinage (Fig. IV.6) utilise de simples freins à disques. Une des solutions de rechange applicables est l'utilisation d'aérofreins, sorte de paravent pour train (sur le toit du train ci-dessus).

IV.3.4 Les parties principales

Les parties principales du système lévitation magnétique Maglev sont le véhicule lui-même, les rails, sont présente dans la figure IV.7 ci-dessous.

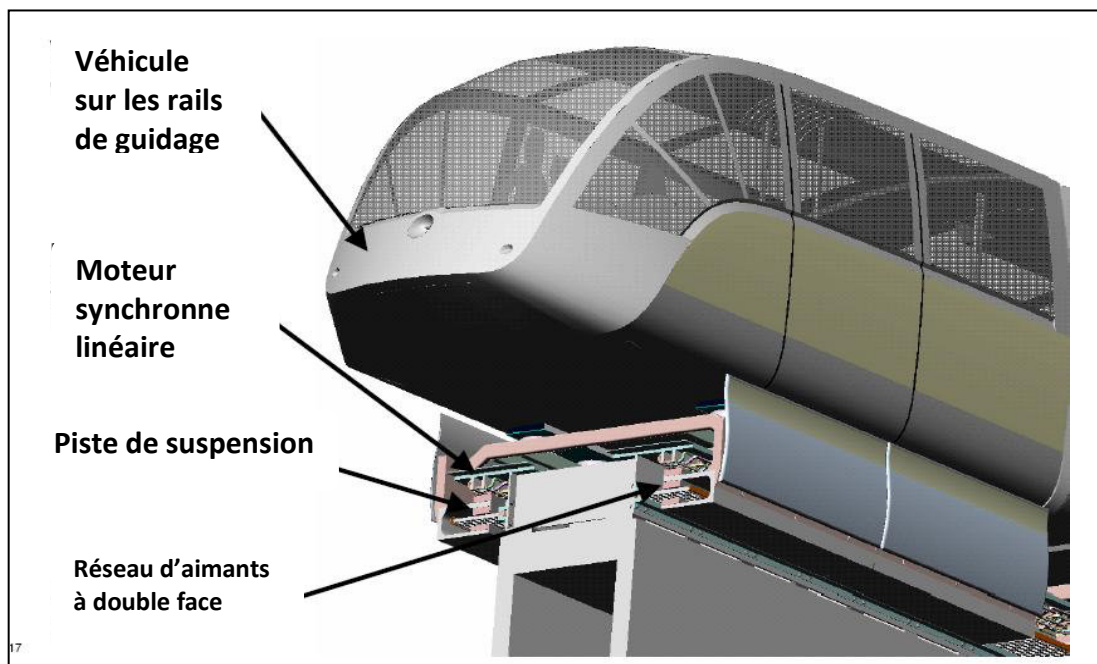


Figure IV.7 Parties Mécaniques essentielles du train Maglev

IV.3.5 Le système de suspension

Principalement trois types de technologies utilisées pour faire de la lévitation du wagon de la piste sont. [30]

IV.3.5 .a- Electromagnétique de suspension (EMS):

Dans ce train en lévitation au-dessus de la technologie d'un rail en acier à cause de la force d'attraction d'un électro-aimant monté dans la partie inférieure de la locomotive. Il garde toujours une distance de piste à environ 15 millimètres ou (0.6 in). L'inconvénient principal de cette technologie est-elle autre système requis pour le mouvement [33].

IV.3.5 .b- Electro-dynamique suspension (EDS):

Cette technologie est basée sur l'induction machines principe. C'est bobines électromagnétiques sont attachés à la locomotive et la voie. Puis un alternant courant traverse cette bobine le train se déplace en fonction de la fréquence du courant d'alimentation, dans cette lévitation technologie qui se passe à cause de la répulsion entre les pistes. Mais au repos de liaison de flux est faible si le courant induit dans la bobine au-dessous du train n'est pas assez grand pour faire léviter le poids du wagon, en raison de ce train il a besoin de quelques roues d'appoint ou de systèmes pour des vitesses lentes.

IV.3.5.c Stabilisé de suspension à aimant permanent:

Dans ce système de suspension adverse ensembles d'aimants de terres rares repousser les uns des autres et mettre le système dans des conditions variables.

IV.3.6 Système de contrôle

Dans toute technologie dont il a besoin pour maintenir une lame d'air entre la voie et le train. Tout système électromagnétique ne peut pas fournir un champ magnétique stable, il faut un système de stabilisation. Ces systèmes de détection en continu de l'entrefer et le cas échéant l'écart qui se produit. Ensuite le train Maglev se stabilise en modifiant le courant dans les bobines.

IV.3.7Système de rétroaction

Capteurs de déplacement sont utilisés pour la mesure de l'entrefer. Deux transducteurs sont situés pour détecter déformations longitudinales et transversales. En ce qui concerne le système de détection de tension de sortie de rétroaction (V_s) contrôle le courant magnétisant la bobine.

On utilise généralement un capteur infrarouge pour la détection de la position du train para port au rail.

IV.3.8 Système de propulsion

Dans le système EMS autre série d'électro-aimants sont utilisés pour la propulsion. Dans le cas des trains EDS il n'y a pas besoin d'un autre système de propulsion.

IV.3.9 Le système d'orientation

Systèmes flux nul est utilisé pour le guidage, il est une bobine qui est enroulée dans deux côtés de la voie. Lorsque le véhicule est dans la position en ligne droite, aucun courant ne circule dans la bobine, mais si elle se déplace tout le flux courbure dans une partie de la bobine est supérieure à l'autre côté, il va créer un flux changeant qui génère un champ qui ralentissent la vitesse de commutation d'un côté du système d'entraînement.

IV.4 Le système de contrôle du train en lévitation magnétique.

Avant d'aller au système complexe, on peut analyser un modèle simple de lévitation magnétique du train. Il s'agit d'une forme simple d'un système de train magnétique sans tenir compte de son frottement et les attributs complexes. Cette opération consiste à former marécageux, électro-aimant, l'aimant permanent, le circuit de commande, un circuit d'entraînement et des capteurs de déplacement. Liée à la sortie du régulateur de Foucault capteurs de courant essayer d'améliorer la force magnétique contre au problème de la pesanteur. Cela signifie que lorsque le courant à travers l'électro-aimant diminue le wagon est à la baisse à cause de la force gravitationnelle le capteur détecte que la distance et la sortie du contrôleur commence à augmenter, de cette manière le wagon garder à une distance permanente [37].

IV.4.1 Analyse du circuit électrique

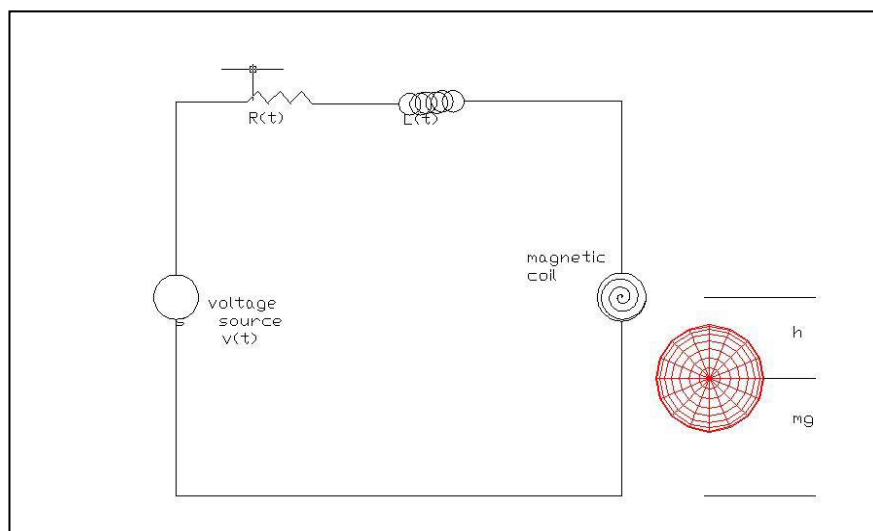


Figure IV.8. Schémas circuit électrique

En utilisant la première loi de Newton, la résultante des forces est :

$F = ma$, Où a est l'accélération.

