

Université Mohamed Khider – Biskra
Faculté des Sciences et de la technologie
Département : Génie Electrique



جامعة محمد خيضر بسكرة
كلية العلوم و التكنولوجيا
قسم : الهندسة الكهربائية

Thèse présentée en vue de l'obtention
du diplôme de
Doctorat en science en Génie électrique.

Option : Electrotechnique

**Identification des paramètres des systèmes
non linéaires basée sur les techniques Soft
Computing**

Présentée par :

Megherbi Ahmed Chaouki

Soutenue publiquement le 04 juillet 2012

Devant le jury composé de :

Dr. Betka Achour	Professeur	Président	Université de Biskra
Dr. Benmahammed Khier	Professeur	Rapporteur	Université de Setif
Dr. Taleb Ahmed abdelmalik	Professeur	Examineur	Université de Valenciennes
Dr. Benchouia Mohamed Taoufik	Maitre de Conférences 'A'	Examineur	Université de Biskra
Dr. Aissaoui Abdelghani	Maitre de Conférences 'A'	Examineur	Université de Bechar
Dr. Tahour Ahmed	Maitre de Conférences 'A'	Examineur	Université de Mascara

Résumé

Les travaux présentés dans cette thèse traitent les apports des techniques Soft Computing au domaine de l'identification des paramètres d'un système non linéaire.

Notre étude se focalise en premier lieu, sur l'introduction de plusieurs notions sur les systèmes non linéaires et les techniques d'identification paramétriques conventionnelles et leurs limitations. En second lieu, notre intérêt est porté sur la présentation de deux techniques soft computing à savoir l'algorithme génétique et l'essaim de particule.

Dans cette étude, en vue de l'amélioration de la convergence et la précision des solutions optimales nous présentons une approche d'identification basée sur les algorithmes génétiques qui intègre une stratégie de variation de poids de pondération de la fonction du coût et un autre algorithme génétique où nous proposons un codage réel modifié de la variable. Une autre approche d'optimisation basée sur un algorithme d'essaim de particules modifié a été proposée.

Nous appliquons enfin ces techniques proposées pour le problème d'identifications des paramètres d'un moteur asynchrone pris comme exemple typique d'un système non linéaire.

Mots-clés : Identification des paramètres, systèmes non linéaires, algorithmes génétique, essaim de particule, moteur asynchrone

Abstract

The work presented in this thesis deal with the contributions of soft computing techniques to identify the parameters of a nonlinear system. Our study focuses primarily on the introduction of several notions of non linear systems and conventional parametric identification techniques and their limitations. Secondly, we introduce the presentation of methods of soft computing: the genetic algorithm and particle swarm.

In this study, to improve the accuracy and convergence of optimal solutions, we present an approach for identification based on genetic algorithms that incorporates a strategy of weight factor (of the objective function) that change along the optimisation procedure then, we propose an algorithm genetic integrating a modified real coding variable. Another optimization approach based on a modified particle swarm algorithm was proposed.

Finally, we apply and evaluate these proposed techniques for the parameters identification of an induction motor used as a typical example of nonlinear system.

Key words: *Parameter identification, nonlinear systems, genetic algorithms, PSO, induction motor.*

ملخص الأطروحة:

الأعمال التي عرضت في هذه الأطروحة هي مساهمات لتطبيق تقنيات الحوسبة المرنة لتحديد وسائط الأنظمة الغير خطية. دراستنا تركز في المقام الأول على تقديم بعض التعريفات و المفاهيم الأساسية كالأنظمة الغير خطية والتقنيات الكلاسيكية لتحديد الوسائط وتطبيقها وكذلك محدوديتها وفي مرحلة ثانية انصب تركيزنا على دراسة الخوارزميات الجينية و سرب الطيور.

لتحسين دقة النتائج المتحصل عليها، قمنا بإدخال تحسينات على الخوارزميات الجينية التي تضمنت إستراتيجية لتغيير أوزان دالة الهدف، و كذلك من خلال اقتراح تعديل التشفير يسمى الترميز الحقيقي، كما تم إدخال بعض التغييرات على خوارزميات سرب الطيور.

وأخيرا لتقييم مدى إمكانية تطبيق هاته الخوارزميات قمنا في هذا العمل بتطبيقها على تحديد الوسائط الخاصة بمحرك لا تزامني كمثال على نظام غير خطي!!

!

مفتاح الكلمات!! تحديد الوسائط ، أنظمة غير خطية ، الخوارزميات الجينية ، خوارزميات سرب الطيور، محرك لا

تزامني. !

Remerciements

Je tiens à remercier Pr. Khier Benmahammed, de l'Université de Setif d'avoir dirigé mes travaux de thèse durant ces années de thèse. Je le remercie également pour la liberté et la confiance qu'il m'a toujours accordées.

Je remercie Pr, Betka Achour de l'Université de Biskra qui me fait l'honneur d'avoir accepté de présider le jury de cette thèse.

Je remercie, Pr. Taleb Ahmed Abdelmalik Directeur de Recherche au laboratoire LAMIH de l'université de Valenciennes, France de m'avoir accueilli dans son équipe et de son soutien moral au cours de mes travaux de thèse.

Je remercie Dr. Benchouia Mohamed Taoufik, Maître de conférence à l'université de Biskra, d'avoir accepté de juger ce travail.

Je remercie Dr. Aissaoui Abdelghani, Maître de conférence à l'université de Bechar pour son aide précieuse et d'avoir accepté de juger ce travail.

Je remercie Dr. Ahmed Tahour Maître de conférence à l'université de Mascara pour son aide précieuse et d'avoir accepté de juger ce travail.

Je n'oublie évidemment pas mes collègues à la faculté des sciences et technologie et les enseignants du département de génie électrique et tous mes amis.

Sommaire

Liste des figures	I
Liste des Tableaux	III
Liste des symboles et d'abréviations	IV
Introduction générale	1

Chapitre 1

Identification des systèmes de “Hard Computing” au “Soft Computing”

I.1. Introduction.....	4
I.2. Représentation des systèmes physiques.....	4
I.2.1 Système linéaire et système non linéaire	5
I.3 Modélisation.....	6
I.3.1 Modèles de connaissance.....	7
I.3.2 Modèle de représentation.....	8
I.4 Identification des systèmes.....	8
I.4.a Identification non paramétrique	10
I.4.b Identification paramétrique.....	10
I.5. Identification basée sur l'erreur de sortie	11
I.6. Identification basée sur l'erreur de prédiction.....	11
I.7 Le critère ou fonction du coût	11
I.8 L'algorithme d'optimisation	13
I.9 Identification en temps différé (Off line)	13
I.10 Identification en temps réel- méthodes récursives (on line).....	13
I.11 Limitation des méthodes classiques.....	14
I.12 Les techniques Soft Computing.....	14
I.12.1 Aperçu sur les travaux réalisés « état d’art ».....	16
I.13 Conclusion.....	19

Chapitre 2:

Les techniques Soft Computing : Les Algorithmes génétiques

II.1 Introduction	20
II.2 Les algorithmes évolutionnaires.....	20
II.3 Les algorithmes génétiques.....	22
II.3.1 Présentation générale des AG.....	23
II.3.1.1 Individu et population.....	23
II.3.1.2. Evaluation des individus de la population.....	25
II.3.1.3 Opérateurs Génétiques.....	26
II.3.1.3.1 Sélection.....	26
II.3.1.3.2 Le croisement.....	27
II.3.1.3.2.a Le croisement à un point.....	28
II.3.1.3.2.b Le croisement en deux points.....	28
II.3.1.3.3 La mutation.....	29
II.3.1.4 Le codage.....	31
II.3.1.4.1 Le codage binaire.....	31
II.3.1.4.2 Le codage réel.....	31
II.4 Les critères de convergence.....	32
II.4.1 Nombre maximal de générations.....	32
II.4.2 Critère sur l'évolution du meilleur individu	32
II.5 Avantages et inconvénients des algorithmes génétiques.....	32
II.5.1 Les avantages des AGs.....	32
II.5.2 Les inconvénients des AGs.....	33
II.5.2 Les inconvénients des AGs.....	33
II.6 Algorithmes génétique proposés.	33
II.6.1 Un algorithme génétique dont la fonction objective est à coefficients de pondération variable	33
II.6.2 Un algorithme génétique à codage réel modifié.....	34
II.6.2.1 Codage (réel codé décimale)	35
II.7 Conclusion.....	37

Chapitre 3:

Les techniques Soft Computing : Essaim de particules

III.1 Introduction.....	38
III.2 Optimisation par essaim particulaire.....	38
III.3 Fonctionnement général de l'OEP.....	39
III.4 Principe de l'évolution de l'essaim de particules.....	40
III.5 Configuration de la méthode.....	43
III.5.1 Le nombre de particules.....	43
III.5.2 La topologie et le choix du voisinage.....	43
III.5.3 Le facteur d'inertie.....	44
III.5.4 Les coefficients de confiances.....	44
III.5.5 Coefficient de constriction.....	45
III.5.6 Initialisation de l'essaim.....	46
III.6 Critères d'arrêt.....	46
III.7 Optimisation par Essaim Particule proposé.....	46
III.8 Conclusion.....	47

Chapitre 4:

Les techniques Soft Computing pour l'identification des systèmes non linéaire.

VI.1 Introduction.....	48
IV.2 Identification des paramètres d'une machine asynchrone	48
IV.2.1 Modélisation de la machine asynchrone	49
IV.2.2 Représentation d'état de la machine asynchrone alimenté en tension.....	49
IV.3 Procédure de l'identification des paramètres du moteur asynchrone par les AGs	51
IV.4 Simulation de l'identification des paramètres du moteur asynchrone par les AGs	54
IV.4.1 Résultats et discussion	58
IV.5 Simulation de l'identification des paramètres du moteur asynchrone par les AGs avec le codage modifié	65

IV.5.1 Résultats et discussion.	66
IV.6 Simulation de l'identification des paramètres du moteur asynchrone par la technique d'OEP modifiée.	69
IV.6.1 Résultats et discussion.	69
IV.8 Comparaison des trois méthodes.....	73
IV.9 Conclusion.....	73
<u>Conclusion Générale et perspectives</u>	75
<u>Bibliographie</u>	77

Liste des figures

Figure I.1 Représentation d'un système.....	5
Figure I.2 Etapes de l'identification.....	9
Figure I.3 Structure de l'identification basée sur l'erreur de sortie.....	12
Figure I.4 Structure de l'identification basée sur l'erreur de prédiction	12
Figure II.1 Organigramme d'un algorithme évolutionnaire.....	21
Figure II.2 Principe du fonctionnement d'un algorithme génétique	24
Figure II.3 Mutation.....	30
Figure II.4 Chromosome modifié (cas d'une fonction objectif à deux paramètres)	34
Figure II.5 Exemple d'un codage réel codé décimale.....	36
Figure III.1 Principe de déplacement de la particule.....	40
Figure III.2 L'organigramme de l'algorithme EP.....	42
Figure III.3 Un graphe (topologie) d'information circulaire d'un essaim de 7 particules.....	44
Figure IV.1 Identification des paramètres par les algorithmes génétique	52
Figure IV.2 Simulation du courant statorique Fig (a) et de la vitesse de rotation Fig (b)	56
Figure IV.3 Méthodologie d'identification des paramètres.....	57
Figure IV.4 Evolution de la fonction de coût avec les poids variables	60
Figure IV.5 Evolution relative de l'erreur % des paramètres du moteur asynchrone	61
Figure IV.6 Identification des paramètres par l'algorithme génétique amélioré	62
Figure IV.7 La minimisation de la fonction de coût avec les poids constants.....	64
Figure IV.8 Comparaison entre le courant statorique obtenu avec les paramètres identifiés et les paramètres réels du moteur	64
Figure IV.9 Evolution de la fonction de coût utilisant l'AG avec codage (RCD)	66

Figure IV.10 Evolution de l'erreur relative des paramètres du moteur asynchrone identifiés par l'AG avec codage (RCD)	67
Figure IV.11 Evolution des paramètres par l'AG avec codage (RCD)	68
Figure IV.12 Evolution de la fonction objectif de l'OEP modifiée.	69
Figure IV.13 Evolution de l'erreur relative des paramètres du moteur asynchrone basée sur OEP modifié	71
Figure IV.14 Identification des paramètres du moteur asynchrone basée sur OEP modifié	72

Liste des Tableaux

Tableau IV.1 Paramètres du moteur asynchrone	54
Tableau IV.2 Les Caractéristiques de l'Algorithme Génétique	58
Tableau IV.3 Résultats de l'identification des paramètres du moteur asynchrone par l'algorithme génétique améliorée avec sept essais.....	59
Tableau IV.4 Résultats de l'identification des paramètres du moteur asynchrone	59
Tableau IV.5 Comparaison du temps d'exécution de l'AG avec une fonction objectif à poids variables et constantes	63
Tableau IV.6 Les Caractéristiques de l'AG avec le codage (RCD)	65
Tableau IV.7 Résultats de l'identification des paramètres par l'AG avec codage (RCD) ..	66
Tableau IV.8 Les paramètres de l'EP modifié	69
Tableau IV.9 Résultats de l'identification des paramètres par l'OEP modifiée	70
Tableau IV.10 Comparaison des trois méthodes d'identifications.	73

Liste des symboles et d'abréviations

AG	Algorithme Génétique.
AGCB	Algorithme Génétique à codage Binaire.
AGCR	Algorithme Génétique à codage réel.
AGCRCD	Algorithme Génétique à codage réel codé décimal.
EP	Essaim de particule.
OEP	Optimisation essaim de particules.
SC	Soft Computing.
MI	Moteur à Induction
MSAP	Machine Synchrone à aiment permanent.
g_{\max}	Le nombre maximum de génération
P_c	La probabilité de croisement
P_m	La probabilité de mutation
J et f	Le critère ou la fonction coût
p_{best}	La meilleure position atteinte par une particule de l'essaim
g_{best}	La meilleure position atteinte par toutes les particules de l'essaim.
R_s et R_r	Les résistances du stator et du rotor.
L_s et L_r	L'inductance du stator et du rotor.
L_m	L'inductance mutuelle.
I_s	Le courant statorique.
i_{ds} et i_{qs}	Les composantes du courant statorique.
φ_{dr} et φ_{qr}	Les composantes du flux rotorique.
v_{ds} et v_{qs}	Les composantes de la tensions d'alimentation statorique.
ω_r	La vitesse de rotation angulaire du moteur.
C_1 et C_2	Les poids de pondération de la fonction objectif.

Introduction Générale

L'identification d'un système représente actuellement l'une des plus importantes opérations de l'ingénierie à cause de son applicabilité à des tâches en rapport à des systèmes industriels. Identifier un système revient donc, à trouver un modèle mathématique capable de décrire ses caractéristiques temporelles ou/et fréquentielles. Ce modèle peut être linéaire ou non linéaire.

Depuis longtemps, l'hypothèse de linéarité a été privilégiée en supposant que la partie non linéaire du système, quand elle existe, est négligeable. Ceci explique le fait que les modèles linéaires ont beaucoup été étudiés et qu'une littérature complète sur le sujet est désormais disponibles [1], [2]. Depuis quelques dizaines d'années et grâce au développement de nouveaux calculateurs très puissants, les structures non linéaires ont été de plus en plus utilisées pour modéliser les systèmes physiques. Grâce à cette connaissance croissante de plusieurs nouveaux types de modèles mathématiques non linéaires, l'identification a connu durant la dernière décennie un fort engouement dans des domaines aussi variés que la physique, les statistiques ou la médecine.

L'identification a pour but d'estimer les paramètres d'un modèle mathématique qui reproduit le comportement entrée/sortie d'un système donné. Ce modèle mathématique décrit le système à l'aide d'une expression mathématique reliant les différentes variables du système. Plusieurs caractéristiques peuvent être attribuées à un modèle mathématique : Il est dit à temps continu ou à temps discret, déterministe ou stochastique, linéaire ou non linéaire. Ces modèles sont beaucoup plus utilisés dans le domaine de la physique et de l'ingénierie. La problématique de l'identification d'un modèle mathématique peut être définie comme suit: Etant donné une séquence d'entrée/sortie, nous cherchons à modéliser le système (représenté par une boîte noire) à l'aide d'un modèle mathématique qui soit représentatif de la réalité dans un certain domaine.

Le problème de l'identification paramétrique peut être mis sous la forme mathématique suivante :

Trouver un vecteur de paramètres Θ , solution d'un problème d'optimisation déterministe [3] :

$$\theta = \arg \min \{J(\theta); \theta \in S\}$$

$J(\theta)$ étant une fonction de coût définie sur $S \subset \mathcal{R}^d$. De nombreux critères sont possibles. Mais, parmi eux, le critère du minimum de l'erreur quadratique moyenne est le plus couramment utilisé. Ce critère signifie que nous souhaitons trouver le modèle optimal de telle manière que l'erreur quadratique soit minimale. Présentons certains arguments en faveur de ce critère. Comme il s'agit d'une quantité quadratique, il possède une interprétation physique et représente l'énergie de l'erreur. D'un point de vue mathématique, il peut être interprété comme un critère de distance [4].

Ce problème d'identification paramétrique peut être résolu à l'aide d'algorithmes classiques. Deux catégories d'algorithmes d'identification paramétrique sont utilisées :

- Les algorithmes non récursifs sont basés sur la minimisation de l'Erreur Quadratique Moyenne (EQM), [5].

- Les algorithmes récursifs sont des algorithmes en ligne traitant directement les mesures d'entrée/sortie. Ils sont aussi appelés adaptatifs puisqu'ils permettent l'ajustement du modèle aux données. Parmi ces méthodes, on trouve l'algorithme des Moindres Carrés (MC), l'algorithme des Moindres Carrés Récursifs (MCR).

En général, les étapes de l'identification expérimentale d'un système consistent à choisir d'abord un modèle paramétrique supposé approcher le comportement du système à identifier. Puis, des méthodes d'estimation sont utilisées pour estimer les paramètres du modèle. Ainsi, une structure expérimentale est choisie et les paramètres correspondants sont estimés. Le modèle (parfaitement connu) est alors testé pour voir s'il reproduit bien la sortie du système. Si le modèle n'arrive pas à suivre les variations du système, une autre famille de modèles plus complexes est testée, les paramètres sont estimés et le modèle est validé [1] [6].

L'identification des systèmes dynamiques non linéaires à partir d'un ensemble de données entrée/sortie est d'une importance fondamentale pour les applications pratiques puisque beaucoup de systèmes physiques possèdent des caractéristiques non linéaires.

L'identification des systèmes non linéaires par les méthodes dites classiques peuvent se révéler insuffisamment applicable pour certains processus et présente malheureusement une certaine limitation dans des situations pratiques.

Récemment, des méthodes d'identifications basées sur les techniques dites 'soft computing' ont été successivement proposées pour certains problèmes d'identification. Parmi ces méthodes on peut citer les réseaux de neurones [7], les algorithmes génétiques [8], les algorithmes à essaim de particules [9] et les colonies de fourmis [10]...etc. Ces approches

simulent des phénomènes physiques, tel le comportement humain ou l'évolution des êtres vivant.

La motivation principale de cette thèse est d'utiliser les techniques 'soft computing' dans les problèmes d'identification paramétrique. Cela nécessite de développer des algorithmes avancés pour l'identification des paramètres d'un système non linéaire.

L'objectif principale de ce travail consiste au développement et à l'amélioration des algorithmes 'soft computing' en vue de les appliquer pour l'identification des paramètres d'un système non linéaire à savoir :

- Amélioration de la convergence des algorithmes.
- Amélioration de la précision des résultats de ces Algorithmes.

Le plan que nous allons donc suivre est le suivant :

Le premier chapitre présente l'état d'art des techniques d'identification en allant des techniques classiques 'hard computing' fondée sur le principe de moindre carré aux techniques récemment développées 'soft computing'. Nous introduisant plusieurs notions utiles sur les systèmes et les techniques d'identification conventionnelle et leurs limitations. Nous introduisant les techniques 'soft computing' en soulignant le rôle important qu'elle jouent dans le domaine de l'identification des systèmes non linéaire

Le deuxième chapitre concerne une présentation théorique des algorithmes génétique. Par la suite on introduit une modification où on propose un algorithme génétique qui intègre une stratégie de variation de poids de la fonction objectif. Nous proposons aussi un algorithme génétique avec un codage réel modifié de la variable.

Le troisième chapitre nous présente l'essaim de particules (EP) sa structure et les différents opérateurs qui la composent ainsi que la modification apporté en vue d'améliorer sa convergence vers la solution optimale.

Le quatrième Chapitre aborde la mise en application des techniques 'soft computing' pour l'identification des paramètres d'un système non linéaire. Ainsi, l'application au moteur asynchrone va être choisie comme exemple typique de la problématique.

Dans la partie Conclusion et perspectives on expose une synthèse des contributions de cette thèse et on présente les travaux de recherche futurs en liens avec la problématique traitée dans ce travail.

CHAPITRE 1

Identification des systèmes de “Hard computing” au “Soft computing”.

I.1.Introduction

L'identification des systèmes a connu un développement considérable durant la dernière décade du XX^e siècle tant sur le plant pratique par l'application des techniques existantes que sur le plant théorique par le développement de nouvelles techniques. Cette progression s'explique par le fait que l'identification est une discipline commune à de nombreux domaines des sciences de l'ingénieur (traitement du signal, statistique, biologie...). Cette diversité d'application implique inévitablement une grande variété de problèmes posés ainsi qu'une multiplicité de solutions proposées par certaines classes [1] [2].

Ce chapitre présente en première partie une vue générale sur les notions de systèmes et modélisation. La deuxième partie est consacrée à introduire l'identification des systèmes et des différentes techniques d'identification classique ainsi que leurs limitations d'applications. Un état de l'art des techniques soft computing et leurs applications dans le domaine de l'identification des paramètres des systèmes non linéaires est présenté en dernière partie.

I.2 Représentation des systèmes physiques.

La notion de système est fondamentale en identification. Un système est défini par ses constituants et les interactions qui existent entre eux, et l'ensemble représente une entité individualisée. Par système, on signifie souvent processus. L'importance de la notion de système réside dans sa généralité. En effet, un système ou un processus peuvent être de nature quelconque : mécanique, électrique, électromécanique, biologique, chimique, physico-chimique, économique, industriel, etc.

Pour ce qui concerne la théorie des systèmes, on considère le système (ou le processus) évoluant dans son environnement et pouvant interagir avec lui. A priori, le système peut être considéré comme une boîte noire ou black box en anglais. Il est dès lors important, en premier lieu, de distinguer les grandeurs d'entrée (inputs) et les grandeurs de sortie

(outputs) du système étudié. Ensuite, il est important d'essayer de déterminer les relations qui les relient et de connaître la nature et les modes d'interaction avec l'environnement.

La figure 1.1 propose une représentation de la notion du système où certaines grandeurs proviennent du système et agissent sur l'environnement, ce sont les sorties, représentées par un vecteur noté (y). D'autres caractérisent l'état interne du système et influent sur les sorties ce sont les états (x). Enfin, les dernières proviennent de l'environnement et agissent sur le comportement du système. On distingue les entrées ou commandes (U) qui sont contrôlées par l'utilisateur et les bruits (b) qui parasitent le comportement du système et ne sont pas contrôlés par l'utilisateur.

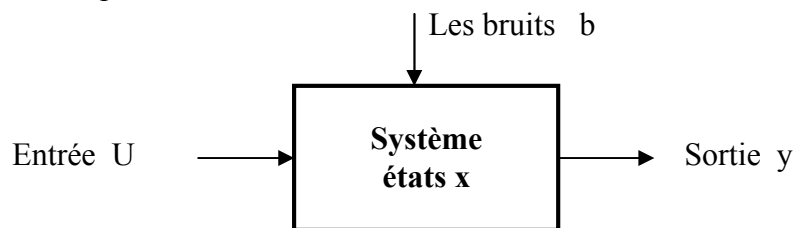


Figure. I.1 Représentation d'un système

Si l'ensemble des grandeurs du système ne peut prendre qu'un nombre fini de valeurs différentes alors le système est dit à événements discrets. A l'inverse si les grandeurs du système peuvent prendre une infinité de valeurs le système est à événements continus.

Si un système excité par des entrées constantes dans le temps a ses sorties constantes dans le temps, le système est statique. A l'inverse, si ses sorties évoluent dans le temps, le système est dynamique [11].