Donc la force totale agissant = Masse du wagon * gravite - Force électromagnétique :

$$M \frac{dy^2}{dt^2} = mg - F \quad (4.6)$$

IV.4.2 l'analyse par fonction de transfert

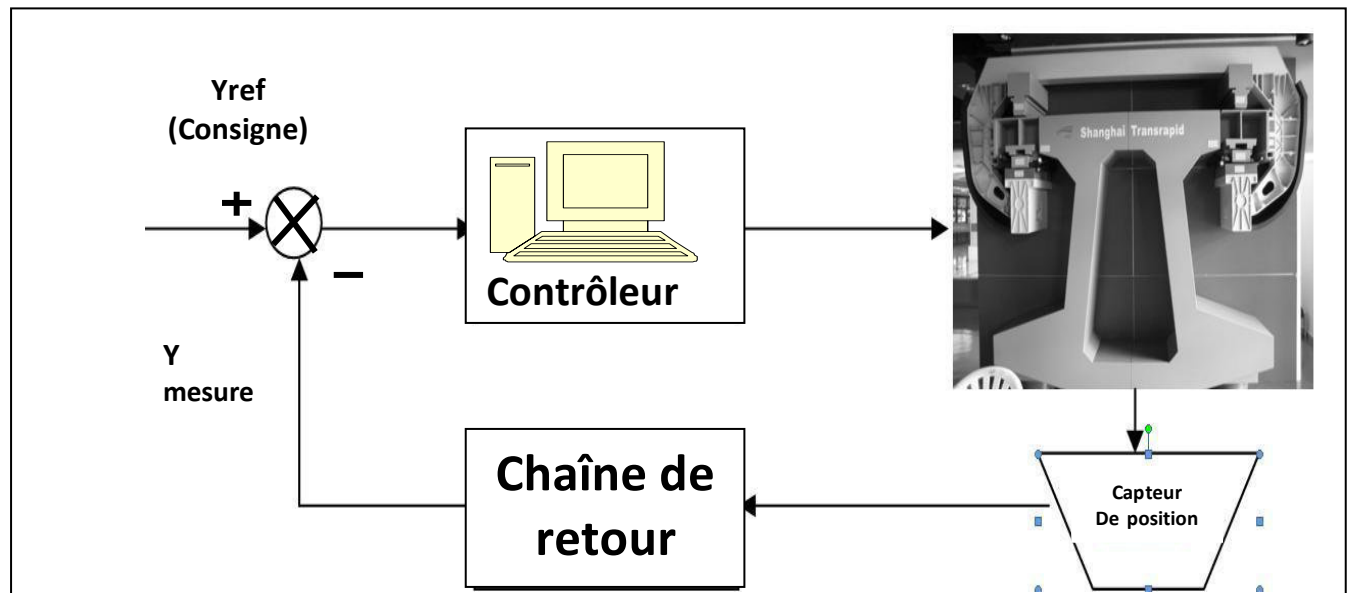


Figure IV.9 Schéma du système de commande en boucle fermée de Maglev

On peut analyser la relation force électromagnétique actuelle par la loi de Kirchhoff, comme dans le cas d'une simple bobine alimentée avec une source de tension $U(t)$.

i : courant circulant à travers la bobine,

R : résistance du circuit,

L : inductance du circuit magnétique (Bobine)

$$U(t) = Ri(t) + L \frac{di(t)}{dt} \quad (4.7)$$

$$\beta = \mu_0 \mu_r \frac{AN^2}{2} \quad (4.8)$$

$$i = \frac{U}{(Ls + R)} \quad (4.9)$$

μ_0 : La perméabilité de l'espace libre

μ_r : La perméabilité de référence à moyen.

N : Le nombre de spires de la bobine.

β : Constante liée à la force magnétique

Pour analyser la stabilité du système, on calcule en général les lieux des racines du système [17]. On peut placer un régulateur PID pour stabiliser la sortie qui est la position de l'objet en lévitation.

En

$$A = \begin{bmatrix} 0 & 1 & 0 \\ -cx_3^2/Mx_1^2 & 0 & -2cx_3/Mx_1 \\ 0 & 0 & -r/l \end{bmatrix} \quad (4.10)$$

$$x = \begin{bmatrix} y \\ y' \\ i \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 1/L \end{bmatrix}$$

$$C = [1 \quad 0 \quad 0]$$

En utilisant la formule bien connue

$$T(s) = C (sI - A)^{-1} \cdot B = \frac{y}{u} \quad (4.11)$$

$$(sI - A) = \begin{bmatrix} s & -1 & 0 \\ -cx_3^2/Mx_1^2 & s & -2cx_3/Mx_1 \\ 0 & 0 & s+r/l \end{bmatrix}$$

On obtient la fonction de transfert d'après les travaux de Huang [39]:

$$T(s) = \frac{\frac{\beta \cdot i_0^2}{x_0^3} / \frac{\beta \cdot i_0}{x_0^2}}{s^3 - \frac{R}{L} \cdot s^2 - \frac{\beta \cdot i_0}{M x_0^2} \cdot s + \frac{\beta \cdot i_0}{M x_0^2} / -\frac{R}{L}} \quad (4.12)$$

Les trains à lévitation magnétique utilisent des moteurs synchrones linéaires pour la propulsion de ses voitures. Ces moteurs génèrent des forces électrodynamiques dues à l'interaction des courants électriques et des forces magnétiques. Des ordinateurs à bord assurent le guidage (l'espacement vertical (entrefer ou 'airgap') et le déplacement horizontal) durant la traversée. Le train Maglev est relativement silencieux car il n'y a pas de contact avec le rail de guidage.

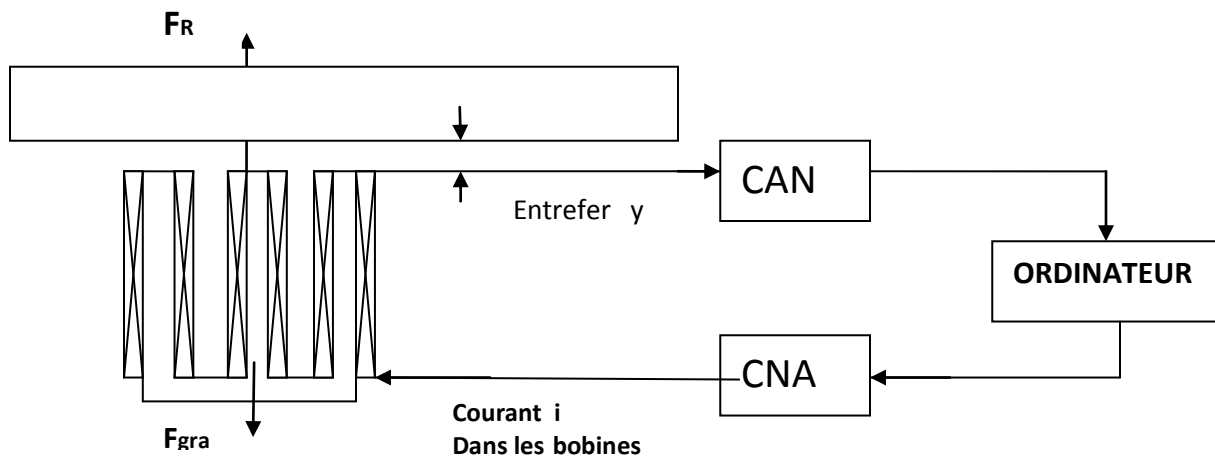


Figure IV.10 La commande pratique de la commande d'un train à lévitation magnétique

IV.4.3 Modélisation du système par la loi de physique

$$F_R = \frac{\mu_0 N^2 A i^2(t)}{2y^2(t)} = K_{mag} \frac{i^2(t)}{y^2(t)} \quad (4.13)$$

Où K_{mag} est la constante de magnétesation

$$\frac{d^2}{dt^2}y(t) = \frac{1}{m}(F_{grav} + F_{dist} - F_R) \quad (4.14)$$

$$M\ddot{x} = Mg - \frac{K_{ap}}{(x - x_{ap})^2} - \frac{K_{ea}I^2}{(x - x_{ea})^2} \quad (4.15)$$

K_{ap} , K_{ea} , x_{ap} et x_{ea} sont les paramètres de l'aimant et de l'électroaimant. M est la masse de la charge (ici la locomotive) et g est la constante de gravité.

L'équation dynamique du système est linéarisée afin d'étudier la stabilité du système à un point d'équilibre bien déterminé

Le régulateur PID classique a une fonction de transfert comme suit :

$$T(s) = K_p \cdot \left(1 + T_d s + \frac{1}{T_i s}\right) \quad (4.16)$$

Où : T_d est la constante de temps de dérivation

T_i Est la constante de temps d'intégration

La commande simplifiée par un régulateur classique PID du procédé de lévitation magnétique est développée en utilisant le logiciel Matlab simulink comme le montre la figure IV.11 ci-dessous :

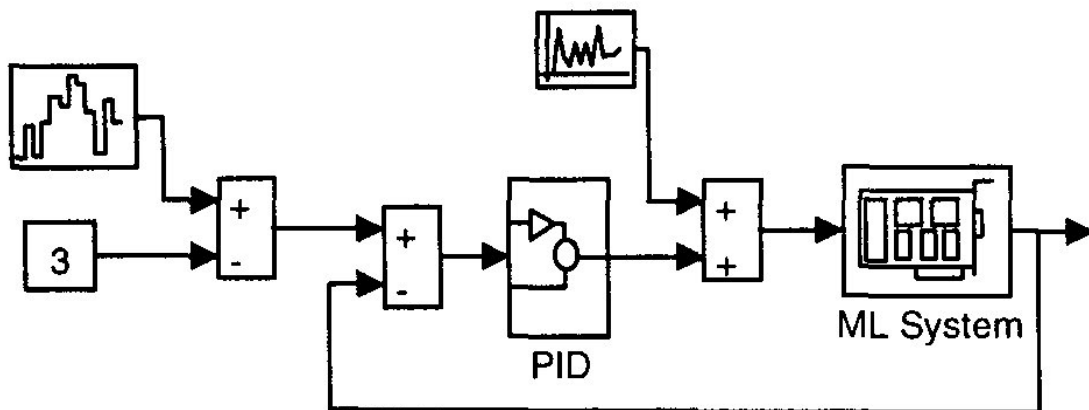


Figure IV.11 : Schéma de la Commande PID d'un Train de Locomotion à Lévitiation Magnétique Maglev

La figure IV.12 montre le modèle simplifié de la commande du train à lévitation magnétique sous Matlab-Simulink.

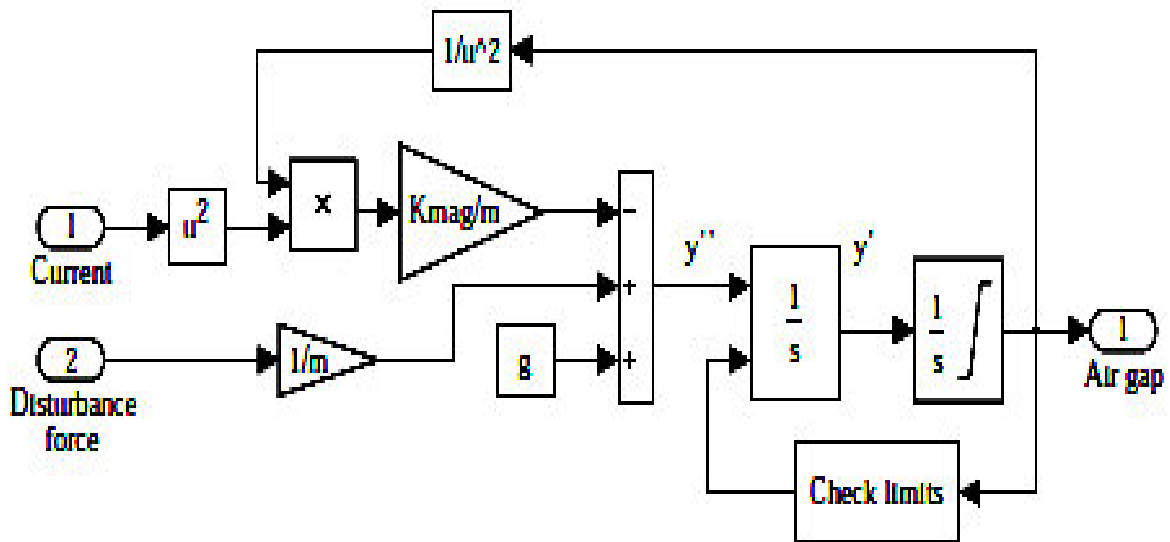


Figure IV.12: Le Modèle Simplifié du Train à Lévitation Magnétique sous Matlab Simulink

IV.5 Résultats de simulation sous Matlab-Simulink de la commande de Maglev

IV.5.1 Données du Système Experimental

Les résultats de simulation sont exposés effectuée sous l'environnement Matlab/Simulink, afin de tester les performances d'une commande non-linéaire d'un système industriel et pédagogique fort intéressant à savoir système de lévitation magnétique.

On utilise pour l'analyse et les simulations à travers Matlab Simulink les données suivantes [36] [37]:

Masse de la locomotive (M) = 5000kg

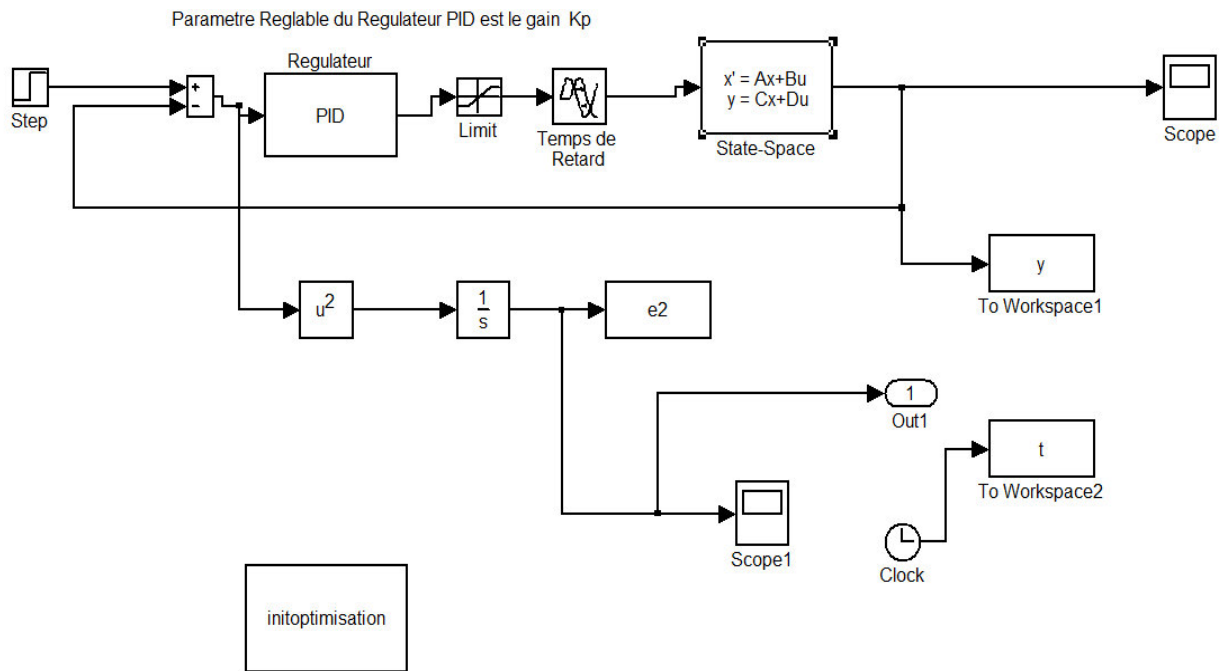
Force de Gravitation (g) = 9.81m/s²

Inductance de la bobine L(H) = 280nH (à 10khz)

Résistance de la Bobine (R)= 300k ohm

Courant consommé dans la bobine(i) = 42A

La distance d'entrefer (entre la locomotive et le rail (x) = 0,015m



Cliquer ici pour Initialiser les parametres du procede et les parametres a optimiser

Figure IV.13 schéma bloc de simulation de la commande PID-PSO du train Maglev

$$J = \int_0^{\infty} e^2 dt$$

Indice de performance du système qui est l'intégrale du carré de l'erreur

$$\frac{dJ}{dK_p} = 0$$

J Tend vers zéro ou une constante lorsque le système est bien optimise

IV.5.2 Modèle simplifié de la commande sous Matlab/Simulink

Le schéma simplifié de la commande optimale de Maglev en utilisant un réseau de neurones artificiels de type MLP sous Matlab/Simulink est montré dans la figure IV.14.

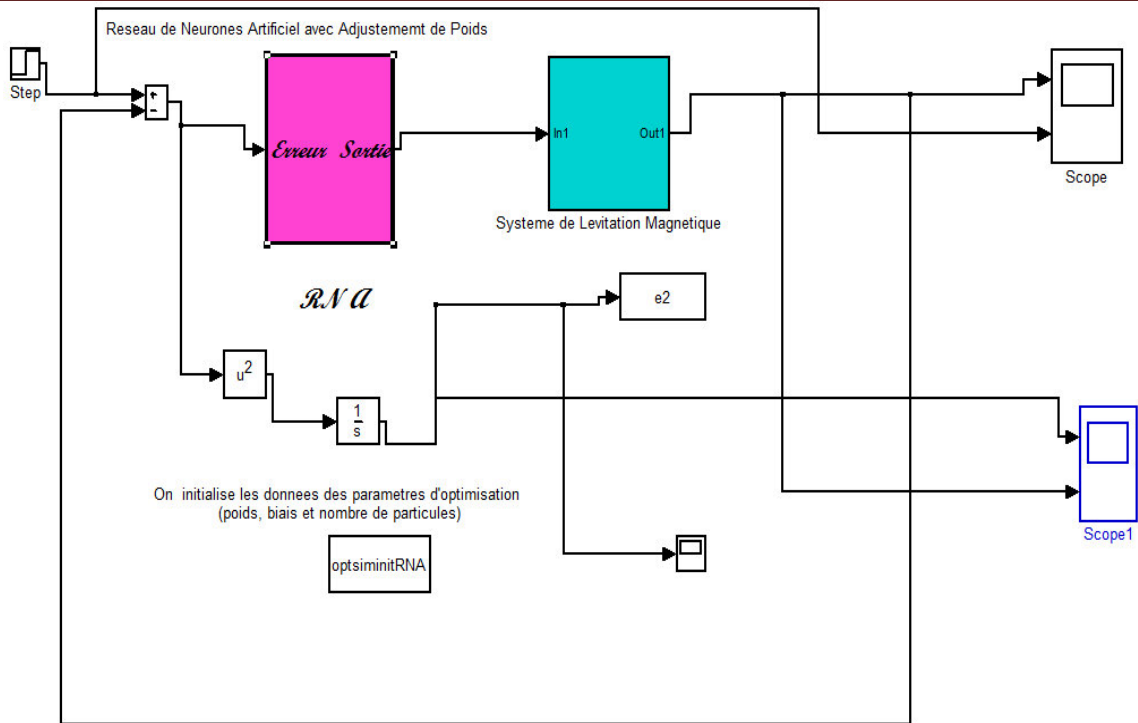


Figure IV.14: Le Modèle Simplifié de la Commande de MAGLEV par un RNA sous Simulink