I.2.1. Système linéaire et Système non linéaire

On va considérer des systèmes non linéaires (c'est à dire des systèmes dont les lois comportent des équations algébriques non linéaires). Toutefois, avant de se lancer dans l'entreprise de modélisation et identification d'un système physique, il est essentiel de déterminer l'utilité de recourir à des modèles non linéaires. En effet, même si la majeure partie des phénomènes physiques fait intervenir des non linéarités, les systèmes étudiés, sous certaines conditions, peuvent être correctement approchés par des modèles linéaires. Un exemple classique étant la modélisation d'un système non linéaire autour d'un point de fonctionnement : en effet, pour de faibles variations des grandeurs du système, les non linéarités peuvent être négligées et un modèle linéaire convient parfaitement.

Dans bien des cas, c'est la connaissance physique du système qui amène à considérer l'utilisation d'un modèle non linéaire. En pratique, on cherchera souvent à utiliser en premier lieu des modèles linéaires. Si ceux-ci ne permettent pas de modéliser avec une précision suffisante le comportement du processus, alors seulement, une modélisation non linéaire est envisagée. En effet, l'utilisation de modèle non linéaire complexifie l'identification et soulève des problèmes particuliers de stabilité, d'analyse et de minima locaux (en fonction de la technique d'identification utilisée) [11].

I.3. Modélisation

La modélisation a acquis dans toutes les disciplines techniques une importance incontournable. L'intérêt de la modélisation est multiple, mais son apport fondamental est de tendre à remplacer, dans divers opérations les processus physiques par des objets mathématiques censés avoir le même comportement. La modélisation est souvent considérée comme la phase initiale de l'automatique moderne. Elle permet en effet de déterminer le modèle du procédé qui sera utilisé pour prédire son comportement futur, diagnostiquer ses évolutions de fonctionnement ou synthétiser son régulateur et sa loi de commande.

Ainsi, on s'intéresse très souvent à trouver une relation entre un ensemble de variables observées dans un intervalle de temps $[t_0, t_0 + T_i]$. En général, certaines de ces variables, appelées variables de sorties ou réponses, notées par y_1, y_2, \dots, y_m . Les autres variables u_1, u_2, \dots, u_r appelées variables d'entrée sont utilisées pour prédire ou expliquer le comportement de y_1, y_2, \dots, y_m .

On suppose souvent que la liaison entre $y_i, i = 1, \dots, m$ et $u_i, i=1, \dots, r$ est exprimée par des relations mathématiques $f_i, i = 1, \dots, m$ dont les formes sont à priori connues, à l'exception de quelques constantes inconnues appelées paramètres, p_1, \dots, p_p . Cette liaison peut être exprimée sous la forme vectorielle suivante :

$$y \approx f(u, p) \quad (1.1)$$

Où $y = [y_1, \dots, y_m]^T$, $f = [f_1, \dots, f_m]^T$, $p = [p_1, \dots, p_p]^T$ et $u = [u_1, \dots, u_r]^T$. On pourrait imaginer que le vecteur d'entrée contient les entrées du système u et les états du système x . Dans ce cas :

$$y = f(u, x_0, t, p) \quad (1.2)$$

Où f est une relation fonctionnelle qui décrit une application entrée-sortie initialisée.

Cette relation reste toujours approximative, d'une part à cause de l'influence des variables non maîtrisées et plus au moins inconnues et d'autre part en raison de l'erreur de mesure.

La fonction f , selon la réquisition de la connaissance à priori sur le système ayant généré les données, peut être soit un modèle de connaissance soit un modèle de représentation [12]. Une fois la structure du modèle choisi, on peut l'identifier en estimant les paramètres p_i , par l'optimisation d'un critère d'optimalité, qui peut consister à mesurer la ressemblance entre les sorties observées et les sorties du modèle [13], [5].

I.3.1 Modèles de connaissance

Ces modèles font appel aux connaissances des spécialistes du domaine considéré, consiste à regrouper, généralement sous forme de systèmes différentiels, algébriques ou graphiques, les lois et relations physique qui décrivent le comportement du processus.

Les modèles de connaissances sont construits à l'aide des lois physiques. Ils se prêtent donc à la prise en compte des informations à priori disponibles et au contrôle à posteriori des ordres de grandeur des paramètres obtenus, car ces paramètres ont un sens physique.

Les modèles de connaissance sont potentiellement riches, ils contiennent toute information nécessaire sur les régimes statiques et dynamiques du système. Autrement dit, ils sont bien adaptés à une simulation détaillée en vue d'une prédiction de comportement à long terme.

Un modèle de connaissance d'un système complet pourrait être [13] :

$$\begin{cases} \dot{x}(t, p) = g[x(t, p), u(t), t, p] \\ y(t, p) = h[x(t, p), p] \\ x_0 = x(t_0, p) \\ \Psi[x(t, p), u(t), p] \geq 0 \end{cases} \quad (1.3)$$

Où $x \in R^n$, $u \in R^r$ et $y \in R^m$ sont respectivement les vecteurs d'état, d'entrée et de sortie. g et h sont deux fonctions vectorielles paramétrisées par p qui définissent les relations (connues) entre la sortie y , l'entrée u et l'état x . Notez que dans la représentation ci-dessus, les fonctions g et h jouent le rôle de la fonction f dans (1.2). Enfin, Ψ représente toutes les contraintes supplémentaires d'égalité et d'inégalité algébriques connues à priori; reliant x , u et p .

I.3.2. Modèle de représentation

Dans de nombreuses situations pratiques, la complexité du procédé à modéliser est telle qu'il est difficile, voire impossible, de connaître ou d'associer les lois physiques gouvernant la dynamique du système. Il est alors nécessaire de faire appel à un modèle "boite noire", construit à partir de mesures des données d'entrée sortie du procédé.

Ces modèles, appelés aussi modèles boites noires, reproduisent un comportement observé, sans requérir aucune connaissance à priori sur le système. Il n'est même pas nécessaire de savoir ce que représente la sortie ni en quelle unité est exprimée.

Ils sont faciles à établir, limités, mais efficaces dans un domaine de fonctionnement. Les modèles ARX (Auto Regressive Model), ARMAX (Auto Regressive Moving Average) sont quelques exemples de modèles de représentation linéaire par rapports aux entrées. En ce qui concerne les modèles de représentation non linéaires, on peut citer les réseaux de neurones; les ondelettes et les modèles à base de la logique floue [5], [15].

Un autre inconvénient des modèles de représentation est la difficulté du choix de la complexité du modèle. Par exemple, la détermination du nombre de couche cachée et du nombre d'unités dans chaque couche cachée d'un réseau de neurones pour un problème donné est loin d'être facile.

Le choix entre un modèle de connaissance et un modèle de représentation dépend de la connaissance à priori disponible sur le système et de la facilité de l'estimation des paramètres.

I.4. Identification des systèmes

L'identification constitue une phase importante dans la définition du modèle, c'est par elle que le choix de la classe de modèle à adopter puis les valeurs des paramètres qui le caractérisent vont ce préciser. Le plus souvent l'identification s'effectue en optimisant un critère de qualité qui caractérise l'écart entre le comportement du processus (repéré par un ensemble de mesures), et celui de son modèle. Il est important de respecter une procédure rigoureuse pour identifier un procédé (Figure.I.2):

- Détermination d'un protocole de test : propriétés statistiques des signaux d'entrée pour balayer toutes les fréquences intéressantes, le ratio signal/bruit doit être suffisamment important et le nombre de points de mesures doit être significatif pour le test (>1000).

- Détermination de la structure du modèle : type de modèle.
- Identification : choix d'un algorithme pour trouver le modèle en minimisant les erreurs entre les mesures et le modèle, en général algorithme basé sur la méthode des moindres carrés.
- Validation du modèle : Réalisation de plusieurs tests de vérification. Il est nécessaire pour cette étape d'utiliser des mesures différentes de celles utilisées lors de l'identification.

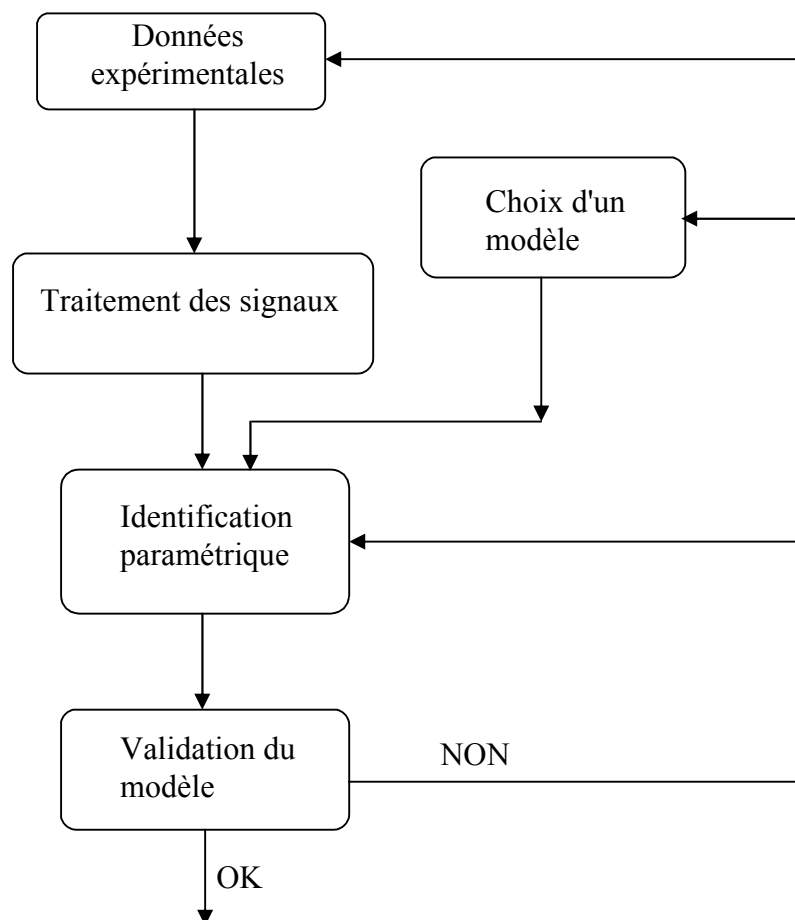


Figure I.2. Etapes de l'identification des systèmes [70].

Identifier un système consiste à obtenir ces caractéristiques à partir d'observations des réponses du système aux tests d'entrée, le monde de l'identification est très vaste. Nous retrouvons une multitude de techniques d'identifications regroupées en deux grandes familles:

- 1- L'identification non paramétrique.
- 2- L'identification paramétrique.

I.4.a. Identification non paramétrique

Quand nous parlons d'identification non paramétrique, nous entendons l'interprétation graphique ou sous forme de tableau d'une réponse transitoire, fréquentielle (Bode, Black, Nyquist, Fourier) ou par corrélation (réponse pulsionnelle),...etc. Avec ces méthodes nous ne pouvons obtenir qu'une idée de la dynamique du système; qui peut servir par exemple; au dimensionnement approximatif de régulateurs ou de filtres. Si la précision de la connaissance du système doit être plus élevée; il faut passer à l'identification paramétrique [70].

I.4.b. Identification paramétrique

Le but de l'identification paramétrique est d'estimer les paramètres d'un modèle mathématique; de façon à obtenir une représentation satisfaisante du système réel étudié. Dans ce genre d'identification nous retrouvons différentes techniques. L'une d'elles est appelée "identification heuristique", elle se base sur la détermination des paramètres d'une fonction de transfert en ayant à disposition la réponse du système à un échelon d'entrée. Cette technique peut être suffisante pour des systèmes simples. Une autre technique appelée "régression linéaire" est utilisée dans la méthode de moindre carrée simple. Nous retrouvons aussi les méthodes basées sur l'erreur de sortie et sur l'erreur de prédiction ainsi que les méthodes récursives.

Une procédure d'identification implique différents choix, notamment:

- Le type du modèle à utiliser.
- Le critère
- L'algorithme d'optimisation.
- Le type d'essai à effectuer.
- La vérification des résultats

Ces choix doivent être remis en question à la fin de la procédure, si la vérification donne de grande différence entre le système et le modèle, les causes peuvent être différentes, le modèle ne représente pas au mieux la physique du système, le critère n'est pas bien adapté par rapport au résultat souhaité, l'algorithme d'optimisation n'est pas performant à cause du temps de calcul ou des problèmes de convergence [70].

I-5. Identification basée sur l'erreur de sortie

De façon générale le principe est très simple. Le système réel et le modèle mathématique du système sont excités par les mêmes entrées u ; les sorties du système réel y et y_m sont comparées pour créer le vecteur des erreurs de sortie e . Un algorithme d'optimisation, change les paramètres regroupés dans le vecteur P pour que le critère J , qui est fonction de e , devient minimal. Si J est minimal, alors nous avons trouvé le meilleur estimé de P au sens du critère choisi. L'erreur de sortie est par conséquent minimale. La figure I.3 présente le principe [17].

I.6. Identification basée sur l'erreur de prédiction

Ce type d'identification se base sur le concept de prédiction de la sortie du système réel à l'aide d'équations algébriques de prédiction. Le principe est de la même nature de la commande prédictive, où l'on cherche à minimiser l'écart entre la consigne et la sortie en calculant la variable manipulée (la variable à la sortie du prédicteur). Le principe est illustré dans la figure I.4. Ces méthodes peuvent être facilement implantées en temps réel [17].

I.7. Le critère ou fonction du coût

Le critère est la partie de programme qui définit la fonction objectif (fonction coût ou critère, généralement notée J) qui est la fonction à minimiser ou à maximiser pour que la sortie du modèle mathématique et celle du système réel soient les plus proches possible. L'une des questions importantes en identification est la suivante : quelle fonction coût ou encore quel est l'estimateur qui permet de mieux calculer la valeur de P ou de s'y rapprocher le plus possible

Le critère à minimiser est une fonction, souvent quadratique; qui dépend de l'erreur de sortie réelle du système et de la sortie du modèle; qui est fonction du vecteur des paramètres :

$$J = f(y(k) - y_m(k, P)) = f(\varepsilon(k, P)) \quad (1.4)$$

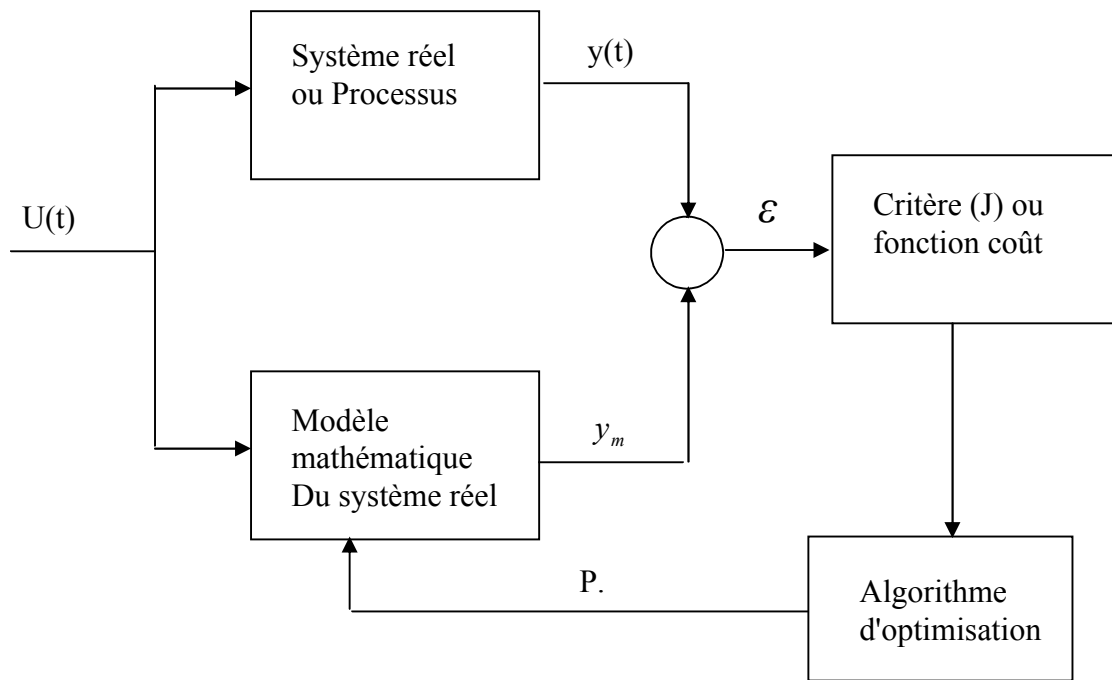


Figure I.3 Structure de l'identification basée sur l'erreur de sortie

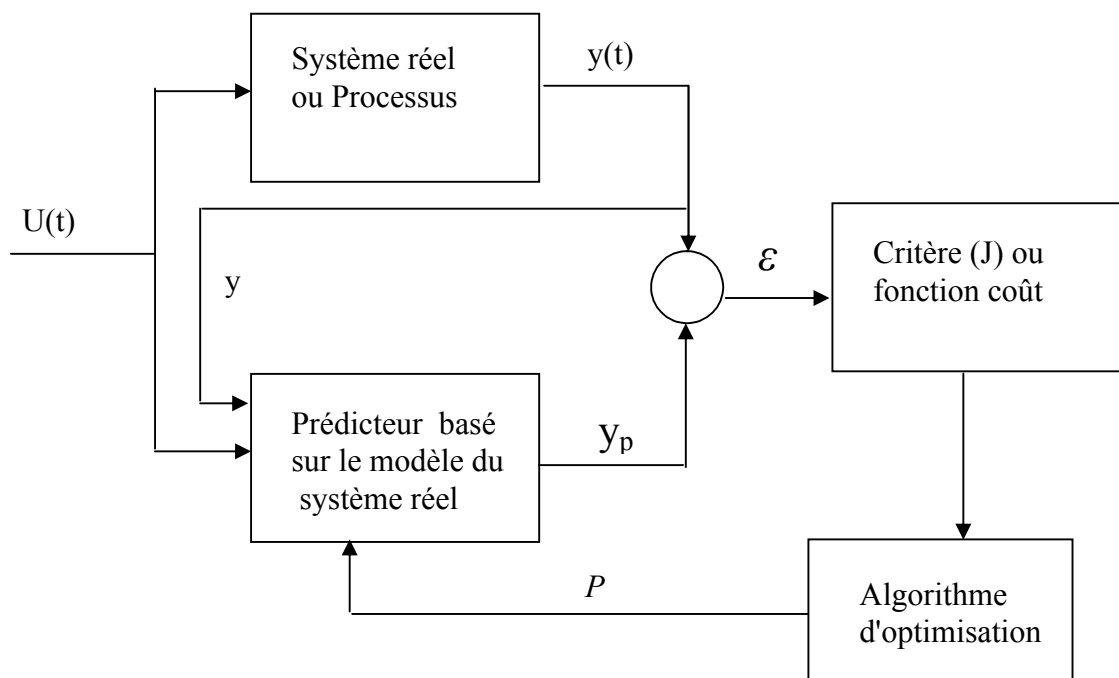


Figure I.4 Structure de l'identification basée sur l'erreur de prédiction

I.8. L'algorithme d'optimisation

Différents types d'algorithme d'optimisation sont disponibles dans la littérature en fonction de la méthode de convergence désirée:

- Méthode du gradient.
- Méthode Newton-Raphson.
- Gauss-Newton.
- ...etc.

La solution ou l'optimum, doit être trouvée le plus rapidement possible à l'aide de contraintes physiques du système (signes des paramètres, stabilité). Quand l'algorithme d'optimisation a convergé, nous avons le vecteur optimal des paramètres P . Les différents types d'algorithmes se distinguent par leur façon de converger vers l'optimum.

I.9. Identification en temps différé (Off-line)

L'identification en temps différé se base sur le fait d'avoir enregistré toutes les mesures de l'essai avant de passer à l'algorithme d'identification. À chaque itération de l'algorithme d'optimisation, nous faisons une simulation complète du modèle et à la fin nous calculons l'erreur de sortie en comparant les sorties du modèle à celles du système qui ont été au préalable enregistrées. Ensuite l'optimisateur propose d'autres valeurs de paramètres pour chercher de se rapprocher de l'optimum. La boucle recommence en lançant une nouvelle simulation avec les nouveaux paramètres. Ce type d'identification peut prendre beaucoup de temps à cause de l'obligation de re-simuler le modèle à chaque itération. Pour une identification en temps différé, nous utilisons normalement des techniques basées sur l'erreur de sortie où le modèle est à chaque fois simulé [70].

I.10. Identification en temps réel-(On-line)

L'identification en temps réel peut être très intéressante pour plusieurs raisons. La première est l'obtention des paramètres à chaque instant permettant ainsi une commande adaptative.

Nous pouvons aussi vouloir suivre l'évolution des paramètres du système pour nous assurer que les variations restent normales (diagnostique en temps réel).

Dans les systèmes non linéaires, ces méthodes nécessitent d'une linéarisation du problème à chaque instant (filtre de Kalman étendu), ce qui implique une perte de précision dans l'estimation paramétrique. Si nous voulons travailler en temps réel, il est nécessaire d'avoir des algorithmes très simples à exécuter; c'est pour cette raison que ce type d'identification est basé prioritairement sur l'erreur de prédiction, ou ne sont présentes que des équations récursives [70].

I.11. Limitation des méthodes classiques

Le principe de l'identification en se basant sur ces méthodes consiste à calculer à chaque instant, le vecteur des paramètres du modèle recherché en fonction des paramètres estimés à l'instant précédent et des nouvelles informations acquise sur le procédé. Les algorithmes cités dans les sections précédentes présentent malheureusement quelques inconvénients qui les rendent difficilement applicable dans certaines situations pratiques:

- La grande majorité des méthodes classiques d'identification ont été développées pour des systèmes mono variables. Il est théoriquement possible de les appliquer à des procédés multi variables. Cependant lorsque le nombre d'entrées et de sorties devient conséquent, l'utilisation de représentation polynomiales n'est pas évidente.

- Ces algorithmes ne permettent pas d'avoir un accès direct à des variables telles que l'état du système, variables particulièrement utiles en filtrage (Luenberger, Kalman) [67], [68] ou en commande optimale.

- De nombreuses techniques récursives ont été proposées pour fonctionner sur une structure de modèle de procédé et de bruit bien spécifique. Il est donc fondamental d'avoir une bonne connaissance a priori du système à étudié pour correctement choisir l'algorithme récursif à appliquer.

Ces difficultés sont principalement liées à la structure du modèle utilisé pour l'approche entrée-sortie: la fonction de transfert. Il semble donc important de développer de nouveaux algorithmes d'identifications basés sur des nouvelles techniques intelligentes.

I.12. Les Techniques Soft computing

Les systèmes complexes qui existent actuellement ne peuvent pas être décrits avec précision par des modèles mathématiques rigoureux, dans l'industrie il ya des besoins

croissants des contrôleurs de très grande précision et d'un comportement autonome dans le contrôle de certain système (exemple la robotique).

Les approches conventionnelles (classiques) utilisées pour comprendre et prévoir le comportement de tels systèmes basés sur de telles techniques d'analyse peuvent se révéler insuffisants. Ces difficultés conduisent à un certain nombre de problèmes difficiles, c'est-à-dire, intégrer l'intelligence humaine dans une machine.

Les Scientifiques et les chercheurs ont bénéficiés de la recherche sur l'intelligence humaine (ou naturelle). Un certain nombre de modèles ont également été mis au point par l'imitation de l'intelligence de l'homme (naturelle) à savoir, les réseaux neuronaux artificiels (RNA) qui permet de réaliser la simulation de l'intelligence par le comportement des neurones dans le cerveau [19][21], la logique floue (FL) par le biais de l'intelligence la simulation de comportement du processus de raisonnement de l'homme, la méthode évolutionnaire qui réalise la simulation de l'intelligence par l'évolution [18] [20] et les systèmes immunitaires artificiels réalise la simulation de l'intelligence par le comportement de mécanismes immunologique avec les antigènes et les anticorps [18] [19].

Afin de faire face aux difficultés mentionnées dans la section précédente, l'émergence d'un cadre des techniques soft computing a été mise au point récemment, présentant les propriétés suivantes:

- Les techniques Soft Computing sont orientées vers l'analyse et la conception de systèmes intelligents. Elles rassemblent les réseaux de neurones artificiels, la logique floue, et l raisonnement probabiliste, y compris algorithmes évolutionnaires, théorie du chaos... etc.
- Les techniques Soft Computing sont destinées à une formalisation de la remarquable capacité humaine à faire décision rationnelle en un environnement incertain et imprécis.
- Les techniques de soft computing sont complémentaires plutôt que concurrentiels. Les expériences acquises au cours de la dernière décennie ont indiqué qu'il peut être plus efficace de les utiliser de manière hybride.
- Les techniques Soft Computing ont un cadre ouvert. Cela signifie que les techniques nouvellement créées proviennent de l'imitation de l'intelligence humaine toujours ouverte à être ajouté au cadre des techniques Soft Computing.

Les techniques Soft computing, selon les idées du professeur Zadeh, diffèrent des méthodes classiques en ce qu'elles sont tolérante de l'imprécision, l'incertitude, la vérité partielle, et le rapprochement. Le principe directeur des techniques soft computing, est d'exploiter la tolérance à l'imprécision, l'incertitude, la vérité partielle et le rapprochement en vue d'atteindre souplesse, robustesse et faible coût des solutions [3]. La capacité de traitement

de l'information d'une manière souple et efficace sont les deux principales propriétés de ces techniques.

Jusqu'à présent, de plus en plus des recherches et des applications d'ingénierie montrent que Les techniques soft computing peuvent jouer un rôle clé dans le contrôle et l'identification des systèmes non linéaires, dans la robotique et le traitement du signal et de l'image...etc.

I.12.1. Aperçu sur les travaux réalisés

De récentes études et travaux de recherches sur l'identification paramétrique des systèmes non linéaires ont été menés. Ces applications se basaient sur différentes techniques de soft computing. Dans ce qui suit nous présentons les travaux significatifs où nous intéresserons seulement aux travaux concernant l'identification paramétrique des systèmes non linéaire où le moteur à induction est pris comme un exemple typique d'un système non linéaire. Cet axe de recherche est très riche aux travaux de chercheurs. Pour chacun d'entre eux, nous mentionnerons La méthode d'identification paramétrique utilisée et l'amélioration apportée à cette technique.

Les travaux **d'Alonge et Al** dans [22], traitent de l'identification "offline" des paramètres des moteurs à induction au moyen des techniques des moindres carrés (LS) et les algorithmes génétiques (GA), en utilisant les tensions et courants statorique et la vitesse en tant que données d'entrée-sortie. La méthode d'identification utilisant l'AG se base sur la détermination des meilleurs paramètres qui correspondent à un comportement d'entrée-sortie du moteur.

Ils ont testé en utilisant deux critères de performance : critère quadratique et le critère absolu. Cependant, les résultats obtenus avec le critère quadratique sont légèrement meilleurs que ceux donnés par critère absolu.

Ils ont testé plusieurs valeurs des poids de la fonction objectif les meilleurs résultats sont obtenus avec coefficients de pondération d'une valeur constante égaux à 0,5 de la fonction objectif utilisée. Les deux méthodes sont étudiées et comparées au moyen d'expériences effectuées sur un moteur à induction de 1 kW.

Dans une étude comparative, sur l'identification des paramètres des moteurs à induction. **Rasmus K. Ursem, P Vadstrup** [23], effectuent une comparaison des performances de huit algorithmes d'optimisation stochastique sur l'identification de deux

moteurs à induction. Ces huit algorithmes représentent les quatre principaux groupes d'algorithmes actuellement utilisés pour l'optimisation numérique. Les quatre groupes sont: la recherche locale (LS), stratégies d'évolution (ES), algorithmes évolutionnaires (AE), et l'optimisation par essaim de particules (PSO). La comparaison ce fait entre un algorithme de base et un autre améliorée de chaque groupe. Les résultats des expériences conduit à tirer deux conclusions. Tout d'abord, les algorithmes améliorés avaient des performances significativement meilleures par rapport à des algorithmes de bases. Cela souligne l'importance d'utiliser des algorithmes avancés à des problèmes d'identification des paramètres du monde réel. En outre, l'amélioration des performances justifie et motive le développement des techniques les plus avancées. Deuxièmement, la population basée sur les techniques d'optimisation stochastique (ES, AE, et les PSO) ont montré nettement leurs performances sur les algorithmes de recherche locale pour l'identification des paramètres de deux moteurs à induction.

Dans [24], **J. Kim et J. Yeol** proposent un algorithme évolutionnaire dont la vitesse de convergence est améliorée, sans avoir une influence sur la diminution de la diversité, l'algorithme proposé a deux opérateurs de mutation en fonction des conditions de l'évolution, respectivement l'un est un opérateur de direction selon la valeur de la fonction objectif. L'autre est une perturbation Gaussian dont la moyenne est nulle. Pour valider l'efficacité de la technique améliorée, une étude comparative entre l'algorithme proposé (qui améliore le taux de convergence) et l'algorithme évolutionnaire est effectuée pour l'optimisation de huit fonctions de test.

H.J, S.M et P.D dans [25], proposent de procéder à l'identification des paramètres pendant la phase de démarrage d'un moteur asynchrone à l'aide de l'AG. Cette étude montre que l'AG est de robustesse très élevé et d'applicabilité étendue. L'identification sous les deux systèmes de références (stator et rotor) a été testée. Une amélioration importante sur la précision de l'identification de tous les paramètres du moteur à induction a été obtenue en combinant les deux systèmes de référence.

Afin de réduire le temps de calcul et améliorer la précision pour l'optimisation des paramètres du modèle de Muskingum, un nouveau algorithme dite algorithme génétique accélérer à codage Gray (AGCG) a été proposé par **J.Chen et X.Yang** [26]. La convergence globale est analysée pour le nouvel algorithme génétique. Son efficacité est vérifiée par l'application au model Muskingum. Afin de faire face avec le compromis entre précision et temps de calcul dans l'algorithme de codage Gray génétiques pour l'optimisation globale sur

les paramètres, une modification de l'espace de recherche a été établit, où l'intervalle de chaque paramètre est diminué avec les meilleurs individus de chaque génération.

Les résultats obtenus sont encourageants, l'efficacité et la précision de l'algorithme amélioré sont beaucoup plus élevées par rapport aux algorithmes existants. Cet algorithme permet de surmonter les phénomènes de Hamming dans des algorithmes génétiques simples.

Dans le travail de **W.Chang** publié dans [27] il a proposé Un algorithme génétique à codage réel (AGCR) appliquée à l'identification avec précision des paramètres inconnus et le contrôle pour une classe de systèmes non linéaires. Il est bien connu en général, qu'un AG à codage réel est plus pratique et adaptée pour faire face à la plupart des applications pratiques d'ingénierie. Dans ce travail, il a utilisé un AG à codage réel pour identifier les paramètres d'un système où sa structure est supposée connue auparavant. Deux exemples de simulation ont été donnés pour démontrer l'efficacité de la méthode proposée.

Dans une autre étude réalisée par N.Tutkun [28], l'estimation des paramètres dans les modèles mathématiques utilisant l'approche des algorithmes génétiques codés réels (AGCR) est présentée. L'approche proposée est mise en œuvre sur les équations différentielle du second ordre oute en effectuant la comparaison avec les approches antérieures. Les résultats montrent que la AGCR est similaire avec le codage binaire des algorithmes génétiques (AGCB) en termes de processus génétiques, il a quelques avantages tels que la haute précision, l'absence du phénomène Hamming, rencontré dans AGB.

Un algorithme génétique à codage réel peut aider à augmenter les performances d'estimation à trouver les valeurs des paramètres améliorés. Le nombre de génération n'est pas très influent sur l'obtention d'une amélioration des paramètres estimés Les résultats obtenus indiquent que l'approche proposée donne de meilleurs résultats estimés par rapport aux résultats antérieurs

Dans l'article [29], les auteurs on proposés une méthode intelligente pour l'identification des paramètres d'un modèle de la machine synchrone à aimant permanent (MSAP) à l'aide de l'approche d'optimisation de particules d'essaim (OEP). Comme une méthode intelligente de calculs basés sur la recherche stochastique, OEP est révélée être un outil polyvalent et efficace pour résoudre ce problème d'ingénierie complexes. La méthode est appliquée à l'identification des paramètres de la MSAP. L'acquisition de ces paramètres au cours de son fonctionnement est une tâche difficile en raison de la non-linéarité inhérente de la dynamique du moteur. Grâce à la simulation et l'expérience, ce document vérifie l'efficacité de la méthode proposée dans l'identification des paramètres du modèle MSAP, la méthode est

généralement applicable à d'autres types de moteurs électriques, et ainsi que d'autres systèmes dynamiques à modèle non linéaire.

Les travaux d'**A.Alfi et H.Modares** dans [30], présentent une méthodologie pour trouver les paramètres du système optimal et les paramètres de contrôle optimal en utilisant un nouvel algorithme d'optimisation d'essaim de particule adaptative (OEPA). Les principaux avantages de l'OEPA proposées sont d'atteindre une vitesse de convergence plus rapide et de précision meilleure des solutions avec un minimum de charge de calcul supplémentaire. Deux exemples de simulation sont donnés pour démontrer l'efficacité de l'algorithme proposé, une comparaison cette technique avec l'algorithme génétique (AG) et OEP a été présentée. Les résultats obtenus montrent que l'OEPA a plus de précision et convergence rapide que l'AG et l'OEP pour l'identification (off line) des paramètres.

I.13. Conclusion

Au cours de ce chapitre nous avons effectué une étude générale sur l'identification en introduisant notion de système, de modélisation, étapes d'identification, les techniques classique, et leurs limitations ensuite nous introduisant les techniques soft computing.

Nous avons fait un état de l'art des travaux qui ont retenu notre attention et nous avons proposé pour chacun d'entre eux un résumé en tenant compte de la technique soft computing adopté à l'identification paramétrique ainsi que des résultats obtenus lors d'application à des systèmes non linéaires.

La maîtrise et la compréhension de l'application des techniques soft computing au domaine d'identification paramétrique des systèmes non linéaires passent obligatoirement par l'étude de chacune d'elle, c'est l'objectif des deux chapitres suivants.

CHAPITRE 2

Les techniques Soft Computing : Algorithmes génétiques

II.1. Introduction

Les ingénieurs se heurtent quotidiennement à des problèmes de complexité grandissante, qui surgissent dans des secteurs très divers, comme le traitement d'images, la conception des systèmes de commande et de diagnostique, etc...le problème à résoudre peut être souvent considéré comme un problème d'optimisation dans lesquels on définit une ou plusieurs fonctions objectives (fonction de cout), que l'on cherche à minimiser par rapport à l'ensemble des paramètres concernés. La résolution d'un tel problème a conduit les chercheurs à proposer des méthodes de résolution de plus en plus performantes, parmi lesquelles on peut citer les méthodes metaheuristiques. Ces dernières qui comprennent notamment la méthode de recuit simulé, la méthode de recherche tabou, les algorithmes de colonies de fourmis, les algorithmes génétiques, etc..., présentent des caractéristiques communes qui sont :

- Raisonnement par analogie avec les systèmes réels (biologie, physique,...).
- Stochastiques.

Et partagent aussi les mêmes inconvénients :

- Difficulté de réglage des paramètres de la méthode.
- Temps de calcul élevé.

Dans ce chapitre nous allons nous intéresser uniquement aux algorithmes génétiques comme étant des méthodes d'optimisation mono objective qui ont connu ces dernières années un développement considérable grâce à l'augmentation vertigineuse de la puissance des calculateurs. Nous détaillons le principe général de fonctionnement d'un algorithme génétique avec une description de ses différentes composantes et nous proposons certaines modifications en vue d'une amélioration de certaines caractéristiques de cet algorithme.

II.2. Les algorithmes évolutionnaires

Les algorithmes évolutionnaires sont une classe d'algorithmes d'optimisation stochastiques qui reposent sur une analogie avec la théorie de l'évolution naturelle des

espèces de Darwin [31]. Dans la nature, les espèces sont capables de s'adapter à leur milieu sous la pression de la sélection naturelle et d'évoluer à partir de leur héritage génétique de manière à accroître leur adaptation au fil des générations. C'est une métaphore du paradigme darwinien qui est utilisée dans les algorithmes évolutionnaires pour la résolution de problèmes d'optimisation.

Dans la figure II.1, on présente la structure générale d'un algorithme évolutionnaire qui est une suite de générations reproduisant le principe l'évolution naturelle en agissant sur une population d'individus divers (généralement tirée aléatoire au départ) à laquelle on applique différents opérateurs (recombinaison, mutation ...) et que l'on soumet à une sélection, à chaque génération alors la population tends à s'améliorer [32]. Les algorithmes évolutionnaires ne nécessitent aucune connaissance du problème : on peut représenter celui ci par une boîte noire comportant des entrées (les variables) et des sorties (les fonctions objectif). L'algorithme ne fait que manipuler les entrées, lire les sorties, manipuler à nouveau les entrées de façon à améliorer les sorties, ... etc. [33].

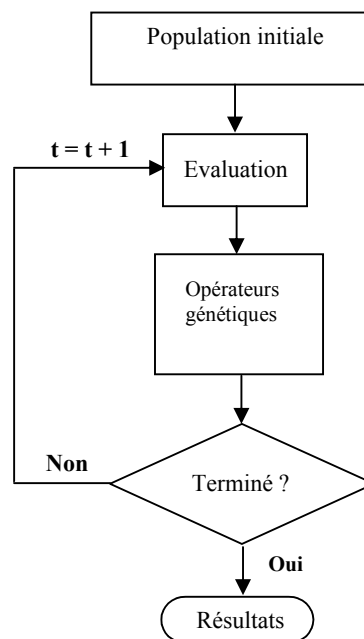


Figure.II.1. Organigramme d'un algorithme évolutionnaire

Depuis les années 80, l'essor de l'évolution artificielle est principalement dû à l'accroissement exponentiel des moyens mis à la disposition des chercheurs, ce qui leur permet d'afficher des résultats expérimentaux et prometteurs. Trois types d'algorithmes évolutionnaires ont été développés isolément et à peu près simultanément, par différents scientifiques : La programmation évolutionnaire [34], les stratégies d'évolution [35] et les algorithmes génétiques [36].

II.3 Les algorithmes génétiques

Les algorithmes génétiques, forment une des principales classes des algorithmes évolutionnaires ; ils ont suscité beaucoup d'enthousiasme depuis plusieurs années. Leur efficacité pour produire des solutions de qualité dans un grand nombre de problème d'optimisation est maintenant bien établie. Ils représentent les méthodes d'optimisations les plus utilisées.

Cet algorithme a pour but de faire évoluer un ensemble de solution candidate à un problème posé vers la solution optimale. Cette évolution s'effectue sur la base de transformations inspirées de la génétique, assurant de génération en génération, l'exploration de l'espace des solutions en direction des plus adaptées.

En se reportant à la version canonique des AG [36], les individus sont des chaînes construites sur l'alphabet binaire $\{0, 1\}$ et sont de longueur fixe, formant une population de taille constante en cours du temps. Trois principaux opérateurs d'évolution sont appliqués à chaque génération :

- 1- sélection généralement basée sur la valeur d'adaptation des individus ;
 - 2- croisement qui consiste à combiner, de manière pseudo-aléatoire, les informations portées par deux individus parents pour former un individu enfant ;
 - 3- mutation qui consiste à altérer, de manière pseudo aléatoire, la structure d'un individu.
- Basé sur le principe de l'AG canonique, de nombreuses variantes d'AG ont été proposées de manière à résoudre efficacement des problèmes divers d'optimisation.

Les AG diffèrent fondamentalement des autres méthodes d'optimisation selon les principaux axes suivants [47] :

1. les algorithmes génétiques utilisent un codage des paramètres, et non les paramètres eux même ;
2. les algorithmes génétiques travaillent sur une population de points, au lieu d'un point unique ;
3. les algorithmes génétiques n'utilisent que les valeurs de la fonction étudiée, pas sa dérivée, ou une autre connaissance auxiliaire.
4. les algorithmes génétiques utilisent des règles de transitions probabilistes, et non déterministes.

II.3.1. Présentation générale des AG

La caractérisation du processus d'évolution sous l'angle de la génétique, nécessite sans doute la transposition des concepts biologique dans un cadre artificiel.

Le principe général de fonctionnement d'un algorithme génétique est illustré dans la figure II.2. :

- Au début une population de N individus est générée de façon aléatoire.
- Evaluer chaque individu chromosome de la population.
- Créer la nouvelle population, cette dernière évolue en générations successives. Les individus les plus forts survivent et se reproduisent entre eux pour créer de nouveaux individus, tandis que les plus faibles disparaissent petit à petit. Cela conduit à définir les trois opérateurs génétiques de base que sont la sélection, croisement (crossover) et la mutation.
- On réitère les opérations précédentes jusqu'à ce qu'un critère d'arrêt soit satisfait. Différents critères d'arrêt de l'algorithme génétique peuvent être choisis : nombre de générations maximum, limite de convergence de la population, population qui n'évolue plus suffisamment.

Dans ce qui suit on tente d'expliquer plus en détail les points suivants :

- La représentation de l'individu et population.
- L'évaluation d'un individu de population.
- L'évolution de l'individu.

II.3.1.1. Individu et population

Les systèmes génétiques artificiels puisent de la terminologie des systèmes biologiques. Ainsi un individu, étant l'unité fondamentale supportant le matériel génétique en biologie, il représente pour le système artificiel, la structure permettant d'encoder une solution candidate. Chaque individu ou chromosome exprimé par un génotype, est constitué d'un ensemble fixe de gène représentant chacune de ces caractéristiques. Le codage d'un individu produit son phénotype. Un gène identifié par sa position appelé (locus), peut prendre plusieurs valeurs dénommées allèle constituant ainsi l'alphabet de l'individu.

Initialement on adopte particulièrement la représentation binaire, ce qui correspond à l'alphabet minimal $\{0,1\}$, on parle alors de la version canonique des AGs. Par la suite d'autres représentations étendues ont été présentées.

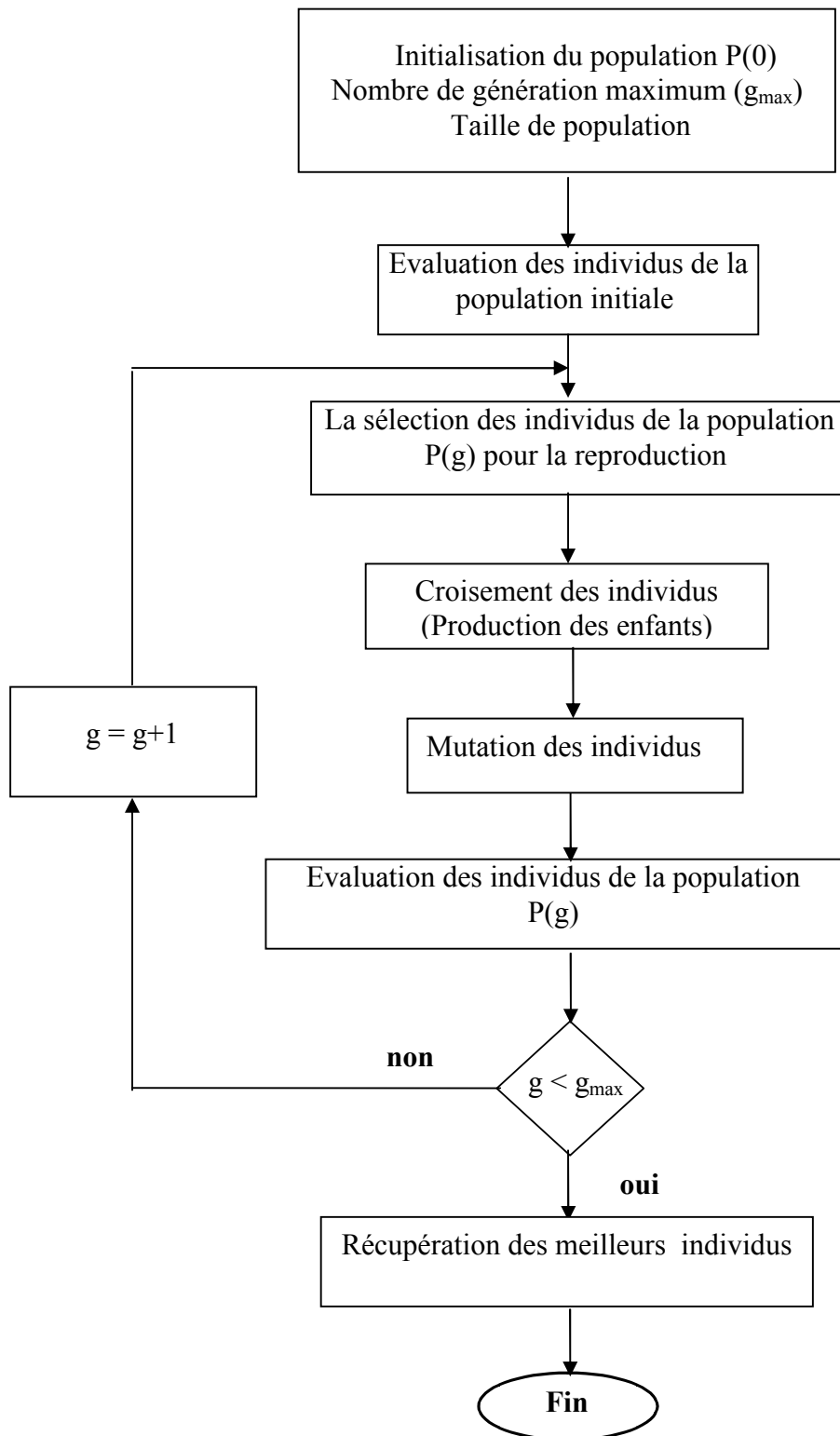


Figure II.2. Principe du fonctionnement d'un algorithme génétique

La population initiale est générée d'une manière aléatoire. Cependant, il est très recommandé de disposer d'un mécanisme permettant de générer une population non homogène qui servira de base pour les générations futures. Ce choix conditionne la rapidité de la convergence vers l'optimum. Cela dit, dans le cas où l'on ne connaît rien du problème à résoudre, il est essentiel que la population initiale soit répartie sur l'espace de recherche.

II.3.1.2. Evaluation des individus de la population

Les transformations qu'opère un algorithme génétique sur une population d'individu est régie par une mesure de leur adaptation ou capacité à résoudre le problème posé. Évaluer un individu ou un chromosome consiste à calculer sa force. Cette évaluation permettra aux individus les plus forts d'être retenus. Chaque individu a une valeur fitness retournée par l'application d'une fonction d'évaluation. Ainsi le but de l'AG est de maximiser la force des individus de la population. Soit $C(x)$ la valeur du critère à optimiser pour l'individu x . Si ce critère doit être maximisé, il peut servir directement de mesure de la force. Dans le cas où l'on désire minimiser ce critère il est nécessaire de le compléter pour le ramener à un cas de maximisation. Goldberg propose de calculer la force $F(x)$ d'un individu x de la façon suivante :

$$F(x) = \begin{cases} C_{\max} - C(x) & \text{Si } C(x) > 0 \\ 0 & \end{cases} \quad (2.1)$$

Où C_{\max} peut être un coefficient fixé ou la plus grande valeur observée de $C(x)$ soit dans une population, soit depuis le début de l'algorithme.

Aucune condition n'est requise pour la fonction objectif. Il suffit simplement que cette fonction retourne des valeurs numériques comparables. La performance de l'AG peut être sensible aux choix de cette fonction. La fonction d'évaluation agit en deux temps [37]:

1. Décodage de l'individu, c'est à dire interprétation de la chaîne de bits (dans le cas d'un codage binaire).
2. Calcul de la valeur de ce phénotype comme une solution au problème, fournissant la performance de l'individu ou capacité d'adaptation.

II.3.1.3 Opérateurs Génétiques

Les opérateurs génétiques représentent des procédures de transformation des individus entre deux générations. Les algorithmes génétiques exploitent principalement trois opérateurs visant chacun d'eux un objectif spécifique relativement à la couverture de l'espace des solutions. Ces opérateurs sont la sélection, le croisement et la mutation.