Le schéma qui utilise le modèle détaillé [37] du système pour une autre possibilité de simulation est montré dans la figure IV.15 ci-dessous :

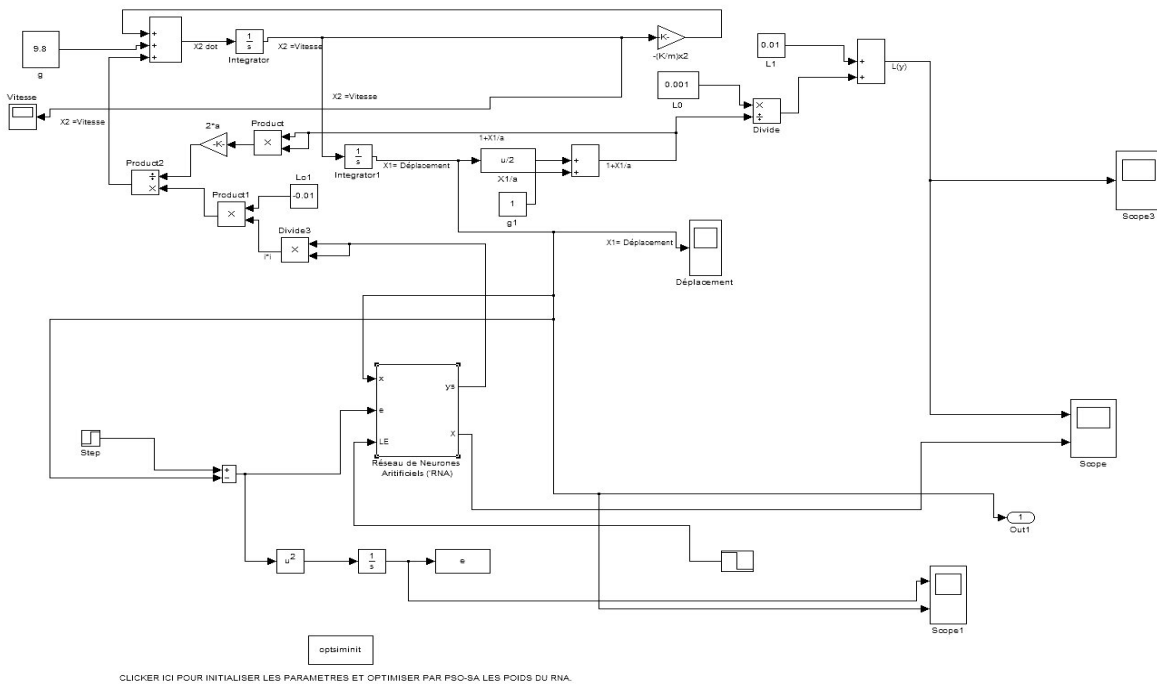


Figure IV.15: Le Modèle détaillé la Commande par un RNA de Maglev sous Matlab Simulink

Le réseau de neurones qu'on utilise afin de l'optimiser par l'heuristique PSO représenté par la figure IV.16

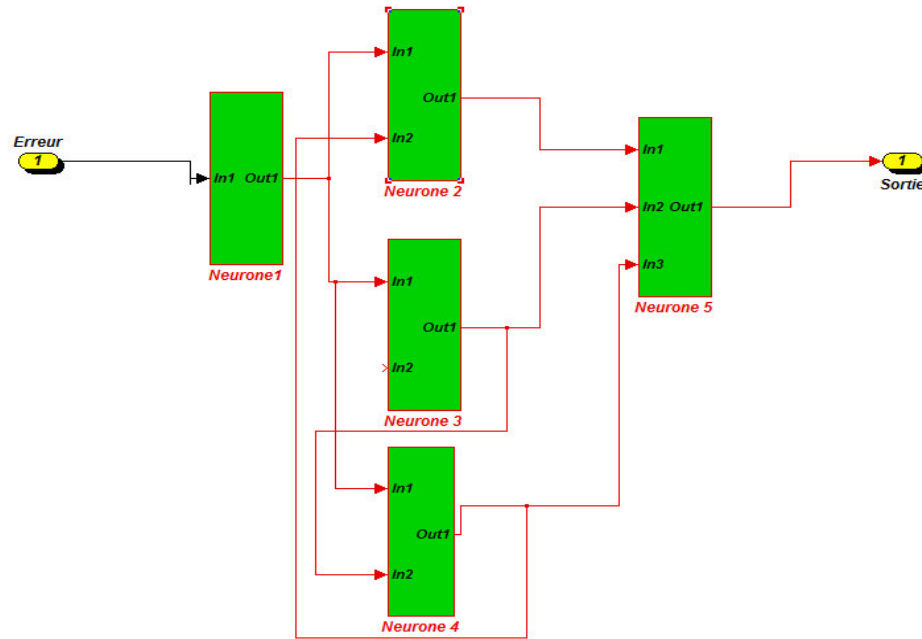


Figure IV.16: Le Modèle du contrôleur à Réseau de Neurones Artificiels utilisé sous Matlab Simulink

Afin d'obtenir les poids et biais on a fait l'apprentissage des cellules neuronales off-line suivant une trajectoire de référence y désirée afin de fixer les poids des neurones 1, 2, 3 et 4. Il reste le neurone de la couche de sortie du contrôleur qu'on fixe d'abord pour la visualisation et qu'on cherchera à optimiser par la suite.

IV.6 Résultats de simulation et interprétation

Le résultat obtenu par simulation en utilisant le logiciel disponible Matlab/Simulink de la figure IV.17 concerne la du procédé de lévitation magnétique qu'est le train Maglev par un régulateur classique PID.

La réponse (sortie) du système est la distance d'entrefer y entre le train (locomotive) et les rails. Elle est obtenue pour toutes les simulations qui suivent lorsque la consigne y_c (l'entrée) est un signal échelon d'amplitude $y_c = 0,015\text{m}$.

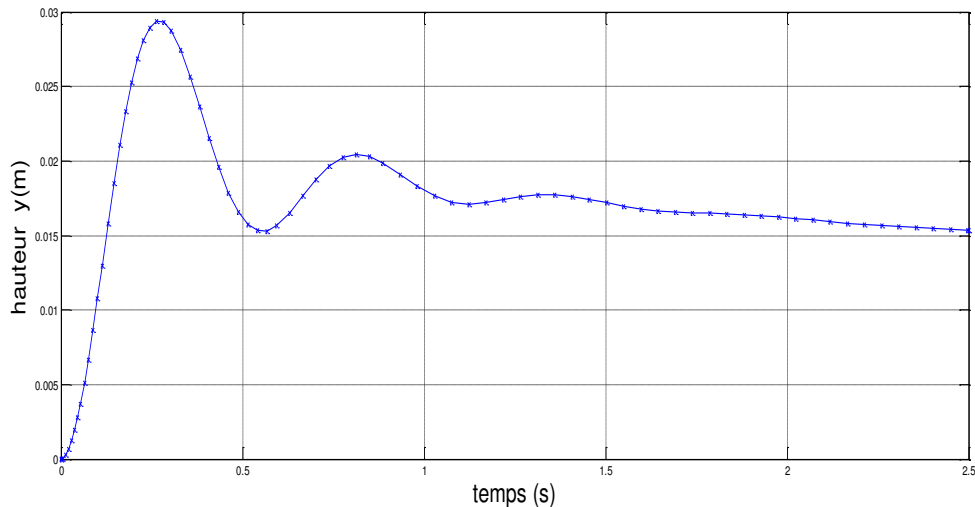


Figure IV.17 Variation de l'entrefer y en utilisant un régulateur PID classique lors du démarrage.