II.3.1.3.1. Sélection

La sélection est le premier opérateur génétique appliqué à une population d'individu en vue de la renouveler. Cet opérateur base la constitution de la nouvelle population sur le fitness des individus de la population qui la précède. Cet opérateur est peut être le plus important puisqu'il permet aux individus d'une population de survivre, de se reproduire ou de mourir. En règle générale, la probabilité de survie d'un individu sera directement reliée à son efficacité relative au sein de la population.

Il existe plusieurs méthodes de sélections. On peut citer la sélection proportionnelle basée sur le principe de la roulette [47], par rang de classement dans la population [33], par tournoi et par élitisme.

Sélection de la roulette (Roulette Wheel sélection) :

Appelé aussi la roue de loterie biaisée de Goldberg (1989) c'est la méthode la plus connue et utilisée. Selon cette méthode, chaque chromosome sera dupliqué dans une nouvelle population proportionnellement à sa valeur d'adaptation. On affecte à chaque individu X_i une probabilité $p(X_i)$ avec laquelle il sera introduit dans la nouvelle population de taille N est :

$$p(X_i) = \frac{f(X_i)}{\sum_{j=1}^N f(X_j)} \quad (2.2)$$

Plus la performance d'un individu est élevée, par rapport à celle des autres, plus il a une chance d'être reproduit dans la population. Les individus ayant une grande fitness relative, ont donc plus de chance d'être sélectionnés. On parle alors de la sélection proportionnelle.

Le nombre de copies espérées pour chaque individu X_i , qui va résulter de la sélection est alors égale à :

$$n = N \times p \quad (X_i) = \frac{f(X_i)}{\frac{1}{N} \sum_{j=1}^N f(x_j)} \quad (2.3)$$

L'inconvénient majeur de ce type de reproduction vient du fait qu'un individu n'étant pas le meilleur peut tout de même dominer la sélection. Elle peut aussi engendrer une perte de diversité par la domination d'un super individu. Pour palier à cet inconvénient on préfère des méthodes qui n'autorisent en aucun cas l'apparition de super individu. Par exemple la sélection par tournoi (tournament selection): On tire deux individus aléatoirement dans la population et on reproduit le meilleur des deux dans la nouvelle population. On refait cette procédure jusqu'à ce que la nouvelle population soit complète. Cette méthode donne de bons résultats.

La sélection par élitisme :

A la création d'une nouvelle population, il ya de grandes chances que les meilleurs chromosomes soient perdus après les opérations de croisement et de mutation. Pour éviter cela, on utilise la méthode d'élitisme. Elle consiste à copier un ou plusieurs des meilleurs chromosomes dans la nouvelle génération. En suite on génère le reste de la population selon l'algorithme de reproduction usuel. Cette méthode améliore considérablement les algorithmes génétiques, car elle permet de ne pas perdre les meilleures solutions.

La sélection par rang de classement :

Dans ce schéma de sélection, les individus d'une population sont classés dans une liste selon l'ordre croissant de leur évaluation, ensuite la sélection est proportionnelle à leur rang dans la liste de la population [33]. Cette méthode est utilisée pour une fonction d'évaluation dont les valeurs sont très proches. Dans cette procédure, les meilleures ont toujours plus de chance d'être choisies, mais moins souvent que la roulette et les moins bons individus auront plus de chance de participer à la reproduction.

Toute fois, aussi important que soit la phase de sélection, elle ne crée pas de nouveaux individus dans la population. Ceci est le rôle des opérateurs de croisement et de mutation.

II.3.1.3.2. Le croisement

Le croisement est le deuxième opérateur génétique appliqué à la population d'individus enfant issue de la sélection. Cet opérateur consiste par analogie aux systèmes biologiques à effectuer un échange d'information entre les chromosomes (individus) par le biais de leur recombinaison avec une probabilité P_c . C'est un opérateur de combinaison qui agit

généralement par paire d'individus en déterminant un ou plusieurs points de coupure, délimitant la frontière des parties à échanger. Lors de cette opération la difficulté réside dans le fait que la bonne partie du chromosome n'est pas connue et qu'il est alors possible de créer des enfants de plus mauvaise qualité que celle des parents initiaux [38].

Ils existent de nombreuses techniques de croisement. On distingue principalement le croisement à points et le croisement en deux points.

II.3.1.3.2.a. Le croisement à un point

Deux individus qui forment alors un couple, sont tirés alors aléatoirement. Les individus sont coupés en un ou plusieurs points aléatoires dits site(s) de croisement. Les segments situés à partir de l'extrémité (cas d'un croisement à un point) ou entre les points (cas d'un croisement à plusieurs points), sont échangés entre eux.

Nous allons présenter un exemple pour illustrer cet opérateur :

Considérons les deux parents :

P1 : 1 1 0 1 0 1 1 0 0 1 0

P2 : 0 1 1 1 1 0 1 0 1 1 1

D'abord nous choisissons aléatoirement un point de coupure, par exemple, la quatrième position dans les deux parents :



P1 : 1 1 0 1 | 1 1 0 0 1 0

P2 : 0 1 1 1 | 1 0 1 0 1 1

Ensuite nous construisons les enfants (nouveaux individus) comme suit :

E1 : 0 1 1 1 | 1 1 0 0 1 0

E2 : 1 1 0 1 | 1 0 1 0 1 1

Donc les deux enfants deviennent finalement

E1 : 0 1 1 1 1 1 0 0 1 0

E2 : 1 1 0 1 1 0 1 0 1 1

II.3.1.3.2.b Le croisement en deux points

On peut étendre ce principe de croisement en choisissant non pas un seul point de croisement mais 2,3,...etc (croisement en multipoints) [39]. L'exemple suivant illustre ce type de croisement :

Considérons les deux parents :

P1 : 1 1 0 1 0 1 1 0 0 1 0

P2 : 0 1 1 1 1 0 1 0 1 1 1

Un croisement localisé entre la deuxième et la quatrième position aura lieu entre les chromosomes parents P1 et P2 :

\downarrow \downarrow
P1 : 1 | 1 0 1 | 0 1 1 0 0 1 0
P2 : 0 | 1 1 1 | 1 0 1 0 1 1 1

Les deux nouveaux chromosomes enfants deviennent :

E1 : 1 1 1 1 0 1 1 0 0 1 0

E2 : 0 1 0 1 1 0 1 0 1 1 1

Ce type de croisement est très efficace et peut s'étendre à n'importe quel type de chaîne (binaire, réelle,...etc.). Néanmoins certains auteurs [40] préfèrent utiliser dans le cas des chaînes réelles.

Cet opérateur permet la création de deux individus. Toutefois, un individu sélectionné lors de la reproduction ne subit pas nécessairement l'action d'un croisement. Ce dernier ne s'effectue qu'avec une certaine probabilité. Plus cette probabilité est élevée et plus la population subira de changement.

Quoi qu'il en soit, il se peut que l'action conjointe de la reproduction et du croisement soit insuffisante pour assurer la réussite de l'algorithme génétique. Ainsi, dans le cas du codage binaire, certaines chaînes peuvent totalement disparaître de la population. C'est pour remédier entre autre à ce problème que l'opérateur de mutation est utilisé [41], [42].

II.3.1.3.3. La mutation

La mutation s'applique à un individu issu de la population ayant subit le croisement. Cet opérateur consiste à modifier aléatoirement la valeur d'un gène dans un chromosome selon une probabilité P_m généralement inférieure à la probabilité de croisement P_c [43]. Dans le cas du codage binaire, le bit $a_i \in \{0,1\}$ choisit aléatoirement est remplacé par son complémentaire. Dans l'exemple de la figure II.3 une mutation sur le 3^{ème} gène du chromosome P et a transformer ce gène de la valeur 0 à 1.

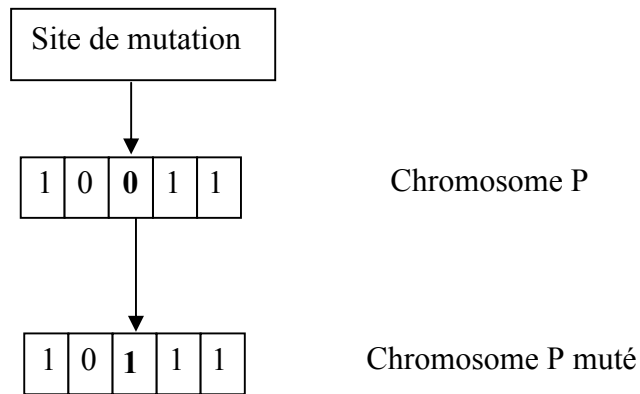


Figure. II.3 Mutation

Dans le cas d'un codage réel, on utilise principalement deux opérateurs de mutation : la mutation uniforme et la mutation non uniforme [44]. En supposant fixée la probabilité de mutation P_m , un tirage au sort pour chaque gène x_k du chromosome P permet de décider si ce gène doit être ou non modifier. Nous supposant que le gène prend des valeurs dans un intervalle $[x_k^{\min} \quad x_k^{\max}]$.

Pour la mutation uniforme, qui est une simple extension de la mutation binaire, on remplace le gène x_k sélectionné par une valeur quelconque x'_k tiré aléatoirement dans l'intervalle $[x_k^{\min} \quad x_k^{\max}]$.

Pour la mutation non uniforme, le calcul de la nouvelle valeur est un peu plus complexe. Le gène x_k subit des modifications importantes durant les premières générations, puis graduellement décroissantes au fur et à mesure que l'on progresse dans le processus d'optimisation [45].

L'opérateur de mutation est généralement considéré comme un opérateur intervenant à la marge dans la mesure où sa probabilité est en général assez faible (de l'ordre de 1%). Mais elle confère aux algorithmes génétiques une propriété très importante : c'est-à-dire que tous les points de l'espace de recherche peuvent être atteints. Cet opérateur est donc d'une grande importance et il est loin d'être marginal [46].

II.3.1.4. Le codage

Il faut représenter les différents états possibles de la variable dont on cherche la valeur optimale sous forme utilisable pour un AG : c'est le codage. Cela permet d'établir une connexion entre la valeur de la variable et les individus de la population, de manière à imiter la transcription genotype-phenotype qui existe dans le monde vivant. Les principaux types de codage : le codage binaire, le codage réel.

II.3.1.4.1. Le codage binaire

Ce codage a été le premier à être utilisé dans le domaine des AG. Ce codage confère aux algorithmes génétiques une très grande robustesse car il est indépendant du domaine de problème traité et les opérateurs standards peuvent être systématiquement utilisés. Il présente un alphabet minimum $\{0,1\}$ et existence de fondements théorique (théorie sur les schémas). Cependant, le codage par chaîne de bits présente des inconvénients :

- Deux individus séparés par une faible distance de Hamming [47] peuvent être très éloignés dans l'espace de recherche.
- Les performances de l'AG sont dégradées devant les problèmes d'optimisation de grande dimension à haute précision numérique. Pour de tels problèmes, les AG basés sur les chaînes binaires ont de faibles performances [44].

II.3.1.4.2. Le codage réel

Une des améliorations majeures consiste à se servir de nombres réels directement. Les résultats donnés par Michalewicz [44] montrent que la représentation binaire aboutit souvent à une moins bonne précision. Le codage réel a le mérite d'être simple. Chaque chromosome est en fait un vecteur dont les composantes sont les paramètres du processus d'optimisation. Par exemple, si on cherche l'optimum d'une fonction de n variables $f(x_1, x_2, \dots, x_{n-1}, x_n)$, on peut utiliser tout simplement un chromosome P contenant les n variables : Avec ce type de codage, la procédure d'évaluation des chromosomes est plus rapide vu l'absence de l'étape de transcoding (du binaire vers le réel). Les résultats donnés par Michalewicz (1992) montrent que la représentation réelle aboutit souvent à une meilleure précision et un gain important en termes de temps d'exécution.

II.4. Les critères de convergence

Une des principales difficultés dans l'application d'un AG est en fait de savoir comment l'arrêter. Il est bon de disposer d'un mécanisme automatique coupant la boucle itérative. Notons que la recherche directe possède le remarquable privilège de ne jamais diverger car une bonne solution découverte est en principe préservée jusqu'à la fin, on ne se préoccupe pas de dépister une évolution divergente.

II.4.1. Nombre maximal de générations

Avec cette option, on fixe à l'avance un nombre de génération à atteindre. A utiliser quand des essais préalables ont déjà été effectués et que le taux de convergence est relativement stable.

II.4.2. Critère sur l'évolution du meilleur individu

Le suivi du meilleur individu P_i de la population et de son évolution fournit aussi un critère d'arrêt. Si, pendant n_g générations consécutives jusqu'à la génération courante g , le déplacement de P_i dans l'espace \mathbb{R}^p où la variation de $f(P_i)$ ne dépasse pas un certain seuil donné, alors l'algorithme doit se terminer :

$$\text{Si } \|P_i^K - P_i^{K-1}\| \leq \varepsilon_p \text{ ou } |f(P_i^K) - f(P_i^{K-1})| \leq \varepsilon_f, K = (g - n_g + 1) \text{ alors fin} \quad (2.4)$$

En particulier, si $\varepsilon_f = 0$, on décide de s'arrêter si aucune amélioration sur la meilleure solution trouvée n'est constatée pendant n_g générations, quelque soit sa localisation dans \mathfrak{R}^p .

II.5. Avantages et inconvénients des algorithmes génétiques

II.5.1. Les avantages des AGs

Les AG opèrent au niveau du codage des paramètres sans se soucier de leur nature, donc ils s'appliquent à de nombreuses classes des problèmes, qui dépendent éventuellement de plusieurs paramètres de natures différentes (booléens, entiers, réels, fonctions...etc) ;

Pour les mêmes raisons un AG est dans l'idéal totalement indépendant de la nature du problème et de la fonction à optimiser, car il ne se sert que des valeurs d'adaptation, qui peuvent être très différentes des valeurs de la fonction à optimiser, même si elles sont calculées à partir de cette dernière.

Potentiellement les AGs explorent tous les espaces des points en même temps, ce qui limite les risques de tomber dans les optimums locaux ;

-Les AGs ne se servent que des valeurs de la fonctionnel pour optimiser cette dernière, il n'a pas besoin d'effectuer de coûteux et parfois très complexes calculs;

-Les AGs présentent une grande robustesse c'est-à-dire une grande capacité à trouver les optimums globaux des problèmes d'optimisation.

II.5.2. Les inconvénients des AGs

Les AG ne sont encore actuellement pas très efficaces en cout (ou vitesse de convergence), vis-à-vis de méthodes d'optimisation plus classiques; Parfois les AG convergent très vite vers un individu particulier de la population dont la valeur d'adaptation est très élevée.

Le respect de la contrainte de domaine par la solution codée sous forme de chaine de bits pose parfois problème. Il faut bien choisir le codage, voir modifier les opérateurs. L'utilisation d'un AG ne garantie pas le succès de l'optimisation;

En pratique l'efficacité d'un AG dépend souvent de la nature du problème d'optimisation. Selon les cas de choix des opérateurs et des paramètres seront souvent critiques, mais aucune théorie générale ne permet de connaître avec certitude la bonne paramétrisation, il faudra faire plusieurs pour s'en approcher.

II.6. Algorithmes génétique proposés :

Dans ce qui suit en vue d'augmenter la convergence et la précision des AG on propose deux variantes d'AG

II.6.1- Un algorithme génétique dont la fonction objectif est à coefficients de pondération variable

Le choix de la fonction de coût est le point clé de la procédure d'optimisation par l'algorithme génétique puisque cette fonction est l'indicateur de pertinence du vecteur solution potentielle, qui pourrait être calculé en utilisant n'importe quel connaissance du système.

Le critère ou la fonction coût consiste à combiner de façon linéaire, plusieurs critères f_i du problème en un critère unique J :

$$J(x) = \sum_{i=1}^N c_i f_i(x) \tag{2.5}$$

Où les paramètres $c_i \geq 0$ représentent les poids affectés aux critères, N désigne le nombre de critère.

Dans plusieurs études et applications les poids de pondération de la fonction objectif sont considérés constants dans la procédure d'optimisation en utilisant l'algorithme génétique. La même solution peut être obtenue avec des poids de pondération différents.

Pour améliorer la performance globale et la vitesse de convergence de la procédure de l'algorithme génétique dans le domaine d'identification paramétrique et dans le but de réduire l'effort de calcul nécessaire pour évaluer la fonction de coût dans l'espace de recherche, on propose que Les valeurs des poids de pondération ne seront pas avec des valeurs constantes le long de la procédure d'identification avec la technique d'algorithme génétique mais ils vont être considérée comme des paramètres à optimisés au cours de la procédure d'optimisation. Les chromosomes renferment non seulement les paramètres du problème (système) à optimisés, mais également les des poids de pondération de la fonction de coût (Figure II.4).

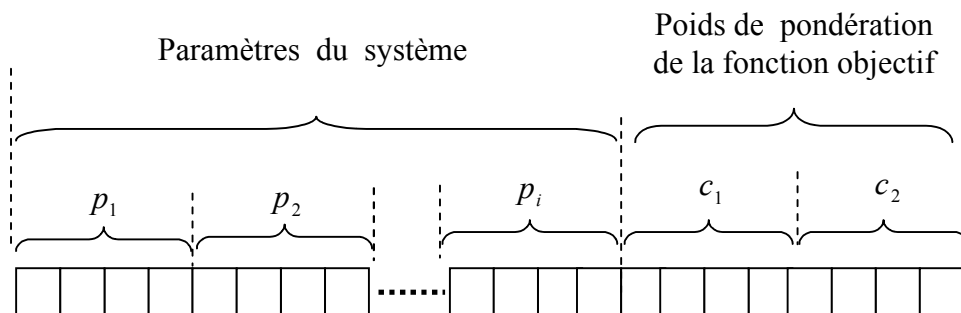


Figure. II.4 chromosome modifié (cas d'une fonction objectif à deux paramètres).

II.6.2 Un algorithme génétique à codage réel modifié

L'algorithme génétique avec codage binaire ne permet pas d'avoir une précision arbitraire de la solution obtenue. La longueur du chromosome doit être fixée initialement pour permettre une certaine précision dans la solution. Plus en demande de précision, plus la longueur de chromosome augmente et plus la taille de population devient importante. Ainsi pour un problème d'optimisation avec plusieurs variables, la longueur de chromosome et la

taille de population tendent à augmenter. Ce qui ralentit la vitesse de calcul et la convergence devient très lente à savoir impossible. Une autre difficulté qui va être rencontrée est l'effet de Hamming-Cliff associé à certains chromosomes ayant des phénotypes adjacents mais complètement différents. Par exemple, pour les phénotypes $x_p = 7$ et $y_p = 8$, la distance entre eux est égale à 1, mais leurs chromosomes $x_c = 0111$ et $y_c = 1000$ possèdent la plus grande distance $|x_c - y_c| = 4$. Par conséquent, le codage binaire ne permet pas une optimisation locale parfaite.

Le codage réel évite ces inconvénients et rend la présentation de la solution proche à la formulation du problème. Cependant, la difficulté apparaît avec les opérateurs d'évolution (les opérateurs de croisement et de mutation). Avec le codage binaire, il est facile d'implémenter l'échange de fragments de bits des deux chromosomes parents. Un simple changement de bit pour réaliser une mutation est aussi convenable et ressemble à la mutation naturelle. Dans l'algorithme génétique à codage réel, le problème principal est comment utiliser la paire de vecteurs de paramètres réels pour créer une nouvelle paire de vecteurs enfants, ou bien comment perturber un vecteur de paramètres pour avoir un autre muté d'une manière significative.

Pour combiner les avantages des deux types de codage on propose le codage réel-codé-décimal.

II.6.2.1 Codage réel-codé-décimal :

Comme son nom l'indique, la valeur réelle est représentée par la même suite de chiffres décimaux utilisés dans le système de numération décimale. Ces chiffres sont juxtaposés pour former le chromosome donc les allèles sont des chiffres compris entre 0 et 9.

En arithmétique, les valeurs réelles possèdent deux représentations : la représentation en virgule fixe et la représentation en virgule flottante. La structure du chromosome doit être fixée initialement selon la représentation choisie.

Pour la représentation en virgule fixe, on doit déterminer le nombre de chiffres dans la partie entière et celui de la partie fractionnaire. Le point décimal n'est pas codé dans le chromosome.

Exemple 1:

Soit à coder les deux nombres réel suivants : $N_1=5467,314$ et $N_2=618,5894$

On propose par exemple 4 chiffres pour la partie entière 4 chiffres pour la partie fractionnaire.

Les chromosomes associés à ces deux nombres sont représentés dans la figure II.5.a

Dans la représentation en virgule flottante, un nombre réel s'écrit sous la forme : $M 10^E$ où M est la mantisse et E est l'exposant.

On a donc deux entités à représenter dans le chromosome : la mantisse et l'exposant.

Pour la mantisse on réserve un seul chiffre pour sa partie entière et on détermine le nombre de chiffre de la partie fractionnaire. L'exposant évidemment est entier qu'on fixe tout simplement sa longueur.

Exemple 2:

Soit à coder le nombre $N= 5,625 10^5$

Par exemple on a choisi 5 chiffres pour la partie fractionnaire et deux chiffres pour l'exposant.

Le chromosome associé à ce nombre est représenté dans la figure II.5.b

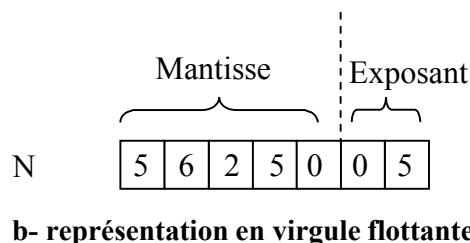
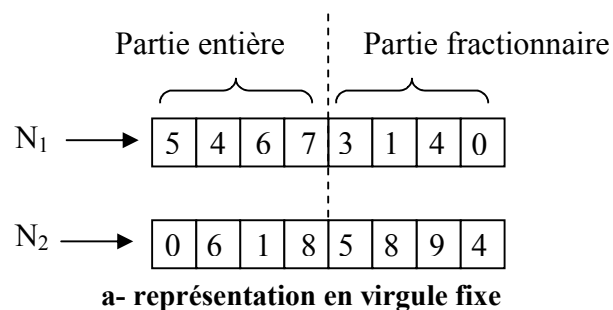


Figure II.5 Exemple d'un codage réel codé décimal

II.7. Conclusion

Dans ce chapitre nous avons donné une présentation sommaire des principes de base nécessaires à la compréhension du fonctionnement des algorithmes génétiques. Nous retenons fondamentalement qu'un AG est une reproduction artificielle de mécanismes naturels liés à la génétique. Basé sur l'application de trois opérateurs que sont la sélection, croisement et mutation, l'AG est un processus cyclique manipulant une population de solutions candidates à un problème dans le but d'optimiser la fonction d'adaptation associée, définie dans un espace éventuellement complexe. A la lumière de cette présentation des AG, ces derniers semblent être une solution intéressante pour approcher le problème d'identification paramétrique d'un système non linéaire.

En vu d'amélioration des performances de l'AG à savoir la convergence et la précision des solutions obtenus, nous avons proposé deux modifications l'une basée sur l'utilisation d'une fonction objectif avec des poids de pondération qui vont être aussi ajouté aux chromosomes individus de la population l'autre à utiliser un codage réel modifié.

CHAPITRE 3

Les techniques Soft computing : Essaim de particules

III.1.Introduction

Dans ce chapitre on va présenter un algorithme basée sur un essaim de particule qui utilise une population de solutions potentielles (particules), qu'elle fait évoluer à l'aide des échanges d'informations (essaim) entre particules. En fait, chaque particule ajuste sa trajectoire vers sa meilleure position dans le passé et vers la meilleure position des particules de son voisinage [59]. La variante globale de ces algorithmes considère la totalité de l'essaim comme voisinage. Les particules profitent ainsi des découvertes et expériences antérieures de toutes les autres particules.

III.2.Optimisation par essaim particulaire

L'optimisation par essaim particulaire (OEP) a été proposée par Kennedy et Eberhart en 1995 [51]. Cette méthode est inspirée du comportement social des animaux évoluant en essaim. L'exemple le plus souvent utilisé est le comportement des bancs de poissons [52], [53]. En effet, on peut observer chez ces animaux des dynamiques de déplacement relativement complexes, alors qu'individuellement chaque individu a une intelligence limitée et une connaissance seulement locale dans sa situation dans l'essaim comme la position et la vitesse de leurs plus proches voisins. On peut par exemple observer qu'un banc de poissons est capable d'éviter un prédateur : d'abord en se divisant en deux groupes puis en reformant le banc original, tout en maintenant la cohésion du banc.