La valeur consigne de y est choisie pour avoir l'entrefer égale à 0,015m.

Le régulateur PID classique est utilisé dans un premier test avec les gains $K_p=2,8$, $K_d=0$ et $T_i=10s$

Dans la figure IV.17 la constante K_p est choisie on remarque que lors du démarrage du Maglev l'entrefer y présente des oscillations importantes qui peuvent s'avérer gênantes.

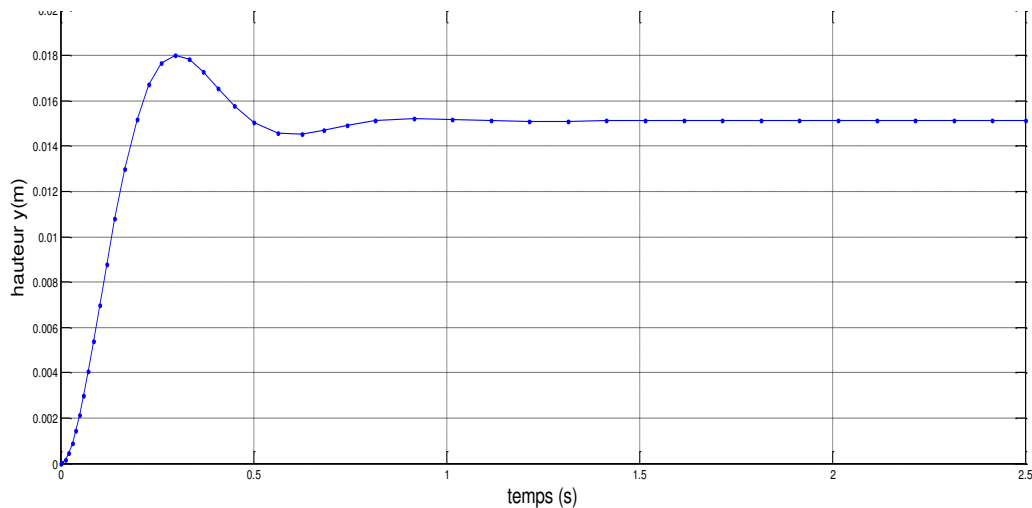


Figure IV.18 Variation de la distance d'entrefer y entre le wagon et le rail lors du démarrage
On variant les paramètres du régulateur PID

Dans la figure IV.18 on a fixé les paramètres du régulateur PID, tels que $K_p=2,8$, $T_d=0,8s$ et $T_i=10s$

Le premier dépassement a baissé notablement et les premières oscillations ont été atténuées considérablement amis le temps de réponse est de l'ordre de 0,7s.

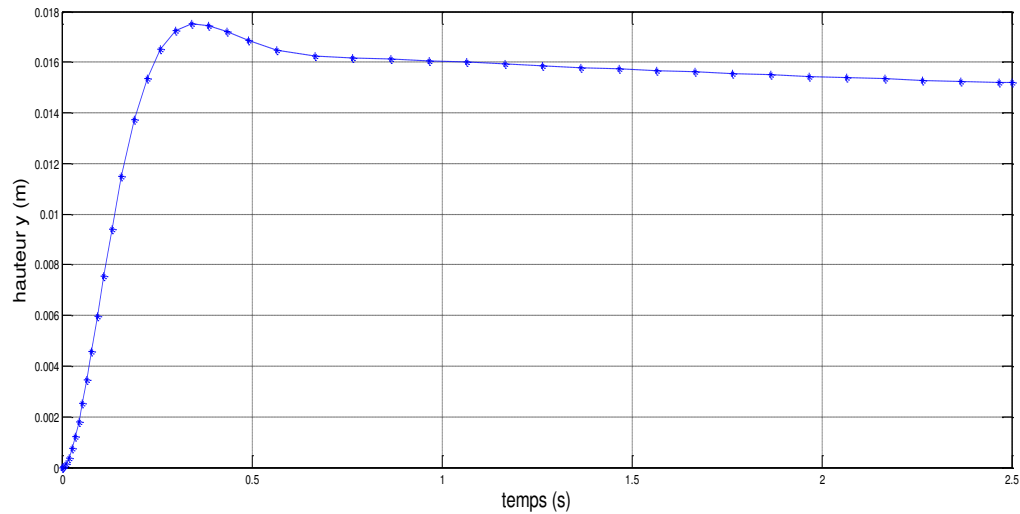


Figure IV.19 : Réponse du système en utilisant la commande par la métaheuristique PID-PSO

Dans la figure **IV.19** on fixe les paramètres le régulateur PID est optimisé par la PSO concernant le gain $K_p=2,2$, $T_d=0,8s$ et $K_i=10s$. Cette valeur donne une meilleure réponse concernant la variation de l'entrefer lors du démarrage de Maglev. Les oscillations sont fortement atténuées et seul le premier dépassement subsiste dans la réponse qui a une valeur de $0,0175m$ $t=0,3s$. On remarque que le temps de réponse t_r a diminué par rapport au régulateur PID non optimisé par la PSO et il est de l'ordre de $0,55s$.

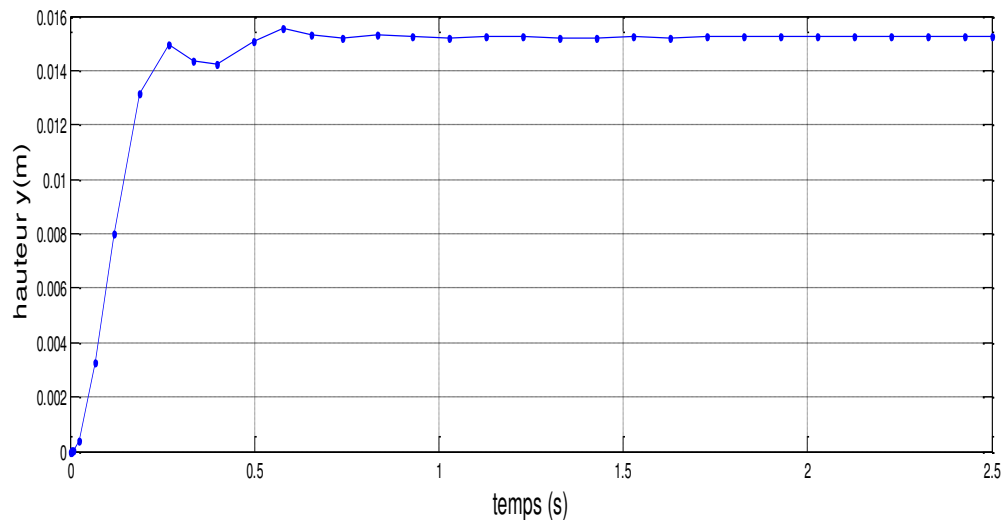


Figure IV.20 Variation de la distance d'entrefer $y(t)$ en utilisant un régulateur à RNA uniquement.

Dans un premier test, on cherche à réduire le premier dépassement de telle manière à éviter un fort amortissement pour $t=0,3s$.

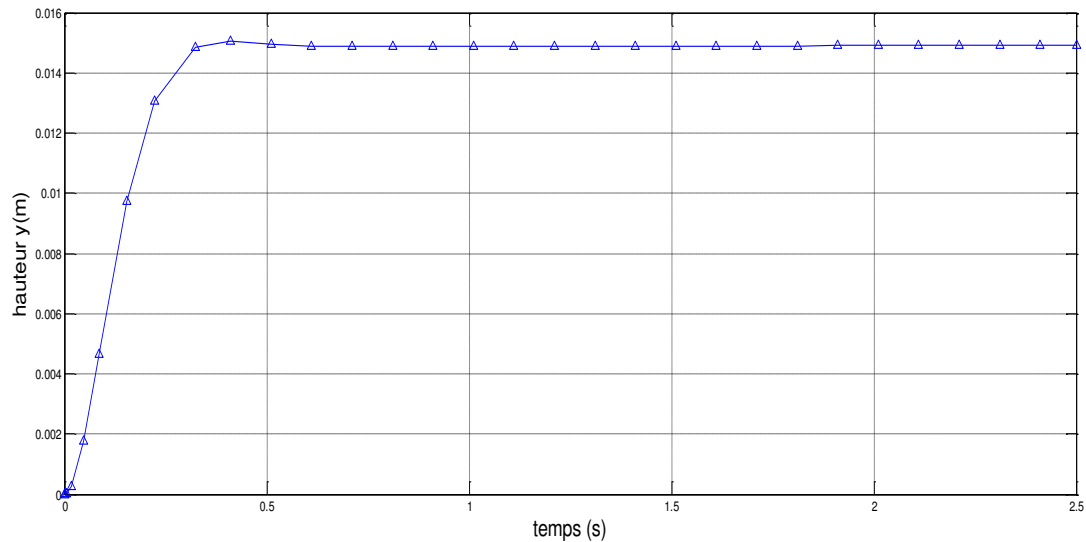


Figure IV.21 : Réponse du système à un échelon en utilisant la métaheuristique PSO-RNA

Dans le cas de la figure ci-dessus (IV.21) on a amélioré le temps de réponse (en diminuant le temps de montée) avec la technique PSO-RNA qui réduit le régime transitoire. En choisissant des poids optimaux estimés par la PSO $W_{41} = 0,8$ et $W_{42} = 0,85$ pour les cellules de la couche de sortie, on a obtenu un temps de réponse de manière optimale $tr \approx 0,3$ s.