Pour résumé, un individu de l'essaim n'a pour connaissance que la position et la vitesse de ses plus proches voisins. Chaque individu utilise donc, non seulement sa propre mémoire, mais aussi l'information locale sur ses plus proches voisins pour décider de son propre déplacement. Des règles simples, telles que "aller à la même vitesse que les autres", "se déplacer dans la même direction" ou encore "rester proche de ses voisins" sont des exemples de comportement qui suffisent à maintenir la cohésion de l'essaim, et qui permettent la mise en œuvre de comportements collectifs complexes et adaptatifs.

L'intelligence globale de l'essaim est donc la conséquence directe des interactions locales entre les différentes particules de l'essaim. La performance du système entier est supérieure à la somme des performances de ses parties.

Les chercheurs Kennedy et Eberhart se sont inspirés de ces comportements socio-psychologiques pour créer l'optimisation de l'essaim de particule OEP. Un essaim de particules, qui sont des solutions potentielles aux problèmes d'optimisation, "survole" l'espace de recherche, en quête de l'optimum globale.

L'OEP présente quelques similarités avec les algorithmes génétiques en ce sens qu'elle fonctionne sur la base d'une population de solutions qui interagissent entre elles afin de trouver la meilleure solution possible à un problème d'optimisation. Cependant contrairement aux AG où l'évolution est basée sur la compétition et l'élimination des individus les moins performants, l'OEP se base sur la coopération entre les individus afin de faire évoluer chacune. Aucune solution n'est éliminée, chacune utilisant les connaissances que ses voisines possèdent du milieu afin de se déplacer efficacement et de s'améliorer.

III.3. Fonctionnement général de l'OEP

L'OEP repose sur un ensemble de solutions disposées aléatoirement au début de l'algorithme. Chaque solution est appelée particule et se situe à une position donnée dans l'espace de recherche. Chaque particule dispose d'une mémoire dans laquelle est conservée sa solution. L'OEP fonctionne aussi de manière itérative. À chaque itération de l'algorithme, les particules se déplacent dans l'espace de recherche selon une certaine vitesse. Le calcul de cette vitesse dépend de plusieurs facteurs telle que la vitesse actuelle pondérée par un coefficient d'inertie w , ainsi que l'écart par rapport à leur meilleure position connue et à celle de leurs voisines pondérées par des coefficients de confiances c_1 et c_2 . Les particules voisines d'une autre particule sont appelées informatrices.

La structure de base d'un algorithme d'OEP est comme dans le cas des algorithmes génétiques, une population initiale est d'abord générée et évaluée. Les principes de génération de cette population et d'évaluation des solutions demeurent les mêmes pour les deux méthodes. A chaque itération de l'algorithme, la nouvelle vitesse et la nouvelle position de chaque particule sont calculées. Les particules sont ensuite évaluées et leurs mémoires, leurs meilleures positions connues, sont mises à jour. La diversité au sein de la population permet à l'algorithme de ne pas rester bloqué dans un optimum local.

Au fil des itérations, la population doit normalement converger vers l'optimum global de la fonction objectif.

III.4. Principe de l'évolution de l'essaim de particules

A la première itération, les positions et vitesses de chaque particule P_i sont initialisées aléatoirement. Puis à chaque itération du processus, trois éléments essentiels sont pris en compte pour la détermination de la nouvelle position de la particule P_i (Figure III.1):

- La vitesse propre de la particule P_i , calculée à l'itération précédente.
- La meilleure position dans le passé de la particule P_i , cette influence étant interprétée comme une attraction exercée par la meilleure position mémorisée de la particule.
- Pour la particule P_i , on tire aléatoirement sur l'ensemble des particules une liste de particule informatrice pour le calcul de la nouvelle position P_i . Cette influence étant interprétée comme une attraction exercée par la meilleure des particules informatrices.

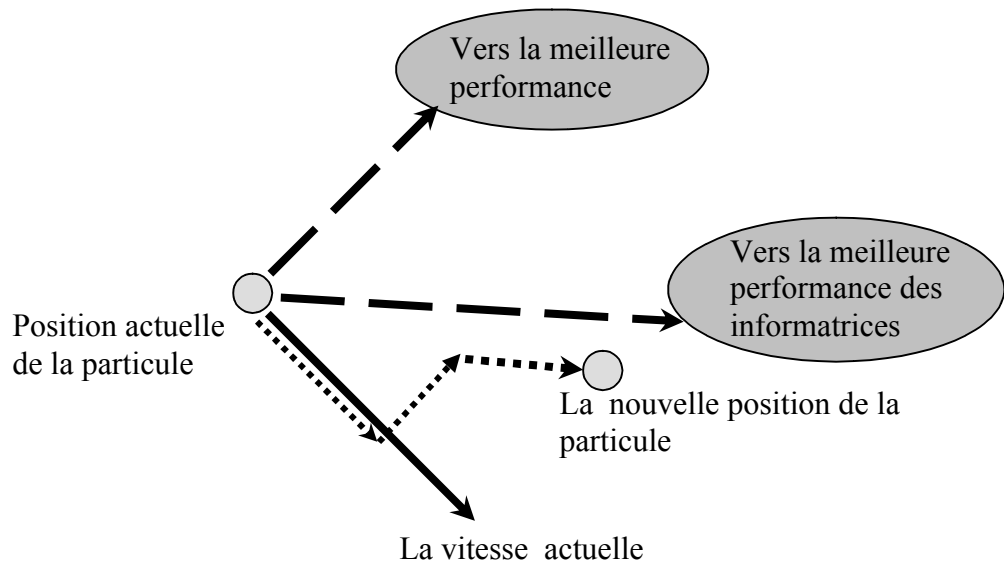


Figure III.1 Principe de déplacement de la particule

Dans le cas d'un problème d'optimisation, la qualité d'un site de l'espace de recherche est déterminée par la valeur de la fonction objectif en ce point.

Dans un espace de recherche, la particule i de l'essaim est modélisée par son vecteur position $\vec{X}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ et par son vecteur vitesse $\vec{V}_i = (v_{i1}, v_{i2}, \dots, v_{iD})$. Cette particule garde en

mémoire la meilleure position ($pbest$) par laquelle elle est déjà passée, que l'on note $p_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ la meilleure position ($gbest$) atteinte par toute les particules de l'essaim est notée $g = (g_{i1}, g_{i2}, \dots, g_{iD})$. Au temps t le vecteur vitesse est calculé à partir de l'équation :

$$v_{ij}(t) = w v_{ij}(t-1) + c_1 r_1 (p_{ij}(t-1) - x_{ij}(t-1)) + c_2 r_2 (g_j(t-1) - x_{ij}(t-1)), j \in \{1, \dots, D\} \quad (3.1)$$

Où w est en général une constante appelée coefficient d'inertie, c_1 et c_2 sont des constantes, appelées coefficients d'accélération, r_1 et r_2 sont deux nombres aléatoires tirés uniformément dans $[0,1]$ à chaque itération et pour chaque dimension.

$w v_{ij}(t-1)$ correspond à la composante physique de déplacement. Le paramètre w contrôle l'influence de la direction de déplacement sur le déplacement futur. Il est à noter que, dans certaines applications, le paramètre w peut être variable [54].

$c_1 r_1 (p_{ij}(t-1) - x_{ij}(t-1))$ correspond à la composante cognitive du déplacement. c_1 contrôle le comportement cognitif de la particule.

$c_2 r_2 (g_j(t-1) - x_{ij}(t-1))$ correspond à la composante sociale du déplacement. c_2 contrôle l'aptitude sociale de la particule.

La combinaison des paramètres w, c_1 et c_2 permet de régler la balance entre les phases diversification et intensification du processus de recherche [55], [56].

Il est à noter que le terme vitesse est ici abusif car les vecteurs \vec{V}_i ne sont pas homogènes à une vitesse. Il serait plus approprié de parler de "direction de déplacement". Cependant pour respecter l'analogie avec le monde animalier, les chercheurs ont préféré utiliser le terme "vitesse" [57].

La position au temps t de la particule i est alors définie par :

$$x_{ij}(t) = x_{ij}(t-1) + v_{ij}(t), \quad j \in \{1, \dots, D\} \quad (3.2)$$

L'organigramme de la technique d'OEP qui illustre le principe de fonctionnement est montré dans la figure III.2

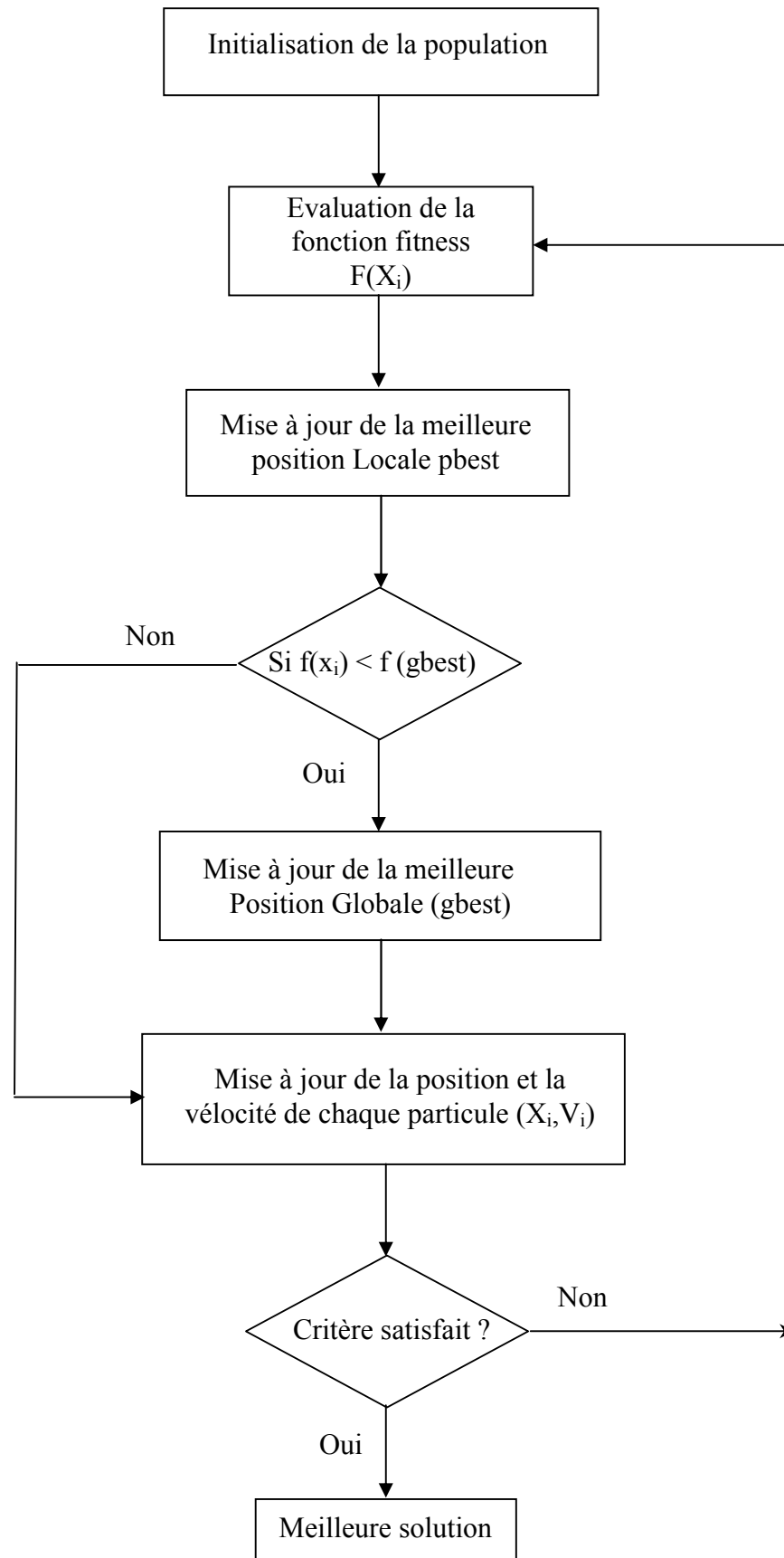


Figure III.2 L'organigramme de l'algorithme OEP

III.5. Configuration de la méthode

La configuration de l'OEP nécessite le réglage des paramètres suivant :

- Le nombre de particules de l'essaim ;
- La vitesse maximale d'une particule, notée V_{\max} ;
- La topologie et la taille du voisinage d'une particule qui définit son réseau social
- Les coefficients de confiance, notés ρ_1 et ρ_2 , qui pondèrent le comportement conservateur (la tendance à retourner vers la meilleure solution visitée) et le panurgisme (la tendance à suivre le voisinage).

Ces paramètres ont un impact direct sur les performances de l'algorithme. Ils sont présentés dans les sous paragraphes suivantes :

III.5.1. Le nombre de particules

Il n'existe pas de règle prédéfinie pour déterminer la taille de la population. La quantité de particules allouées à la résolution du problème dépend essentiellement de deux paramètres : La taille de l'espace de recherche et le rapport entre les capacités de calcul de la machine et le temps maximum de recherche. Il n'ya pas de règle pour déterminer ce paramètre, faire de nombreux essais permet de ce doter de l'expérience nécessaire à l'appréhension de ce paramètre.

III.5.2. La topologie et le choix du voisinage

La topologie du voisinage défini avec qui chacune des particules va pouvoir communiquer. Le voisinage d'une particule peut être global ou local.

Eberhart [58] ont défini une version dite locale de l'OEP, qui utilise un graphe (topologie) d'information circulaire. Les particules de l'essaim sont virtuellement disposer en cercle et numérotées séquentiellement à partir de 1 en parcourant le cercle. La particule n'est donc plus informée par toutes les particules de l'essaim, mais par elle même et ces deux voisins (Fig. III.3), c'est la topologie la plus utilisée. Bien qu'elle converge moins rapidement que la version globale [59].

Dans un voisinage global, chaque particule est reliée à toutes les autres particules.

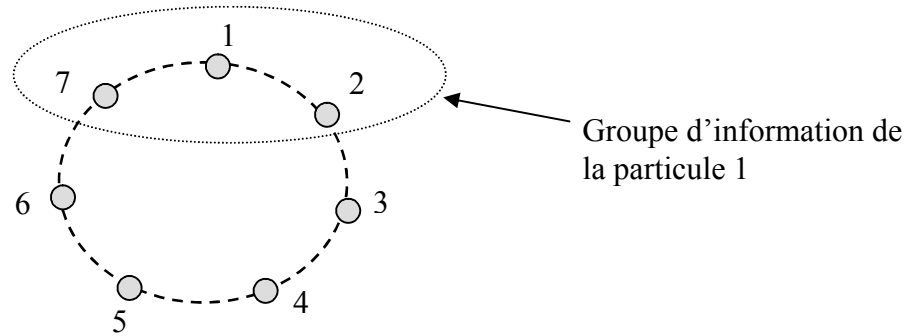


Figure III.3 Un graphe (topologie) d'information circulaire d'un essaim de 7 particules.

Les techniques d'OEP utilisant différentes topologies sont disponibles dans [59]. L'information importante est qu'aucune topologie n'est 'meilleure' que les autres, dans le sens où il n'y a aucune topologie qui donne de meilleurs résultats que toutes les autres.

III.5.3. Le facteur d'inertie

Le facteur d'inertie w représente la confiance que la particule a en sa vitesse actuelle. Il a été introduit par Shi et Eberhart [55], Il permet de définir la capacité d'exploration de chaque particule en vue d'améliorer la convergence de la méthode. Une grande valeur de w (>1) est synonyme d'une grande amplitude de mouvement et donc d'exploration globale de l'espace de recherche.

Au contraire, une faible valeur de w (<1) est synonyme de faible amplitude de mouvement et donc d'exploitation locale. Fixer ce facteur revient donc à trouver un compromis entre l'exploitation et l'exploration de l'espace de recherche. Bien que Shi et Eberhart [55] conseillent une valeur entre 0,8 et 1,2. Au delà de 1,2 l'algorithme tend à avoir certaines difficultés à converger. On rencontre généralement dans la littérature des valeurs de w situant entre 0,5 et 1.

III.5.4. Les coefficients de confiances

Les coefficients de confiance sont des constantes qui pondèrent les tendances de la particule à vouloir suivre son instinct de conservation (c_1) et à aller vers sa meilleure position connue, ou son panurgisme (c_2) en se rapprochant plus de la meilleure position de ses informatrices. Ces valeurs sont des constantes positives, déterminées de façon empirique [71].

III.5.5. Coefficient de constriction

Pour éviter que les particules se déplacent trop vite et au hasard dans l'espace de recherche, un mécanisme de confinement d'intervalle est souvent mis en place. La pratique la plus courante est de définir une vitesse maximale V_{\max} à ne pas dépasser, afin d'améliorer la convergence de l'algorithme.

Cependant, on peut s'en passer si on utilise un coefficient de constriction χ , introduit par Maurice Clerc [60], ce qui permet d'améliorer la convergence de l'algorithme sans avoir besoin de confiner la vitesse des particules. Le calcul de ce paramètre se fait à partir de l'équation suivante [71]:

$$\chi = \frac{2}{|2 - \rho - \sqrt{\rho^2 - 4\rho}|}$$

Où $\rho = \rho_1 + \rho_2$ et $\rho > 4$

Les variables aléatoires peuvent être définies de la façon suivante :

$$\begin{cases} \rho_1 = r_1 c_1 \\ \rho_2 = r_2 c_2 \end{cases}$$

$$V_i(t) = \chi(V_i(t-1) + \rho_1(x_{pbest_i} - x_i(t-1)) + \rho_2(x_{vbest_i} - x_i(t-1))) \quad (3.3)$$

Dans [60], de nombreux tests sont menés pour déterminer les valeurs optimales de ρ_1 et de ρ_2 . Dans la majorité des cas, on utilise $\rho = 4,1$ et $\rho_1 = \rho_2$, se qui donne un coefficient multiplicatif χ approximativement égal à 0,7298.

La définition du paramètre ρ permet de contrôler le système, afin qu'il présente les caractéristiques suivantes :

- Le système converge vers un état d'équilibre.
- Plusieurs régions de l'espace de recherche sont explorées tout en évitant une convergence prématurée.

L'équation traduit le calcul de la vitesse V_i lorsque le coefficient de constriction est utilisé. On obtient alors une relation entre le facteur d'inertie w , qui devient équivalent au coefficient de constriction χ et demeure inférieur à 1.

III.5.6. Initialisation de l'essaim

La position des particules ainsi que leur vitesse initiale doivent être initialisés aléatoirement selon une loi uniforme sur $[0,1]$. Cependant, en ce qui concerne la position des particules, il est préférable d'utiliser un générateur de séquence qui est plus pertinent dans la disposition homogène des particules dans un espace de dimension n .

III.6. Critères d'arrêt

La convergence vers la solution optimale globale n'est pas garantie dans tous les cas même si les expériences dénotent la grande performance de la méthode. De ce fait, il est fortement conseillé de doter d'une issue de sortie en définissant un nombre maximum d'itération.

L'algorithme doit alors s'exécuter tant que l'un des critères de convergence suivant n'a pas été atteint :

-Nombre d'itération ou temps de calcul.

-Plus d'amélioration de la meilleure particule pendant un nombre minimal d'itérations fixé à priori.

-Le fitness de la solution est suffisant.

Ces critères ne sont pas exclusifs et peuvent donc être utilisés simultanément.

III. 7 Optimisation par Essaim Particule proposé

Le réglage de tous les paramètres de l'OEP est un processus qui s'avère long et difficile. En effet chaque paramètre ayant une forte influence sur le comportement de l'algorithme [56], il est important de trouver un jeu de paramètres bien adapté au problème posé. De ce fait, chaque problème nécessiterait en théorie une étude qui permettrait de dégager le jeu de paramètres optimal pour le traiter.

Pour améliorer la vitesse et la convergence de l'OEP pour atteindre la solution globale on propose de faire diminuer le facteur d'inertie au cours du temps. De bons résultats ont été trouvés pour une valeur décroissant linéairement de 0,9 à 0,4 [55].

Pour améliorer les performances du OEP On va mettre l'évolution des coefficients r_1 et r_2 de manière adaptative :

$$\begin{aligned}
 r_1 &= 1 - \frac{f(p_{best})}{f(p_i)} + rand \\
 r_2 &= 1 - \frac{f(g_{best})}{f(p_i)} + rand
 \end{aligned}
 \tag{3.4}$$

Où *rand* est un nombre aléatoire prend des valeurs entre 0 et 1.

Le coefficient r_1 a une influence sur le mouvement individuel (second terme de l'équation III.1) comme un facteur de pondération lorsque une particule se rapproche près (proximité) de sa meilleur position p_{best} donc le mouvement individuel de la particule devient lent. De même pour le coefficient r_2 qui a une influence sur le troisième terme de l'équation III.1. Également on peut l'interpréter par :

$$f(g_{best}) \leq f(p_{best}) < f(p_i) \tag{3.5}$$

$f(g_{best})$ est considérée comme la plus faible valeur de la fonction de coût de tous les particules de l'essaim ainsi lorsque les particules de l'essaim converge vers la valeur optimale $f(p_{best})$ et $f(p_i)$ vont avoir la même valeur de $f(g_{best})$.

III.8. Conclusion

Nous avons présenté dans ce chapitre une méthode de résolution des problèmes d'optimisation OEP appartenant à une famille de méthodes dites métaheuristiques. Les paramètres utilisés par cet algorithme ont été présentés avec leur influence sur la convergence de la methode. Pour améliorer la vitesse et la convergence de l'OEP pour atteindre la solution globale on a proposé de faire varier les coefficients r_1 et r_2 de manière adaptative.

Dans le chapitre suivant, l'adaptation de cette méthode pour résoudre les problèmes d'identification des paramètres des systèmes non linéaires va être appliquée.

CHAPITRE 4

Les techniques Soft Computing pour l'identification des systèmes non linéaires

VI.1.Introduction

Dans ce chapitre, les techniques présentées dans les deux chapitres précédents seront utilisées pour l'identification des paramètres d'un moteur à induction pris comme un exemple d'un système non linéaire. Nous intégrons ces algorithmes avec des améliorations à la mise en œuvre de notre approche d'identification des paramètres de cette machine tout en présentant une validation croisée des paramètres identifiés, pour démontrer l'efficacité des approches de l'identification paramétriques proposées dans ce travail.

IV-2. Identification des paramètres d'une machine asynchrone

Grâce à sa structure simple et robuste, la machine asynchrone constitue une partie essentielle du système d'entraînement industrielle moderne. Récemment plusieurs techniques ont été développées pour contrôler la machine asynchrone de la même manière que la machine à courant continu. Ces techniques de contrôle se basent sur l'exactitude des paramètres, ces derniers sont connus qu'ils varient selon les différentes conditions d'exploitation de la machine [61]. Par conséquent, la connaissance des paramètres précis de la machine asynchrone devient essentielle pour atteindre les performances dynamiques souhaitées sous différentes conditions de fonctionnement. Toute fois il est bien connu que la constante de temps rotorique est nécessaire pour le réglage du contrôleur et les variables électriques du rotor ne sont pas mesurables, ainsi que les paramètres du rotor varient et la négligence de leurs variations peut conduire à une mauvaise interprétation des mécanismes de contrôle. La précision du flux du rotor estimé est fortement influencée par la valeur de la résistance du rotor utilisé pour le contrôle. En raison de l'échauffement du rotor la résistance du rotor peut varier jusqu'à 100 % la récupération de cette information à l'aide d'un modèle de température ou un capteur de température n'est pas facile. Plus généralement, les

paramètres électromagnétiques ne sont pas connus avec précision. L'effet de peau dans l'enroulement du rotor et saturation du noyau de fer peut conduire à des complications encore plus importantes dans le processus de modélisation de la machine. La réactance de fuite du stator est également connue par sa variation avec le courant [62].

IV-2.1. Modélisation de la machine asynchrone

La modélisation de la machine asynchrone a fait l'objet de nombreux travaux et publications [61], [62]. Dans ce qui suit le but est de ne pas réaliser une analyse détaillée de toutes les hypothèses faites pour avoir le modèle de la machine dite idéalisée ou de développer la transformation de Park ... etc., mais d'utiliser un modèle permettant de décrire la dynamique de la machine asynchrone. On montre dans plusieurs références bibliographiques [62] [63] [64] [65] [66] qu'il suffit de cinq paramètres électriques (R_s , R_r , L_s , L_r , L_m) pour décrire la dynamique de la machine.

IV-2.1.2. Représentation d'état de la machine asynchrone alimentée en tension

Le choix des variables d'état par la mise en œuvre du modèle dédiée à la commande peut se faire en considérant soit :

- Les courants statoriques et les courants rotoriques
- Les flux statoriques et les flux rotoriques
- Les courants statoriques et les flux statoriques
- Les courants statoriques et les flux rotoriques

La mise sous forme d'équation d'état du modèle de la machine asynchrone permet la résolution d'équation en mettant en évidence ses entrées et ses sorties.

Le système peut se mettre sous la forme d'état suivante :

$$\begin{aligned} \dot{X} &= A X + B U \\ Y &= C X + D U \end{aligned}$$

Avec :

$$A = \begin{bmatrix} -\left(\frac{R_s}{\sigma L_s} + \frac{1-\sigma}{\sigma \tau_r}\right) & 0 & \frac{L_m}{\sigma L_s L_r \tau_r} & \frac{L_m \omega_r}{\sigma L_s L_r} \\ 0 & -\left(\frac{R_s}{\sigma L_s} + \frac{1-\sigma}{\sigma \tau_r}\right) & \frac{L_m \omega_r}{\sigma L_s L_r} & \frac{L_m}{\sigma L_s L_r \tau_r} \\ \frac{L_m}{\tau_r} & 0 & -\frac{1}{\tau_r} & -\omega_r \\ 0 & \frac{L_m}{\tau_r} & \omega_r & -\frac{1}{\tau_r} \end{bmatrix} \quad (4.1)$$

$$B = \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Où X est le vecteur d'état.

Le système (4.1) peut se mettre sous la forme suivante :

$$\begin{cases} \dot{x}_1 = a_{11}x_1 + a_{13}x_3 + a_{14}x_4x_5 + b_1v_{ds} \\ \dot{x}_2 = a_{11}x_2 - a_{14}x_3x_5 + a_{13}x_4 + b_1v_{qs} \\ \dot{x}_3 = a_{31}x_1 + a_{33}x_3 - x_4x_5 \\ \dot{x}_4 = a_{31}x_2 + x_3x_5 + a_{33}x_4 \\ \dot{x}_5 = a_{51}x_1x_4 - a_{51}x_2x_3 - a_{52}T_1 \end{cases} \quad (4.2).$$

$$X = [x_1 \ x_2 \ x_3 \ x_4 \ x_5]^T = [i_{ds} \ i_{qs} \ \varphi_{dr} \ \varphi_{qs} \ \omega_r]^T$$

Où i_{ds} et i_{qs} sont les courants statoriques φ_{dr} et φ_{qs} les composantes du flux rotorique, v_{ds} et v_{qs} les composantes de la tension d'alimentation statorique ω_r , la vitesse de rotation angulaire du moteur.

$$\begin{aligned} a_{11} &= -\left(\frac{1}{\sigma L_s}\right)\left(R_s + \frac{L_m^2}{L_r T_r}\right) \\ a_{13} &= \frac{k}{T_r} \quad a_{14} = k \quad a_{31} = \frac{L_m}{T_r} \\ a_{33} &= -\frac{1}{T_r} \quad a_{51} = -\frac{3P^2 L_m}{8J L_r} \quad a_{52} = \frac{P}{2J} \quad b_1 = \frac{1}{\sigma L_s} \\ k &= \frac{L_m}{\sigma L_s L_r} \quad \sigma = 1 - \left(\frac{L_m^2}{L_s L_r}\right) \quad \tau_r = \frac{L_r}{R_r} \end{aligned}$$

Où R_s et R_r sont respectivement les résistances du stator et du rotor de même L_s et L_r , représente l'inductance du stator et du rotor, L_m est l'inductance mutuelle, σ est le facteur de

dispersion, τ_r , représente la constante du temps rotorique. P est le nombre de pole de la machine et J l'inertie du moteur.

IV.3. Procédure d'identification des paramètres du moteur asynchrone par les AGs

Nous allons examiner le problème de déterminer les paramètres du modèle mathématique décrit par l'ensemble des équations différentielles définies dans (4.2). Le problème d'identification des paramètres à la base d'AG peut être décrit comme suit:

$$\dot{X} = f(P, X, V)$$

Où X est le vecteur variable d'état.

$P = [p_1 \ p_2 \ p_3 \dots p_l]^T$ est le vecteur des paramètres à identifier.

Le vecteur de mesure

$$Y(t_k) = [y_1(t_k), y_2(t_k), \dots, y_i(t_k), \dots, y_m(t_k)]$$

Où y_i représente le variable d'état, m est le nombre de variable à mesurer.

Avec les données mesurées on peut identifier le vecteur paramètres optimal \hat{P} en se basant sur les équations suivantes :

$$F(\hat{P}) = \min_{P \in \mathbb{R}^n} F(P);$$

$$F(P) = \sum_{i=1}^m [y(t_i, P) - y_i]^2$$

Le système non linéaire décrit par (4.2) est résolu en utilisant la méthode de Runge kutta, où

$$X = [i_{ds}, i_{qs}, \varphi_{dr}, \varphi_{qr}]^T$$

$$V = [v_{ds}, v_{qs}]^T$$

$$P = [R_s, R_r, L_s, L_r, L_m, J]^T$$

Le vecteur de mesure utiliser dans ce travail est le courant statorique I_s et la vitesse de rotation du moteur ω_r :

$$Y(t_k) = [i_{ds}(t_k), i_{qs}(t_k), \omega_r(t_k)]^T$$

La procédure d'identification des paramètres du système non linéaire à savoir le moteur asynchrone est illustrée dans la figure IV.1

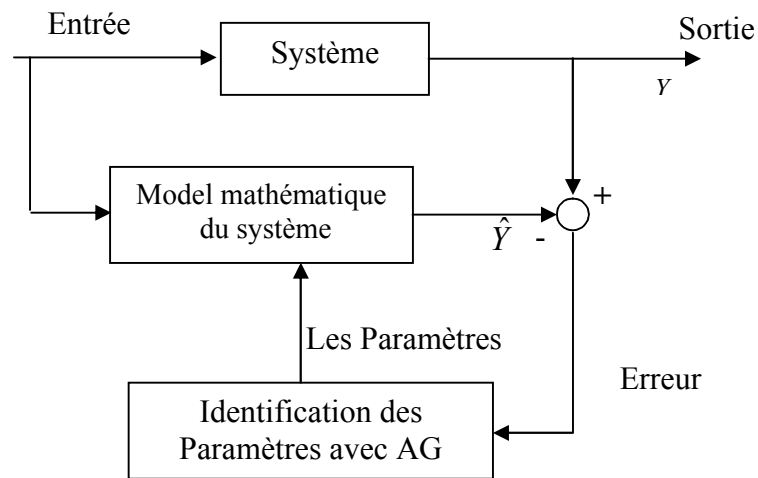


Figure.IV.1. Identification des paramètres par un algorithme génétique

On considérant que g est le nombre de génération, les principaux étapes de l'identification de paramètre d'un système non linéaire en utilisant les AG peut être décrite comme suit :

1^{er} étape : Génération aléatoire de la population initiale des paramètres

La population initiale est générée aléatoirement de taille N individu :

$$\hat{P}^{(0)} = \{ \hat{P}_i^{(0)} \mid i = 1, 2, \dots, N \}$$

Chaque individu \hat{P}_i^g de la population est une solution possible au problème, les paramètres de l'individu du vecteur P sont contraints par les conditions suivantes :

$$p_{\min j} \leq \hat{p}_{ij}^{(g)} \leq p_{\max j} \quad j = 1, 2, \dots, m$$

Où $p_{\min j}$ et $p_{\max j}$ sont les limites "espace de recherche" du $j^{\text{ème}}$ élément du vecteur des paramètres $\hat{p}_{ij}^{(0)}$ donné par une connaissance à priori du système.

$\hat{p}_{ij}^{(g)}$ indique l'estimation de $j^{\text{ème}}$ élément de l' $i^{\text{ème}}$ individu dans la $g^{\text{ème}}$ génération du paramètre. Le processus d'identification commence avec $g = 0$.

Les limites initiales doivent être assez étendues pour contenir l'optimum global. Cependant si rendu trop grand, l'AG ne pourrait pas pouvoir trouver l'optimum global, la convergence sera donc lente et moins d'efficacité. Ainsi, tous les paramètres de valeurs imprécis peuvent

affecter le modèle de machine aux résultats ridicules ou produire erreurs d'exécution, d'où ces limites doivent être bien choisit afin de réduire le temps de convergence tout en évitant d'éliminer des solutions bénéfiques.

2^{ème} étape

Chacun individu $\hat{P}_i^{(g)}$ est employé pour calculer la variable d'état $\hat{X}(t_k)$ et le vecteur de sortie $\hat{Y}(t_k)$ selon :

$$\hat{X}(t_k) = f(\hat{p}_i^{(g)}, \hat{X}(t_k), V(t_k))$$

Le problème d'optimisation de paramètres peut être énoncé comme suit : Déterminer les valeurs de l'ensemble de l paramètres $P = [p_1 p_2 p_3 \dots p_l]^T$ qui réduisent au minimum la fonction de coût du procédé de l'AG.

La structure de la fonction de coût s'écrit :

$$J = C_1 f_1 + C_2 f_2$$

3^{ème} étape

Une nouvelle génération $\hat{P}^{(g+1)}$ avec le même nombre d'individu de $\hat{P}^{(g)}$ est formé au moyen des opérateurs génétiques reproduction, croisement et mutation appliqués à la précédente population.

4^{ème} étape

Le processus s'arrêtera si $g = \max g$ où ($\max g$) est un grand nombre entier représentant le nombre maximum de génération et $P = \hat{P}^{(g)}$ correspondant au minimum de la fonction objectif. Si le critère d'arrêt n'est pas satisfait alors, le processus est répété de l'étape 2.

IV.4. Simulation de l'identification des paramètres du moteur asynchrone par les AGs.

La procédure d'identification basée sur l'AG décrite dans le paragraphe précédent va être utilisée pour identifier les paramètres du modèle dynamique du moteur à induction. Le procédé a été utilisé en mode hors ligne (off line) pour l'identification de paramètres basée sur la phase de démarrage à vide du moteur.

Pour une bonne validation on va procéder à la construction des mesures simulées, ainsi pour ce faire, nous procédons à des simulations de la machine avec des valeurs de paramètres que nous connaissons. En d'autre terme nous considérons une simulation comme étant un essai réel. L'algorithme appliqué pour l'identification doit permettre d'obtenir les mêmes paramètres de cette machine.

Dans le tableau IV.1 on a les paramètres (électriques et mécaniques) du moteur asynchrone qui ont été identifiés par les méthodes classiques, qui vont être utilisé pour la génération de données:

Paramètres	R_s	R_r	L_s	L_r	M	J
valeurs	0.2877 Ω	0.246 Ω	68.12 mH	70.33 mH	66 mH	0.3 Kg m ²

Tableau.IV.1 Paramètres du moteur asynchrone

Ces paramètres vont constitués aussi Le vecteur de paramètre P à identifier par l'AG à savoir la résistance du stator et du rotor R_s , R_r et les inductances de l'enroulement statorique et rotorique L_s , L_r , l'inductance mutuelle L_m et le moment d'inertie du moteur J. Ce moteur asynchrone est quadripolaire (4 pôles) alimenté avec une tension composée de 220V et d'une fréquence de 60Hz.

Pour la simulation du moteur avec les paramètres du Tab IV.1 nous avons développé un programme sous le logiciel MATLAB, en effet ce n'est qu'une résolution numérique des équations différentielles (4.2) par la méthode de Runge kutta d'ordre 4.

Le moteur est alimenté directement à partir des réseaux d'alimentation triphasée équilibrée de tensions simple V_a , V_b , V_c où :

$$\begin{cases} V_a = V_{\max} \sin(\omega_s t) \\ V_b = V_{\max} \sin\left(\omega_s t - \frac{2\pi}{3}\right) \\ V_c = V_{\max} \sin\left(\omega_s t - \frac{4\pi}{3}\right) \end{cases} \quad (4.3)$$

Avec $\omega_s = 2\pi f_s$ et $V_{\max} = \frac{220}{\sqrt{3}}V$.

Les courbes de la figure (IV.2) représentent les résultats de simulation du démarrage à vide du moteur asynchrone. Le modèle est simulé dans le référentiel (d,q) lié au champ tournant.

La figure (IV.2.a) montre l'évolution du courant statorique en fonction du temps, en reconnaît les réponses du moteur asynchrone alimentée en tension. Au démarrage le courant statorique est excessif, vu qu'il est de courte durée, le moteur pourrait le supporter sans risque d'un échauffement dangereux. Cette valeur de courant (régime transitoire) disparaît au bout de quelques alternances afin d'obtenir une forme sinusoïdale (régime permanent).

L'évolution de la vitesse de rotation du moteur en fonction du temps est illustrée dans la figure (V.2.b) où elle présente des oscillations dans les premiers instants de démarrage avec un accroissement presque linéaire. La vitesse atteint la valeur de 376.9911rad/s, sans dépassement et se stabilise au régime permanent.

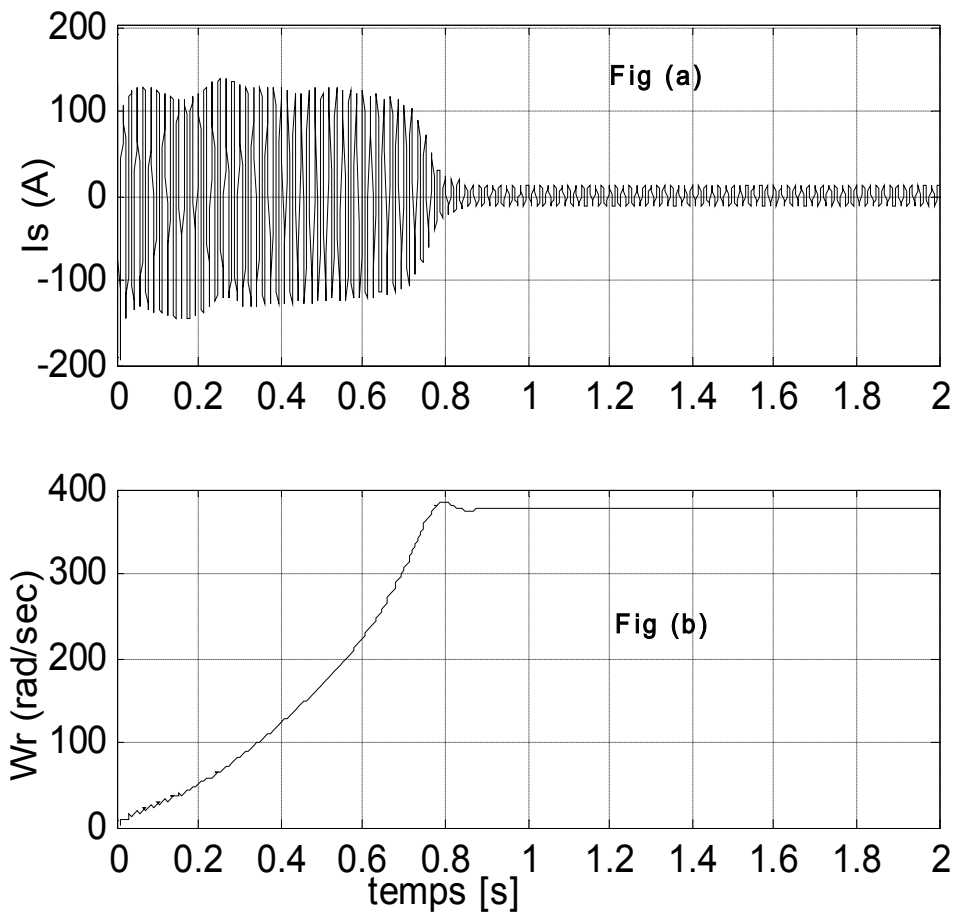


Figure. IV.2 Simulation du courant statorique Fig (a) et de la vitesse de rotation Fig (b)

L'algorithme génétique proposé (au 2^{ème} chapitre) dont la fonction objectif est à coefficient de pondération variable étant employée pour identifier les paramètres du moteur asynchrone (Figure.IV.3) où la méthode proposée emploie les données obtenu à partir de la simulation d'un moteur (cette méthode pourrait également être employée avec des données expérimentales) où est I_s , V , ω_r sont respectivement le courant, tension d'alimentation et la vitesse de rotor.

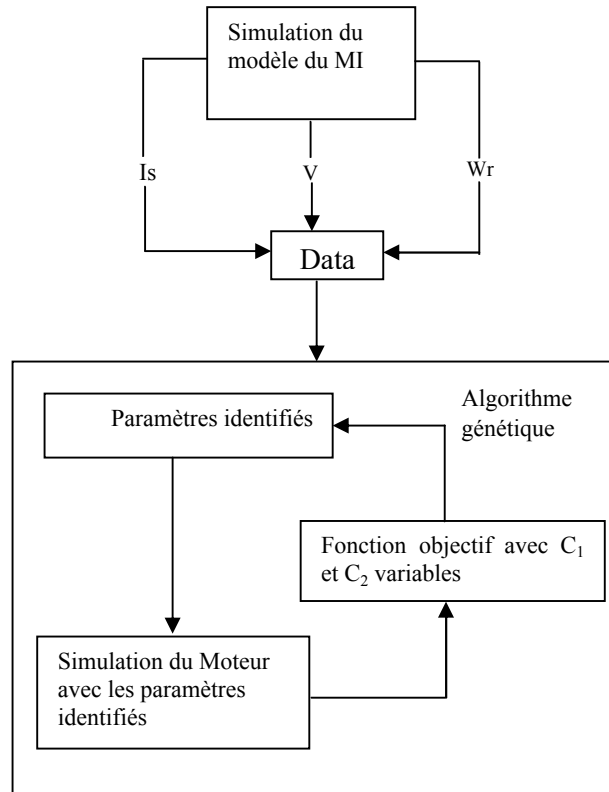


Figure.IV.3. Méthodologie d'identification des paramètres.

L'essai du procédé proposé pour l'identification est réalisé seulement par l'intermédiaire de la simulation du moteur avec un modèle connu. (4.2)

L'identification des paramètres est basée sur l'AG qui choisit, à chaque génération, un ensemble de valeurs de paramètre et évalue la réponse du moteur simulé. Si la différence entre les résultats (data) et simulés est au-dessous d'un niveau satisfaisant fixe, nous pouvons indiquer que les paramètres du moteur sont identifiés. Autrement, le procédé d'optimisation continuera jusqu'à ce que le critère d'arrêt de l'AG soit vérifié.

La fonction objectif à optimisée (utilisée dans les techniques soft computing proposé dans ce travail) est une sommation quadratique des différences entre le courant statorique et de la vitesse de la machine réelle et celui du modèle. Cette fonction dépend des six paramètres qui seront par la suite identifiés :

$$J = C_1 \sum_{k=1}^N \left(i_s(t_k) - \hat{i}_s(t_k) \right)^2 + C_2 \sum_{k=1}^N \left(\omega_r(t_k) - \hat{\omega}_r(t_k) \right)^2 \quad (4.3)$$

Les caractéristiques de l'AG employées pour identifier le vecteur de paramètre utilisé dans ce travail sont indiquées dans le Tableau suivant :

Paramètres	VALEUR	Description
Popsiz	50	Taille de la population
Length	12	Nombre de bit pour chaque paramètre
Maxgen	100	génération maximum
Pc	0.6	Taux de croisement
Pm	0.01	Taux de mutation

Tableau.IV.2 Les Caractéristiques de l'Algorithme Génétique

IV.4.1. Résultats et discussion

L'algorithme génétique proposé fonctionne avec la population initiale aléatoire. C_1 et C_2 sont les poids de pondération de la fonction objectif de l'AG variant dans l'intervalle [0 1].

Sept essais d'exécution initiale de cet algorithme améliorée avec une valeur dynamique des poids de pondération de la fonction de coût. Les résultats d'identification des paramètres obtenus sont affichés dans le tableau IV.3. L'examen de ce tableau prouve que les meilleurs paramètres identifiés sont ceux qui se trouvent dans la quatrième ligne de ce tableau.

Les paramètres du moteur avec les paramètres identifiés par l'algorithme génétique modifié sont montrés dans le tableau IV.4.

N	C₁	C₂	R_r	R_s	L_r	L_s	L_m	J	F
1	0.013431	0.007814	0.249743	0.277838	0.070426	0.067868	0.067704	0.301076	0.010200
2	0.131135	0.004395	0.244542	0.295494	0.068105	0.070311	0.065767	0.291897	0.000200
3	0.094993	0.091330	0.245860	0.286813	0.070004	0.068399	0.065822	0.308725	0.000519
4	0.013675	0.091575	0.246007	0.285970	0.069582	0.068824	0.066785	0.305865	0.000196
5	0.013431	0.007814	0.249743	0.277838	0.070742	0.067761	0.065767	0.303941	0.000204
6	0.094993	0.091330	0.245860	0.286813	0.069793	0.068611	0.065822	0.301655	0.000382
7	0.227838	0.004395	0.250183	0.273663	0.069687	0.068717	0.066350	0.307980	0.000411

Tableau. IV 3 Résultats de l'identification des paramètres du moteur asynchrone par l'algorithme génétique modifié avec sept essais

Paramètres	References	Identifiés
R_r	0.2460	0.246007
R_s	0.2877	0.285970
L_s	0.06812	0.068824
L_r	0.07033	0.069582
L_m	0.06600	0.066785
J	0.300000	0.305865

Tableau.IV.4. Résultats de l'identification des paramètres du moteur asynchrone

La figure IV 5 présente les résultats de la minimisation de la fonction de coût au cours des générations avec les poids optimisés C_1 et C_2 (les valeurs indiqués dans la quatrième ligne de la table IV.3), où cette fonction de coût diminue avec l'augmentation du nombre de génération et l'AG converge presque près de la soixante quatorzième génération. Le nombre de génération nécessaire pour que la fonction objectif ait une valeur au dessous de 0.05 est de 42 générations.

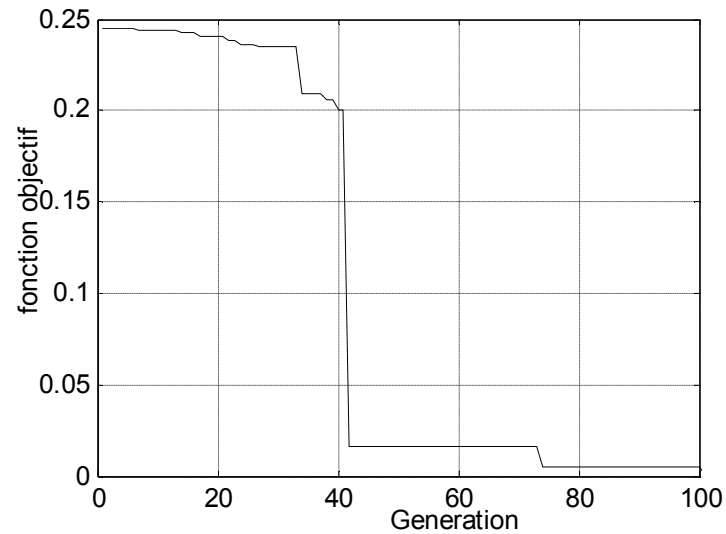


Figure. IV 4. Evolution de la fonction de coût avec les poids variables

La Figure.IV.6 présente l'erreur en pourcentage (%) des paramètres identifiés qui sont évalués à partir de l'expression suivante :

$$error (\%) = \left| \frac{P - \hat{P}}{P} \right| \times 100 \quad (4.4)$$

Où P indique le paramètre à identifier

La Figure IV.7 montre la convergence de l'AG pour identifier chacun des paramètres du moteur, elle indique la meilleure valeur des paramètres dans chaque génération. On constate que les meilleures valeurs sont obtenues à partir de la 73^{ième} génération.