La figure IV.22 illustre le processus méthode avec lequel la PSO optimise un paramètre tel que la constante K_p du régulateur PID. A l'état initial on a choisi 15 particules éparpillées autour du maximum. Après 20 itérations les particules convergent vers le maximum global montré dans la figure IV.23. La fonction fitness [22] est choisie telle qu'elle assure une convergence vers le maximum en un minimum d'itérations [15].

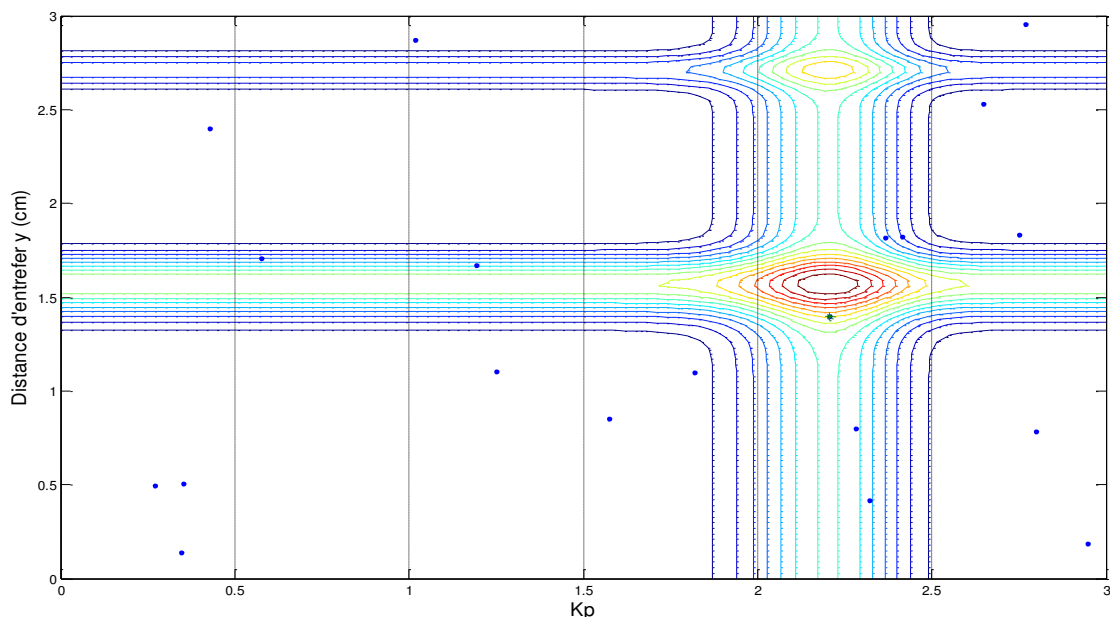


Figure IV-22 : Etat initial des particules pour l'optimisation de K_p du régulateur PID par la PSO.

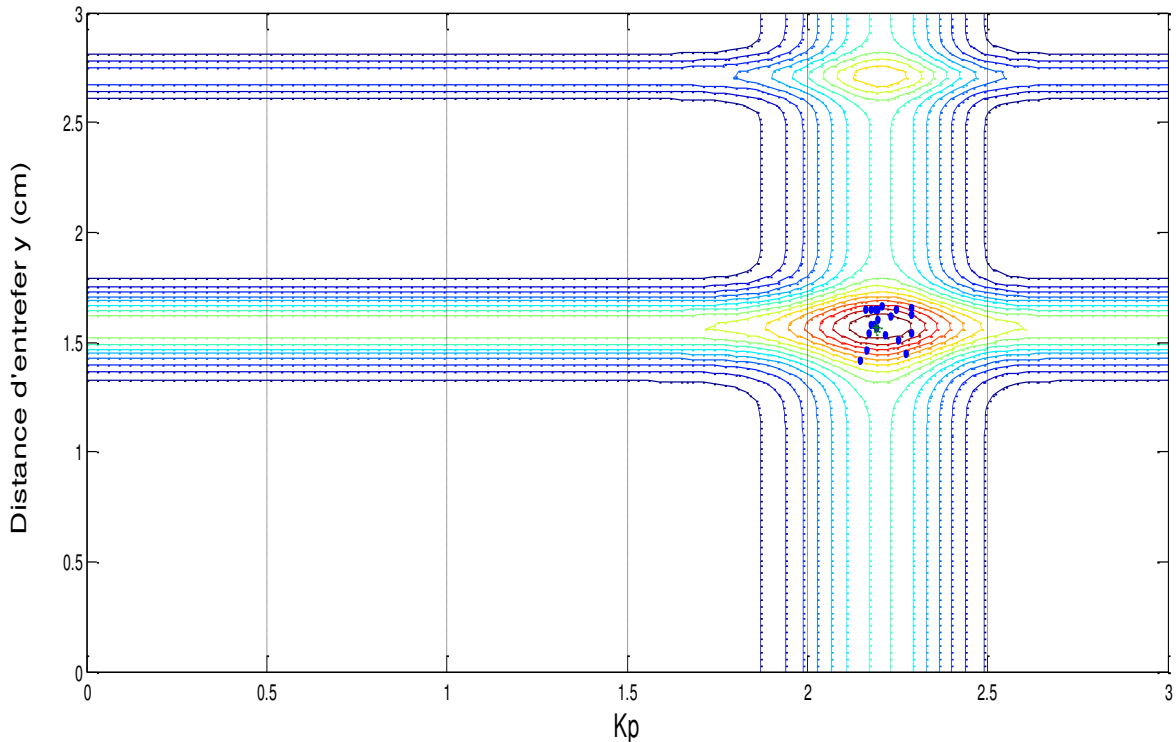


Figure IV-23 : Convergence des particules pour l'optimisation de K_p par la PSO après 20 itérations.

D'après les résultats de simulation obtenus des figures IV.17-18-19-20-21, on peut déduire que la stratégie de commande du procédé fortement non-linéaire étudié qu'est le train à lévitation magnétique Maglev par deux métaheuristiques différentes en utilisant la PSO conjointement avec le régulateur classique PID et avec le contrôleur intelligent à RNA. En comparant les résultats obtenus avec la métaheuristique PSO-PID de la figure IV.19 et ceux obtenus par celle PSO-RNA de la figure IV.21, cette dernière donne une meilleure réponse concernant le temps de réponse réduit et l'élimination totale des oscillations qu'on trouve même avec la métaheuristique classique PSO-PID. Ainsi qu'elle augmente la marge de stabilité. La stratégie proposée de la métaheuristique peut être élaborée et appliquée en temps réel dans la commande des procédés non-linéaires. Elle permet une optimisation des paramètres du contrôleur.

Vu la complexité du modèle réel du train Maglev et les contraintes de fonctionnement, les métaheuristiques employées dans notre étude peuvent être hybrides. Nous avons soulevés un état de l'art de la commande intelligente qui utilise les algorithmes évolutionnaires. Tous les aspects du problème de la commande des procédés non linéaires complexes passent par l'analyse des grandeurs à commander, les paramètres variables du procédé, les paramètres du contrôleur à optimiser. Enfin on

opte pour les métaheuristiques qu'on utilise pour résoudre le problème de la commande. Le recuit simulé n'a pas été élaboré dans notre étude présente.

IV-7 Conclusion

Dans ce chapitre nous avons défini la problématique la commande d'un système industriel non linéaire réel dont l'usage dans le monde est grandissant, à savoir le train Maglev. Les métaheuristiques PID-PSO et PSO-RNA ont été implémentées avec succès pour l'optimisation des paramètres du contrôleur PID et neuronale. Et on a montré que ces techniques apportent une amélioration notable et satisfaisante à la réponse du procédé qui est ici la distance d'entrefer. La technique du recuit simulé cependant n'a pas été abordée et est suggérée pour des travaux de recherche future dans le domaine.

CONCLUSION GENERALE

Dans le présent mémoire, nous avons défini la problématique de la commande d'un système industriel non linéaire réel, le train Maglev, dont l'usage dans le monde est de plus en plus grandissant. Les métaheuristiques PSO-PID et PSO-RNA ont été implémentées avec succès pour l'optimisation des paramètres du contrôleur PID et neuronale. On a montré que ces techniques apportent une amélioration notable et satisfaisante au contrôle de la réponse cruciale du procédé : la distance d'entrefer. La technique du recuit simulé cependant n'a pas été abordée et est suggérée pour des travaux de recherche future dans le domaine.

Vu la complexité du modèle réel du train Maglev et les contraintes de fonctionnement, les métaheuristiques employées dans notre étude peuvent être hybrides. Nous avons soulevés un état de l'art de la commande intelligente qui utilise les algorithmes évolutionnaires. Tous les aspects du problème de la commande des procédés non linéaires complexes passent par l'analyse des grandeurs à commander, les paramètres variables du procédé, les paramètres du contrôleur à optimiser. Enfin on a opté pour les métaheuristiques qu'on a pu utiliser pour résoudre le problème de la commande optimale de la distance d'entrefer.

Les résultats de simulation obtenus sous l'environnement Matlab/Simulink sont satisfaisants de la commande d'un train à lévitation magnétique. Ils montrent que la PSO est un outil d'optimisation performant qui permet de réaliser une exploration globale de l'espace pour la recherche du maximum global, contrairement aux méthodes déterministes classiques.

Pour compléter le présent travail de recherche, une identification plus poussée du modèle du processus à commander Maglev pourrait être faite. On suggère aussi, d'utiliser des outils visuels disponibles pour visualiser les actions de celui-ci, tels que le 'Virtual Reality Toolbox' de Matlab/Simulink. Des commandes à distance pourraient aussi être incluses dans la simulation en utilisant ce logiciel.

BIBLIOGRAPHIE

- [1] **CLERC et SIARRY**, « *Une nouvelle métaheuristique pour l'optimisation difficile : la méthode des essaims particulaires* » Vol. 3-7 France Télécom R&D; Université Paris 12 2004.
- [2] **Lapetoule kévine**, « *Les algorithmes métaheuristiques*», édition Springer 01 juin 2006.
- [3] **Rachid chelouah**, «*Chapitre I : Généralités sur les méthodes d'optimisation*», université de Cergy, France , 15 Décembre 1999.
- [4] **J.F. Jodouin**, « *Les Réseaux de Neurones. Principes et Définitions* », édition Hermès, Paris, 1994.
- [5] **Maurice Clerc**, « *L'optimisation par essaim particulaire* » Tutoriel pour OEP 2003 (partie I), Carré des Sciences, Paris,France, Octobre 2003.
- [6] **F. Tangour et P. Borne**, « *Presentation of Some Metaheuristics for the Optimization of Complex Systems* », Studies in Informatics and Control, Volume 17, No 2, juin 2008.
- [7] **F.T.S Chan et M. K. Tiwari**, « *Swarm Intelligence : Focus on Ant and Particule Swarm Optimization* », I-TECH Education and Publishing, Croatia 2007.
- [8] **R. V. Dukkipati**, « *Analysis and Design of Control Systems Using Matlab* », New Age International (Ltd) Publishers, New-Delhi, 2006.
- [9] **R. Babuska and S. Stramigioli**, « *Matlab Simulink for Modeling and Control* », Delft University of Technology, Hollande, Novembre 1999.
- [10] **M. Lairi et G. Bosh**, « *A Neural Network With Minimum Structure for Maglev System Modeling and Control* », Proceedings of IEEE , International Symposium on Intelligent Control/Intelligent Systems and Semiotics, Septembre 1999.
- [11] **K. S. Lee and Z. W. Geem ZW**, « *A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice* », Computer Methods Applied Mechanical Engineering, 2005.
- [12] **T. Benmiloud**, «*Techniques de Modélisation et de Commande Neuro-floue Adaptative*» (ANFIS), Mémoire de Magister, Université des Sciences et de la Technologie Mohamed Boudiaf d'Oran, USTO, 2006.

[13] **S. Varkiani**, « *Application des Réseaux Neuronaux Flous à l'Identification et la Protection d'un Transformateur Triphasé* », Mémoire présenté à l'université du Québec à Chicoutimi comme exigence partielle de la maîtrise en ingénierie, Avril 1998.

[14] **I. Kebati**, « *Identification et Commande adaptative par réseaux de neurones d'un moteur asynchrone* », Mémoire de Magister, Ecole Normale Supérieure de Technologie d'Oran, E.N.S.E.T-Oran, juillet 2007.

[15] **M. Zerikat, M. Guedda**, « *Neuro-predictive speed control for induction motor drives using Dynamic Neural networks* », Revue Communication Science et Technologie de l'E.N.S.E.T d'Oran, C.O.ST, Numéro 02, pp 69-74, 2003.

[16] **M. Zerikat and S. Chekroun** « *Adaptation Learning Speed Control for a High performance Induction Motor using Neural Networks* » Proceedings of World Academy of Science, Engineering and Technology, Volume 35 November 2008, ISSN 1307-6884, pp 294-299.

[17] **T. Pajchrowski and K. Zawirski**, « *Robust Speed Control of Servodrive Based on ANN* », IEEE ISIE 2005, International Symposium on Industrial Electronics, June 20-23, 2005, Dubrovnik, Croatia, pp. 81-86.

[18] **J. Soltani and R. Yazdanpanah** « *Robust Backstepping Control of Induction Motor Drives Using Artificial Neural Networks* », CES/IEEE 5th International Power Electronics and Control Conference, August 14-16, 2006 Shanghai, China.

[19] **LAYEB Abdesslem**, « *Les_methodes_de_recherche_locale* »: University Mentouri of Constantine, Algeria, Computer science Department. 2009.

[20] **O. GUENOUNOU**, « *Méthodologie de conception de contrôleurs intelligents par l'approche génétique- application à un bioprocédé* », Mémoire présenté à l'université du Toulouse III sep 2009

[21] **M.Mordjaoui**, « *Modélisation des effets électromagnétiques Apport de la logique floue et Neuro-floue* » Thèse de doctorat es-sciences, Université De Batna, Juin 2008.

[22] : **BENSTRIRA K.**, et **Bouarouj** « *recherche tabou, rapport de recherche, module recherche opération avancée* », Master 2, 2010.

[23] **Robert Babuska and stefano Stramigioli** « *Matlab and Simulink for Modeling and Control* » Delft University of technology November 1999.

[24] **Mostafa Lairi and Gérard Bloch**, « *A neural network with minimal structure for MagLev system modeling and control* » IEEE September 15-17,1999.

- [25] **M. Lairi**, « Identification et commande neuronales des systèmes non linéaire : application à un système de sustentation magnétique », Thèse de Doctorat de l'université de Henri Poincaré-Nancy spécialité Automatique 1998.
- [26] **W. J. Kim, D. L. Trumper, and J. H. Lang**, “Modeling and vector control of planar magnetic levitator,” IEEE Trans. Ind. Appl., vol. 34, no. 6, pp. 1254–1262, Nov. 1998.
- [27] **K.H. Lundberg, K.A Lifienkamp and G.Marsden**, « Low-Cost Magnetic Levitation Project Kits » IEEE Control Systems Magazine, Octobre 2004,pp 65-69.
- [28] **Xiao-Jing, Zheng, Xiao-Fan Gou**, « Influence of flux Creep on dynamic Behavior of magnetic Levitation Systems with a high-*t_c* Superconductor », IEEE Trans. Appl. Supercond. Vol 15 No 3, p. 3856-3863, september 2005.
- [29] **E.P Furlani**,« A Formula for the Levitation force between Magnetic Disks », IEEE Trans. Magn. Vol 29, p. 4165-4169, November 1993.
- [30] **Lahssane Idoumghar , M. Melkmi**, « Hybrid PSO-SA Type Algorithms for multimodal function », Universite Henri Poincard Vandoeuvre France Vol 2011, Article 138078, Apr 2011.
- [31] **Tooraj Hakim, Abdolamir Nekoubin** « Optimal control design for linear Magnetic Levitation Rail System », World Academy of science engineering and Technology 2011
- [32] **Hamid Yaghoubi**, « Practical Application of Magnetic Lévitation Technology», Universite Henri Poincard Vandoeuvre France, sep 2011.
- [33] **P. Suster and A. Jadlovsk** « Modeling and Control Design of Magnetic Levitation System », Technical University of Kosice/Department of Cybernetics and Artificial Intelligence, Košice, Slovakia2007
- [34] **Ishtiaq Ahmad, Muhammad Akram Javaid**, « Nonlinear Model & Controller Design for Magnetic Levitation System », University of Engineering & Technology, Taxila, Pakistan, 2009
- [35] **Yousfi Khemissi** « Control Using Sliding Mode Of The Magnetic Suspension System», International Journal of Electrical & Computer Sciences IJECS-IJENS Vol: 10 No: 03
- [36] **Yang. Huang**«Model Predictive Control of Magnetic Bearing System» thesis Master of Engineering School of Electrical and Computer Engineering Royal Melbourne Institute of technology December 2007
- [37] **Shibo Ren, Arie Romeijn, Kees Klap** « Dynamic Simulation of the Maglev Vehicle Guide way System », Journal Of Bridge Engineering Asce / May/June 2010
- [38] **Kang Seok Lee, Zong Woo Geem** « A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice »,science direct Comput. Methods Appl. Mech. Enirg. 194 (2005) 3902–3933

Annexe I

Organigramme et algorithme de la PSO –SA

I.1 Algorithme d'optimisation par essaim particulaire

La position au temps t de la particule i est alors définie par (1.3.4).