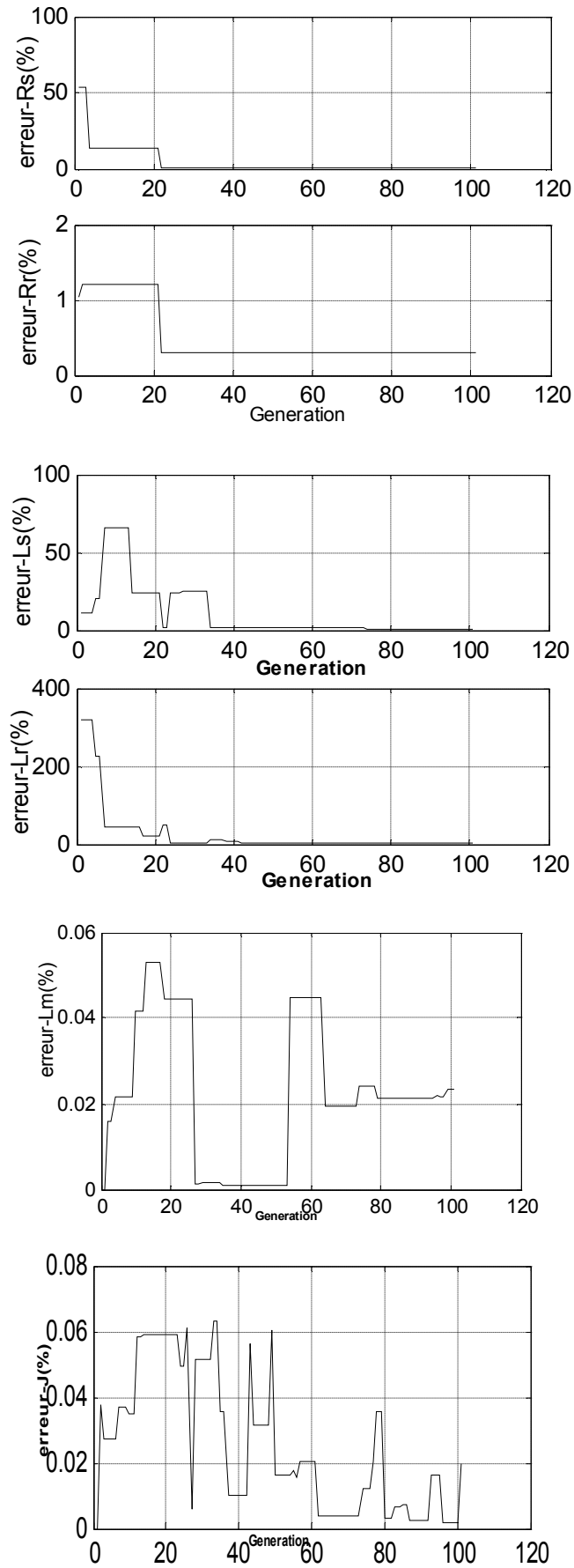


Figure.IV.5. Evolution de l'erreur relative des paramètres du moteur asynchrone

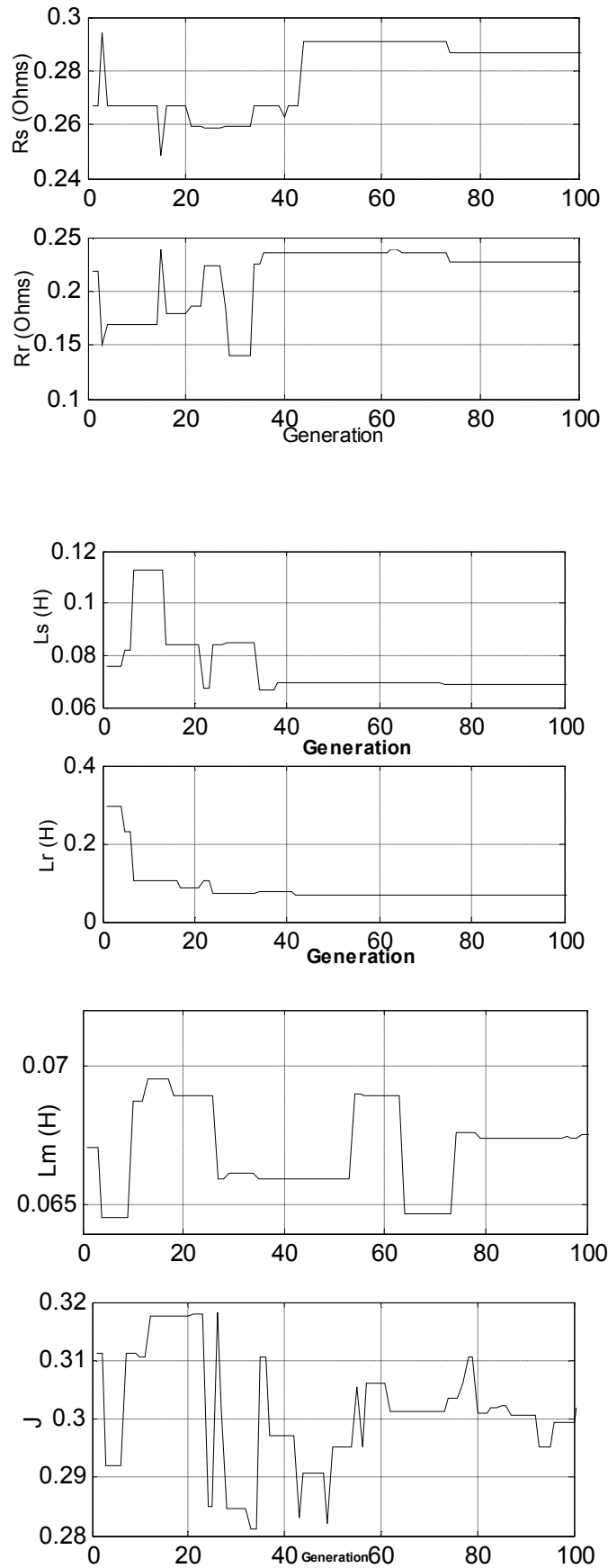


Figure. IV 6. Identification des paramètres par l'algorithme génétique amélioré

L'erreur d'identification des inductances sont au-dessous de 2.2630 % à la 42 génération et prennent une valeur constante à la soixante-cinquième génération de 1.2130 % les meilleures valeurs de la résistance de rotor et de stator sont obtenus à la soixante-quatorzième génération.

Les valeurs des paramètres du moteur asynchrone identifiées avec l'AG modifié proposées dans ce travail sont avec une erreur moins de 1% des paramètres réel du moteur. L'identification de Paramètre dans [41], est considérée extrêmement satisfaisante si l'erreur est de plus ou moins de 5% de la valeur vraie, qui est la meilleure précision pour l'identification des paramètres en utilisant des techniques traditionnelles d'identification de système.

Ces résultats montrent que la convergence a été obtenue dans 74^{ième} génération avec cet algorithme génétique proposé. Une Comparaison de la fonction objectif avec les poids variables et ceux avec des valeurs constantes d'un AG (Figure.IV.7) montre une excellente performance dans le terme du temps d'exécution et de vitesse de convergence qui sont donné dans le tableau IV.5 où la comparaison de ces deux algorithme est effectué dans le même état des caractéristiques de GA et dans le même espace de recherche.

Fonction objectif	Poids variables	C₁=0.5 C₂=0.5	C₁=0.8 C₂=0.2
Temps d'exécution	13.5 min	20 min	23 min

Tableau.IV.5. Comparaison du temps d'exécution de l'AG avec une fonction objectif à poids de pondérations variables et constants

Concernant la convergence, il est clair qu'à partir de la figure.IV.7 on constate que l'AG dont la fonction objectif possédant des poids de pondération variables est l'algorithme qui converge rapidement si on emploie le nombre de génération nécessaire pour atteindre une valeur de F au-dessous de 0.05 en tant que mesure de vitesse de convergence. Alors que pour l'AG dont la fonction objectif possédant des poids de pondération constants a eu besoin de presque plus de génération (plus 60 générations) pour être au dessous de cette valeur.

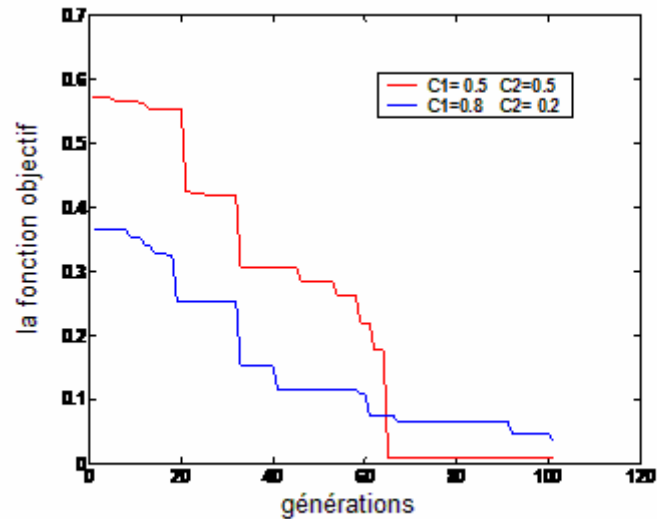


Figure IV. 7. La minimisation de la fonction de coût avec les poids de pondérations constants.

Une comparaison des courbes du courant statorique entre le modèle du moteur à induction avec des paramètres réel et le modèle utilisant paramètres identifiés par l'AG modifié durant la phase de démarrage pendant 0.1 seconde sont montrés dans la Figure.IV8.

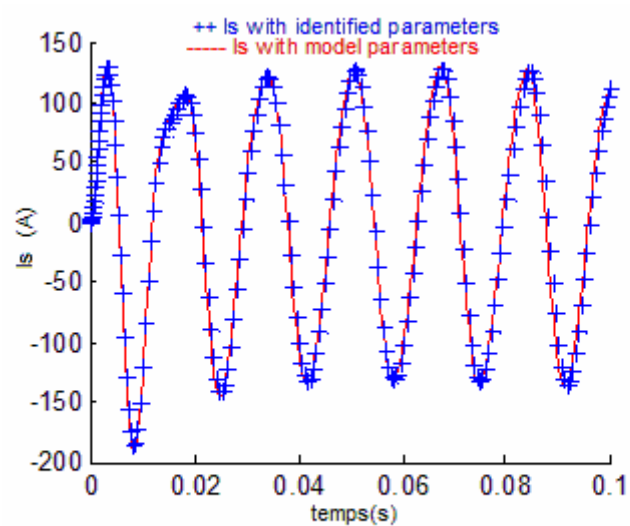


Figure. IV 8. Comparaison entre le courant statorique obtenu avec les paramètres identifiés et les paramètres réels du moteur.

IV.5. Simulation de l'identification des paramètres du moteur asynchrone par les AGs avec le codage modifié.

L'efficacité de la méthode de codage modifié (RCD) présentée dans le chapitre II sera appliquée à l'identification des paramètres d'un moteur à induction.

Les paramètres du moteur asynchrone, utilisés pour la génération de données (références) sont donnés dans le tableau I.V.1.

Les caractéristiques de l'AG avec codage (RCD) employées pour identifier le vecteur de paramètre utilisé dans ce travail sont indiquées dans le Tableau suivant :

Paramètres	VALEUR	Description
Popsiz	50	Taille de la population
Maxgen	400	génération maximum
Pc	0.8	Taux de croisement
Pm	0.3	Taux de mutation

Tableau.IV.6 Les Caractéristiques de l'AG avec le codage (RCE)

IV.5.1. Résultats et discussion

Les résultats obtenus de la minimisation de la fonction objectif lors de l'application de la procédure d'identification par l'AG avec un codage modifié sont montrés dans la figure IV.10, qui montre l'évolution de la fonction objectif au cours des générations. On constate une décroissance de la courbe qui correspond à une convergence vers l'optimum à la 33^{ème} génération.

Dans la figure IV.11, l'évolution de l'erreur en pourcentage des paramètres à identifier est illustrée.

La figure IV.12 montre les valeurs des paramètres du moteur identifiés par l'AG avec codage modifié (RCD), cette courbe représente la meilleure valeur de paramètres pour chaque génération. On constate que les meilleures valeurs acceptables sont obtenues à partir de la 133^{ème} génération. Malgré qu'on a limité l'exécution à 400 générations, les meilleurs valeurs des paramètres peuvent s'améliorer encore si on augmente le nombre de générations.

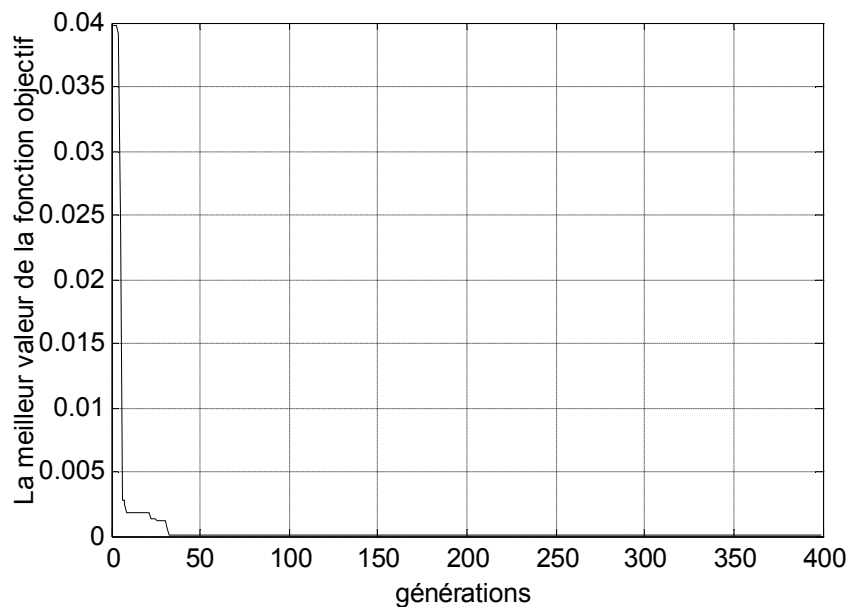


Figure.IV.9. Evolution de la fonction objectif de l'AG avec codage (RCD)

Les résultats de l'identification des paramètres obtenus par l'AG avec codage modifié (RCD), sont indiqués dans le tableau IV.8. Ce tableau montre les meilleurs paramètres identifiés au cours de toutes les générations.

Paramètres	References	Identifiés
Rr	0.2460	0.245000
Rs	0.2877	0.286849
Ls	0.06812	0.068338
Lr	0.07033	0.070323
L_m	0.06600	0.066113
J	0.300000	0.300971

Tableau.IV.7. Résultats de l'identification des paramètres par l'AG avec codage (RCD)

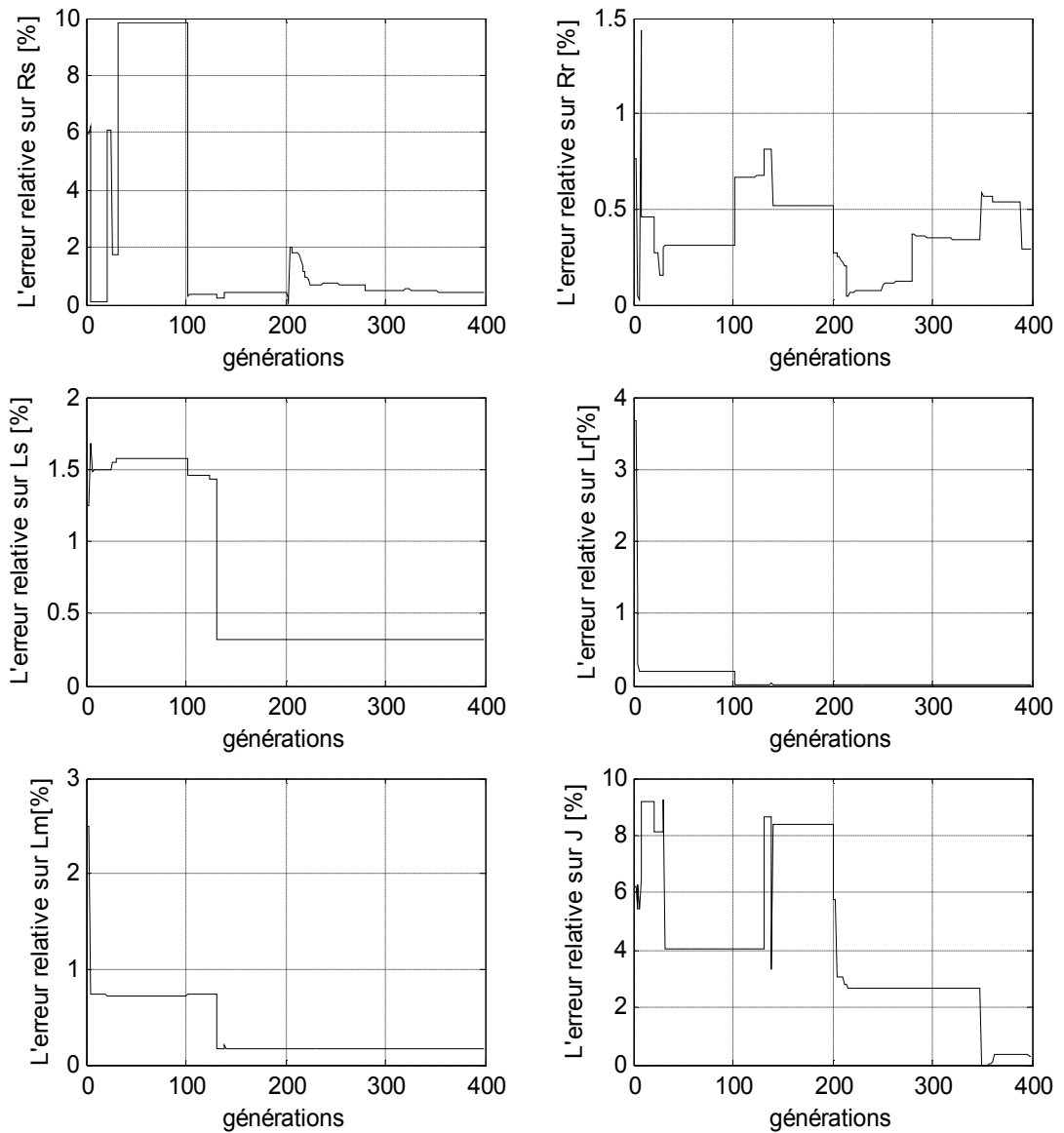


Fig.IV.10. Evolution de l'erreur relative des paramètres du moteur asynchrone Identifiés par l'AG avec codage (RCD)

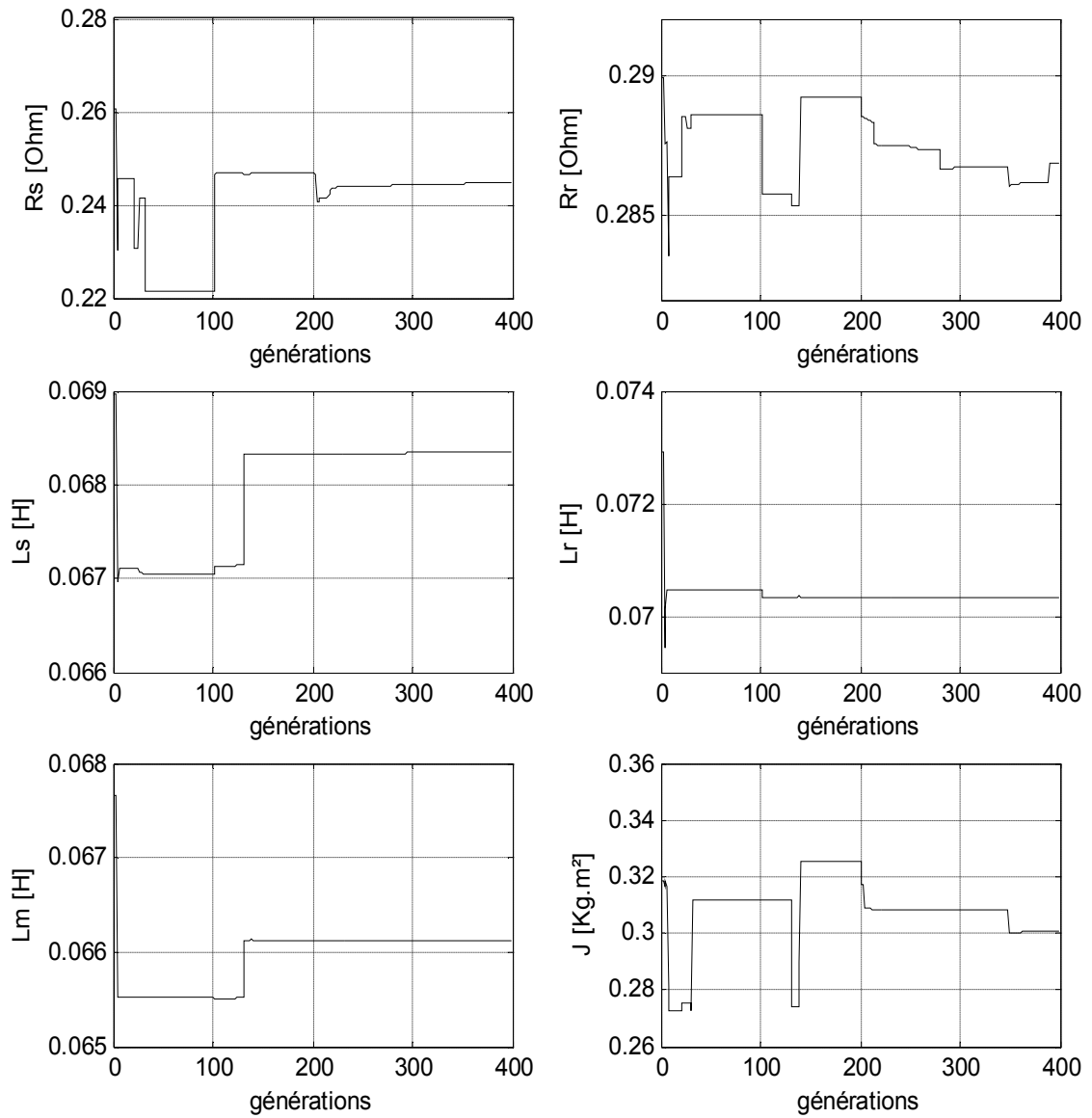


Fig.IV.11. Evolution des paramètres identifiés par l'AG avec codage (RCE)

IV.6. Simulation de l'identification des paramètres du moteur asynchrone par la technique OEP modifiée.

La méthode de OEP dont l'organigramme a été présenté dans le chapitre III, va être appliqué pour l'identification des paramètres du moteur asynchrone dont les paramètres sont indiqués dans le tableau IV.1. Les paramètres de la technique OEP utilisés sont indiqués dans le tableau IV.9 :

Paramètres	VALEUR	Description
N	60	Nombre de particule dans l'essaim
Iter	100	Nombre d'itérations maximum
w	0.4-0.9	Facteur d'inertie
$c_1 = c_2$	0.5	Coefficients d'accélération

Tableau.IV.8 Les paramètres de l'EP modifié.

Il est à noter que chaque particule est de dimension 6 (les paramètres à identifier) à savoir : $R_s, R_r, L_s, L_r, L_m, J$.

IV.6.1. Résultats et discussion

L'optimisation de la fonction objectif au cours de la procédure d'identification est illustrée dans la figure IV.13.

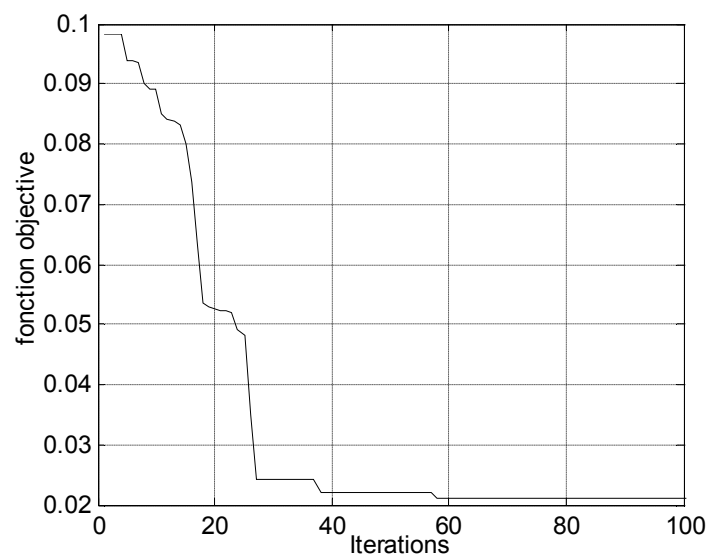


Figure IV.12. Evolution de la fonction objectif de l'OEP modifiée.