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), j \in \{1, 2, \dots, d\}$$

Les différentes étapes de l'OEP sont présentées dans l'algorithme 1.2.

OEP	
1	Initialiser aléatoirement les positions et vitesses de chaque particule
2	Evaluer les positions des particules
3	Pour chaque particule i , $\vec{p}_i = \vec{x}_i$
4	Calculer les \vec{g}_i
5	tant que le critère d'arrêt n'est pas satisfait faire
6	Déplacer les particules selon (1.3.2) et (1.3.4)
7	Evaluer les positions des particules
8	Mettre à jour \vec{p}_i et \vec{g}_i
9	fin

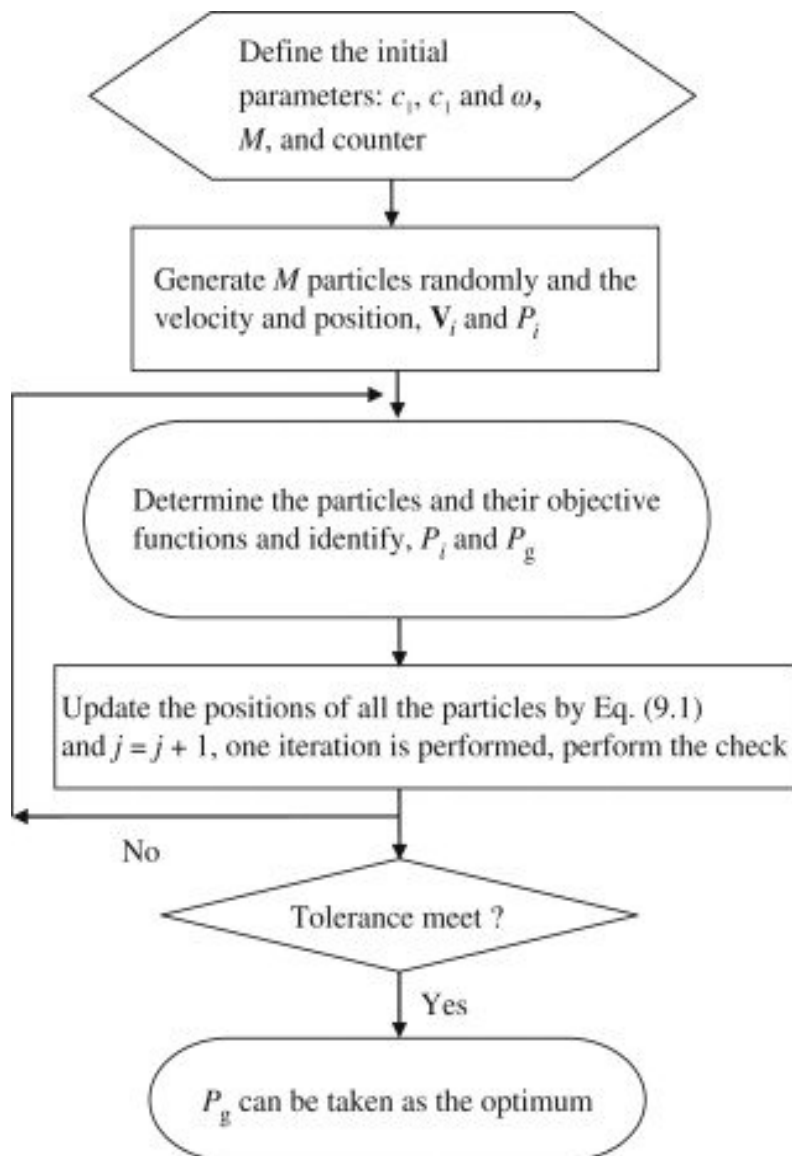
Algorithme d'optimisation par essaim particulaire.

I.2 Organigramme de La technique Hybride PSO avec la SA

```
Initialize swarm with random positions and velocities;
begin
initialize PSO and SA;
while (termination !=true)
do{
generate swarm;
compute and find best Pgd;
}
set particle that gives best Pgd as initial solution to SA;
while (Tcurrent>Temp_end)
do{
generate neighborhood;
evaluate and update best solution and temperature;
}
end.
```

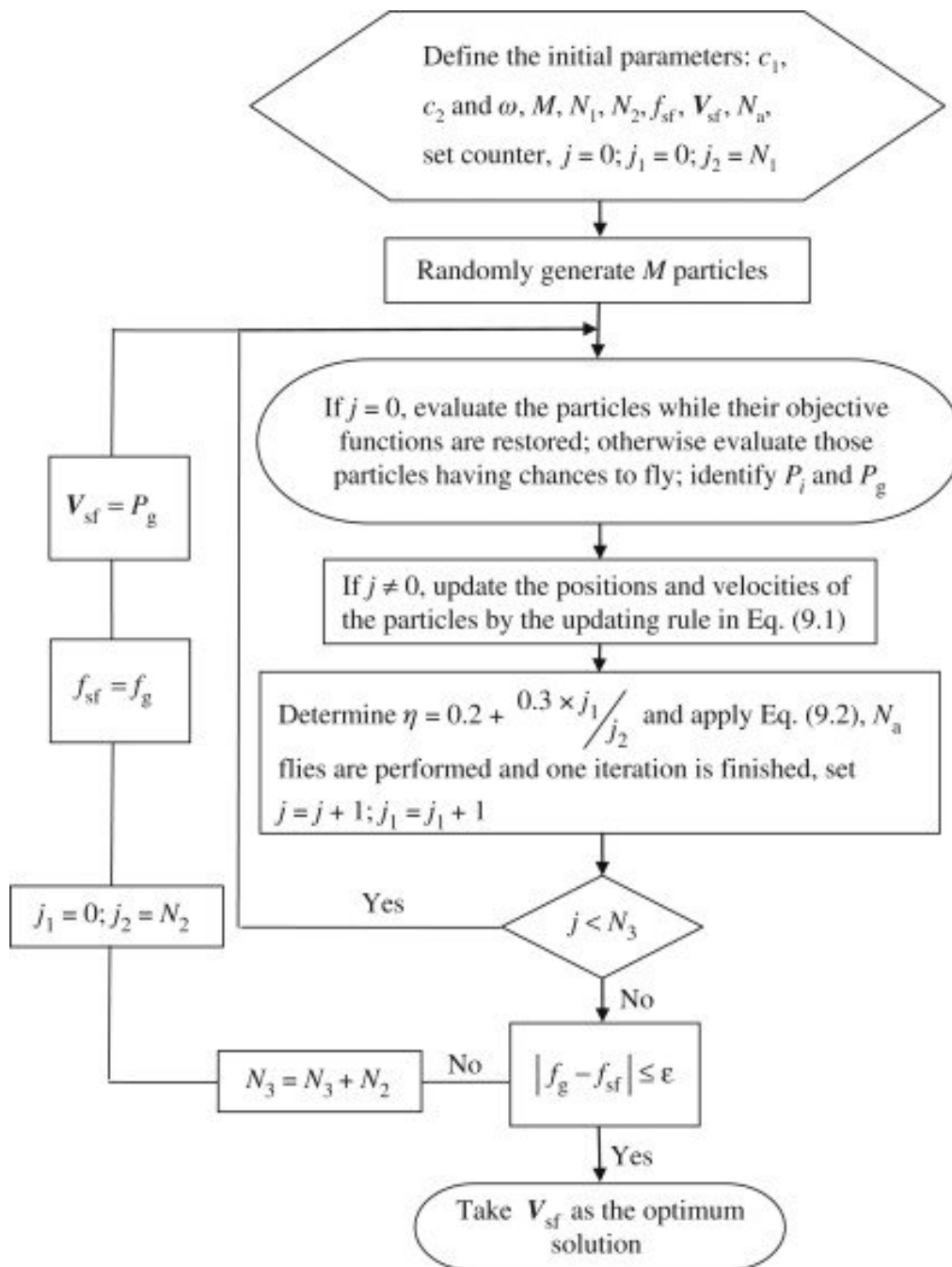
Organigramme de La technique Hybride PSO avec la SA

I.3 Organigramme pour PSO



Organigramme pour PSO (M est communément considéré comme le double du nombre de variables de contrôle et est liée au nombre de solutions potentielles; $N1$ désigne le nombre d'essais de boucle interne et $N3$ est un paramètre pour contrôler la cessation des essais).

I.4 Organigramme pour MPSO



Organigramme pour MPSO.

Annexe II

Structure détaillée du système de commande du train

Maglev