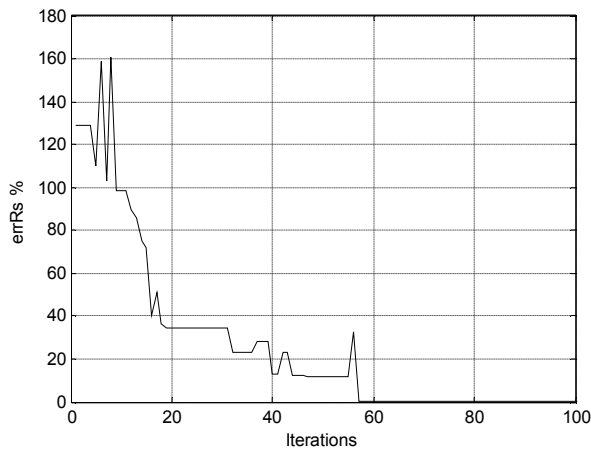
La figure IV.14 montre la variation des paramètres en fonction des itérations.

Les résultats d'identification des paramètres obtenus par l'EP sont indiqués dans le tableau IV.10. Ce tableau nous montre les meilleurs paramètres identifiés au cours des cents itérations.

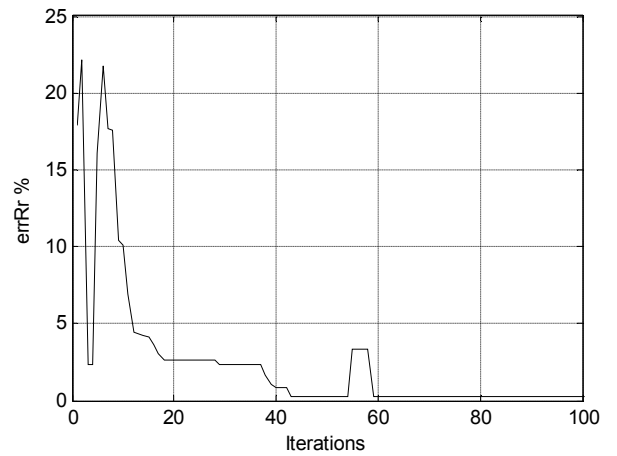
Paramètres	Référence	Identifiés
Rr	0.2460	0.246050
Rs	0.2877	0.287300
Ls	0.06812	0.068001
Lr	0.07033	0.070310
L_m	0.06600	0.066785
J	0.300000	0.301514

Tableau.IV.9. Résultats de l'identification des paramètres par l'OEP modifiée.

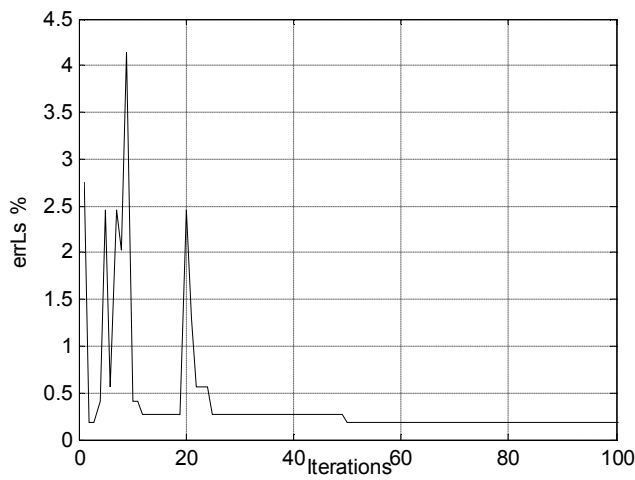
Les résultats prouvent que l'algorithme EP est comparable à l'AG. Il est important de préciser que cette méthode converge aux solutions fortement optimales après 63 itérations.



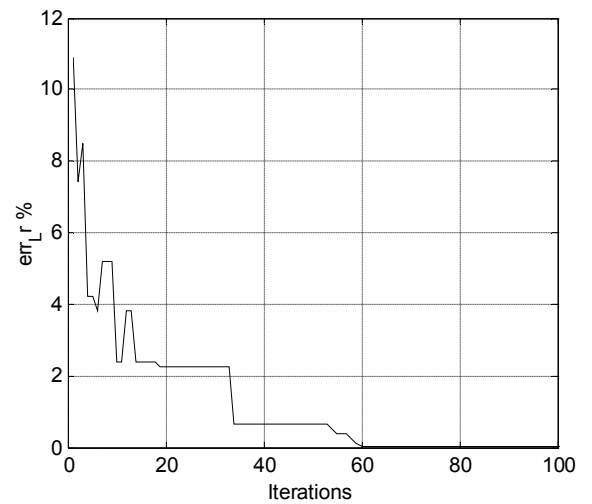
(a) erreur Rs (%)



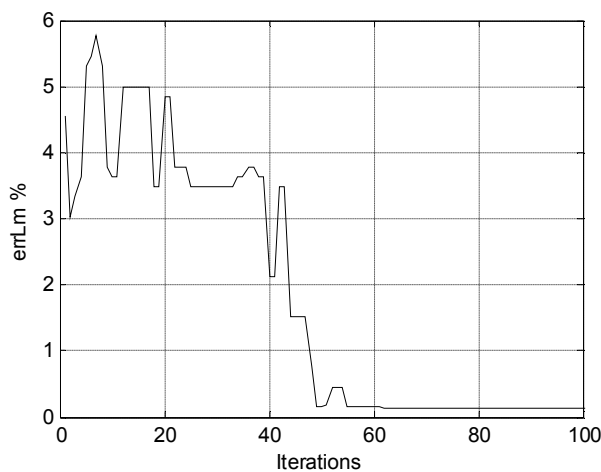
(b) erreur Rr (%)



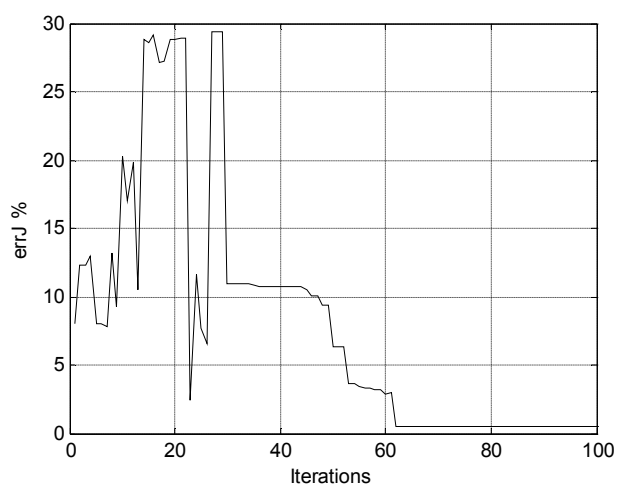
(c) erreur Ls (%)



(d) erreur Lr (%)

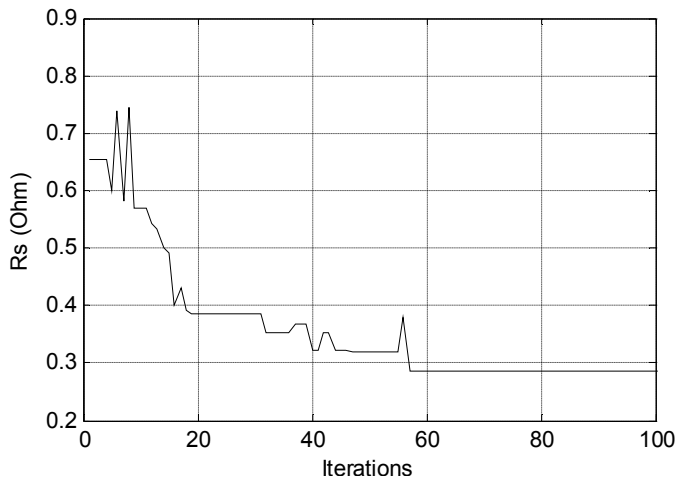


(c) erreur Lm (%)

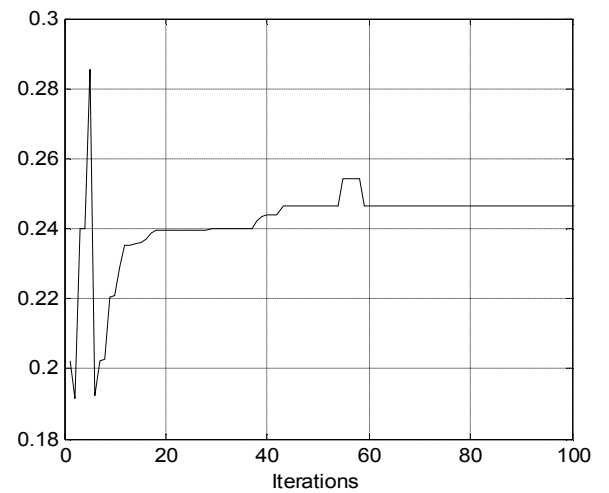


(d) erreur J (%)

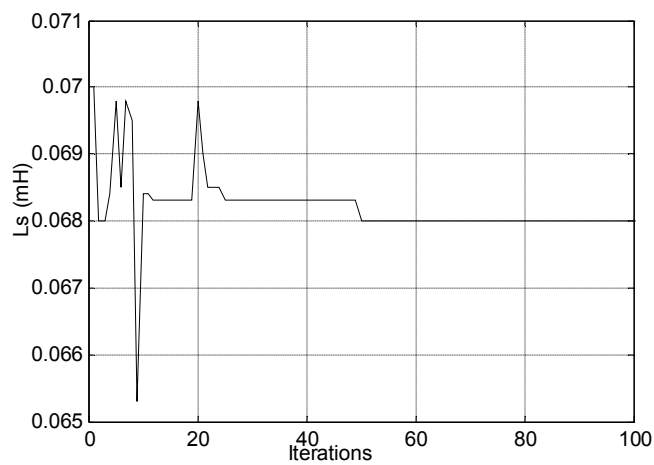
Fig.IV.13. Evolution de L'erreur relative des paramètres du moteur asynchrone basée sur OEP Modifié.



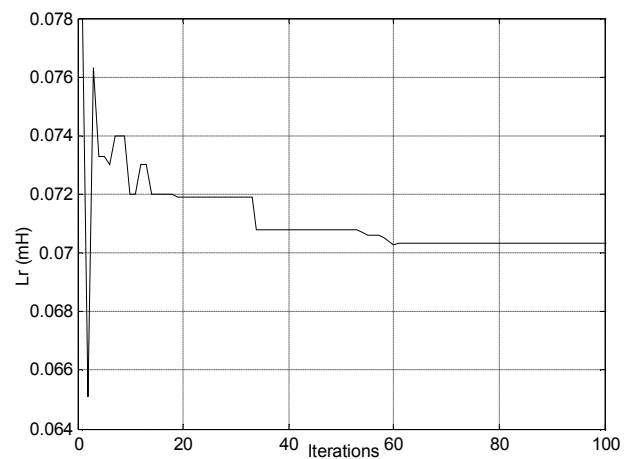
(a) Résistance R_s



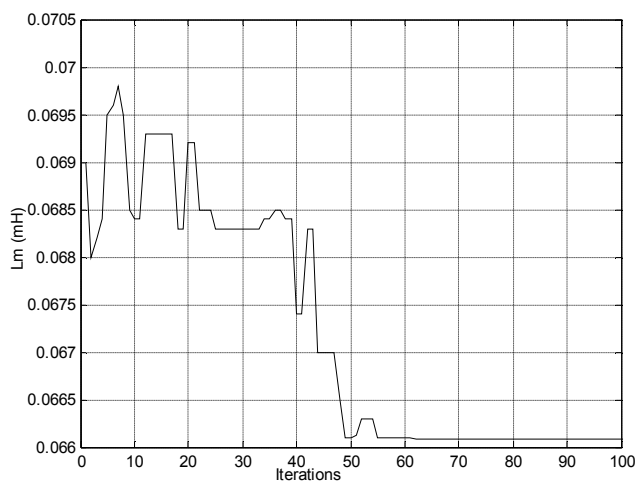
(b) Résistance R_r



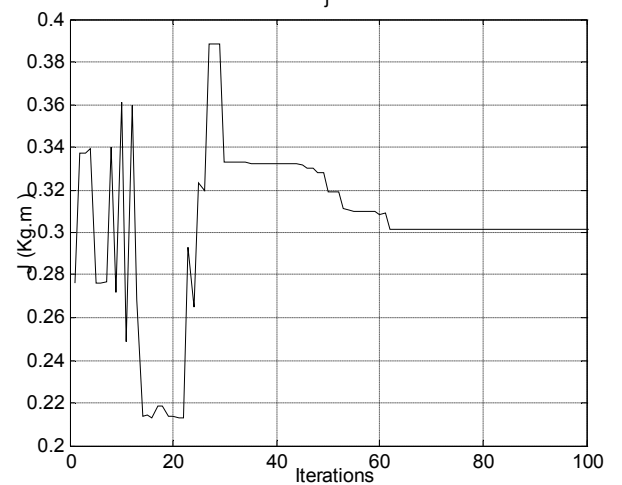
(c) L'inductance L_s



(d) L'inductance L_r



(e) L'inductance L_m



(f) Moment d'inertie J

Figure IV 14. Identification des paramètres du moteur asynchrone basée sur OEP Modifié.

IV.7. Comparaison des trois méthodes

Pour permettre la comparaison entre les trois méthodes qu'on a utilisées pour l'identification des paramètres du moteur asynchrone, on a regroupé les résultats des trois techniques dans le tableau avec l'erreur relative.

Paramètres	Valeurs connues	Résultats de l'identification			L'erreur relative (%)		
		AG	AG(RCD)	OEPm	AG	AG(RCD)	OEPm
Rr	0.2460	0.246007	0.245000	0.246050	0.002845	0.406509	0.02032
Rs	0.2877	0.285970	0.286849	0.287300	0.601320	0.295959	0.13903
Ls	0.06812	0.068824	0.068338	0.068001	1.03347	0.320305	0.174691
Lr	0.07033	0.069582	0.070323	0.070310	0.02843	0.010379	0.028437
Lm	0.06600	0.066785	0.066113	0.066775	1.18939	0.170684	1.17424
J	0.300000	0.305865	0.300971	0.301514	1.955	0.323737	0.50466

Tableau IV.11 Comparaison des trois méthodes d'identifications

On remarque que l'erreur relative de tous les paramètres du moteur obtenus par l'AG avec codage RCD sont tous inférieur à 0.5%, par contre il y a trois paramètres (Ls, Lm et J) dont l'erreur relative est supérieur à 0.5 % avec l'AG modifié et un seul paramètre (Lm) avec l'OEP. Ainsi la technique qui donne des résultats précis est la technique de l'AG avec codage RCD.

L'erreur relative des paramètres atteignent des valeurs moins de 5% à partir de la 41^{ème} génération pour l'AG et à partir de la 102^{ème} itération pour L'AG avec codage RCD et à partir de la 63^{ème} itération pour l'OEP ce qui montre que les meilleurs résultats peuvent être obtenu plus rapidement avec la technique de l'AG amélioré.

IV.8. Conclusion

Dans ce chapitre on a l'identifier les paramètres du moteur asynchrone en se basant sur les algorithmes améliorées décrit dans les chapitres précédents. Les AG proposés possède une meilleure convergence globale pour l'identification de ces paramètres, et remédie ainsi au problème des méthodes traditionnelles d'identification. Les résultats de simulation obtenus

montre que les algorithmes génétiques avec les poids de pondération variables de la fonction de objectif au cours du processus d'optimisation a une influence sur la convergence du procédé de l'AG. Les résultats de l'identification obtenue avec l'AG avec codage (RCD) montrent qu'il donne une solution optimale avec une très bonne précision mais nécessite un nombre de génération relativement grand que les deux autres méthodes. La technique d'OEP amélioré avec l'évolution des coefficients r_1 et r_2 de manière adaptative permet d'améliorer la convergence vers la solution optimale.

A la fin de ce chapitre une comparaison entre les trois méthodes a été établit qui montre qu'elles convergent presque vers la solution optimale.

Conclusion Générale et Perspectives.

L'objectif de ce travail était d'étudier les potentiels des techniques Soft Computing principalement les algorithmes génétiques et l'essaim de particules comme méthodes de résolution pour un problème d'identification des paramètres d'un système non linéaire. Pour atteindre ce but, nous avons proposé dans le deuxième chapitre une technique évolutionnaire basée sur les algorithmes génétiques dont les coefficients de pondération de la fonction objectif sont variables au cours du processus d'optimisation. En vue d'obtenir une meilleure précision des valeurs des paramètres identifiés, nous avons proposé aussi un AG modifié basé sur une technique de codage des paramètres qu'on a nommée « réelle codé décimal (RCD) ». Dans le troisième chapitre, on a proposé de modifier la technique OEP en faisant varier les coefficients r_1 et r_2 de manière adaptative pour atteindre la solution globale.

Pour évaluer nos approches proposées, nous avons pris comme exemple d'application d'un système non linéaire, un moteur asynchrone, étant donné son importance pratique dans les systèmes d'entraînement industriel. Les techniques modifiées de l'AG et de l'OEP ont été appliquées à l'identification des paramètres de ce moteur. Les résultats obtenus sont très encourageants.

Les résultats de simulation obtenus par les algorithmes génétiques avec les poids de pondération variables de la fonction objectif ont montrés l'influence de ces poids de pondération sur la convergence du procédé de l'AG. L'AG avec codage (RCD) a réussi d'aboutir à une solution optimale avec une très bonne précision mais avec un nombre de génération relativement élevé par rapport aux autres méthodes proposées. Les résultats ont montré que la technique OEP modifié est bien adaptée pour déterminer les valeurs des paramètres à identifier puisqu'elle nécessite que peu de paramètres à ajuster.

En conclusion, les trois méthodes proposées sont meilleures du fait que la précision des paramètres obtenus est inférieure à 5%. Cependant, elles présentent un point faible sérieux qui est son temps d'exécution considérablement long qui confine son application seulement à l'identification hors ligne (off line identification).

Les techniques de l'AG et l'OEP sont donc des méthodes alternatives et efficaces pour la résolution des problèmes d'identification des paramètres d'un système non linéaire.

Les perspectives que nous pouvons donner à ces travaux sont de plusieurs ordres :

- Sur le plan des applications, d'autres problèmes d'identification des paramètres peuvent être facilement adaptés à nos techniques proposés comme le problème d'identification des paramètres de 'circuit magnétique', le problème d'identification MSAP, ...etc.

- Sur le plan des performances, pour obtenir des solutions de qualités supérieures, il serait intéressant d'introduire les autres approches des techniques Soft Computing pour les problèmes d'identification des paramètres. On peut aussi se pencher sur l'utilisation d'algorithmes génétiques multi-objectif puisqu'il s'agit de minimiser à la fois l'erreur du courant statorique et de la vitesse.

Bibliographie

- [1] T. Soderstorm and P. Stoika, “System Identification ” Printice Hall international, (1989).
- [2] I.D. Landau, A.Besançon-Voda. “Identification des systèmes ”, édition HERMES Sciences publication (2001).
- [3] A. Khouaja, “ Modélisation et identification des systèmes non linéaire à l’aide de modèles de voltterra à complexité réduite ”, Thèse doctorat, de l’université de Nice Sophia Antipolis, France (2005).
- [4] B. Picinbono, “Signaux aléatoires, Tome 3, bases du traitement statistique du signal avec Problèmes résolus ”. Dunod, Paris, (1995).
- [5] L. Ljung, “System Identification : Theory for user ”, Printice Hall international , (1987).
- [6] J.Sjoberg, Q. Zhang, L.Ljung, A. Benveniste, B. Delyon, P Y Glorrenec, H. Hjalmarson and A. Juditsky, "Non linear black box modeling in system identification: a unified overview", Automatica, 31 (12),: pp 1691-1724, (1995).
- [7] Bhama, S. ;Singh, H., “Single layer neural networks for linear system identification using gradient descent technique”, IEEE Trans, Neural Networks, vol,4, pp 884-888, Sep (1993).
- [8] B. Jiang, B.W. Wang, “Parameter estimation of nonlinear system based on genetic algorithm”, Control Theory Appl, vol 17, , pp 150-152, (2000)
- [9] C.Picardi, N.Rogano, N. “Parameter identification of induction motor based on particle swarm optimization”, Inernational Symposium on SPEEDAM, pp 968-974, 23-26 May (2006).
- [10] S.Lixia ; Q.Ping; H.Qixin, J.Ping, “Parameter Identification of Synchronous Generator by Using Ant Colony Optimization Algorithm”, 2nd IEEE Conference ICIEA, pp 2834 – 2838, 23-25 May, (2007).

- [11] F.Benoit-Marand," Modélisation et Identification des Systèmes Non Linéaires Par Réseaux de Neurones à Temps Continu. Application à la Modélisation des Interfaces de diffusion Non Linéaires". Thèse doctorat, l'Université de Poitiers, France, (2007).
- [12] J.Rachelet, "Pratique de l'identification ", HERMES, Paris, (1997).
- [13] E. Walter and L.Pronzato. , " l'identification of parametric models from experimental data ", Sprnger, New york,(1997).
- [14] J.J.destifano and C. Cobelli , "On parameter and structural identifiability: non unique observability / reconstructability for identifiable systems, other ambiguities and new definitions", IEEE Trans on Automat. Control, AC 25 (4) pp 830-833, (1980).
- [15] J. Sjoberg, Q. Zhang, L. Ljung, A. Benveniste, B.Deylon, P-Y. Glorennec, H.Hjalmarsson, and A. Juditsky, " Non linear black-box modeling in system identification: a unified overview", Automatica, 31 (12), pp: 1691-1724, (1995).
- [16] P.Borne, G.DauphinTanguy, JP Richard, F Rotilla, 'Modélisation e identification des processus Tome 1", editions TECHNIP, (1992).
- [17] G.MERCERE "Contribution à l'identification récursive des systèmes par 'approche des sous-espaces". Thèse doctorat, l'Université des Sciences et Technologies de LILLE, Lille, France, (2004).
- [18] M. Onder Efe et al., "A comparative study of soft-computing methodologies in identification of robotic manipulators", Robotics and Autonomous Systems, Vol.30, 221-230, (2000).
- [19] Yuehui Chen, "Hybrid Soft Computing Approach to Identification and Control of Nonlinear Systems", Phd thesis, kumamoto university, (2001).
- [20] P. J. Angeline and D. B. Fogel, "An Evolutionary Program for the Identification of Dynamical System", Proc. of SPIE (Volume 3077): Application and Science of Artificial Neural Networks III, S.Rogers (ed.), SPIE-The International Society for Optical Engineering, Bellingham, WA., 409-417, (1997).

- [21] F.Benoit-Marand," Modélisation et Identification des Systèmes Non Linéaires Par Réseaux de Neurones à Temps Continu. Application à la Modélisation des Interfaces de Diffusion Non Linéaires. Thèse doctorat, l'Université de Poitiers, France, (2007)
- [22] F. Alonge, F .D'Ippolito, F.M. Raimondi, "Least squares and genetic algorithms for parameter identification of induction motors", *Control Engineering Practice* 9 ; pp 647-657, Elsevier Science (2001).
- [23] Rasmus K. Ursema, Pierré Vadstrup "Parameter identification of induction motors using stochastic optimization algorithms", *Applied Soft Computing* 4 pp 49–64, Elsevier Science,(2004).
- [24] Jong-Hwan Kim, Jeong-Yeol, Jeontl Hong-Kook Chae and Kwangill Koh,"A Novel Evolutionary Algorithm with Fast Convergence", *IEEE* (1995).
- [25] Hai Jin, Shouguang Ma and Pengying Du, "Application of genetic algorithms in parameters identification of asynchronous motor", *Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics San Antonio, TX, USA - October* (2009).
- [26] Jianjun Chen , Xiaohua Yang, "Optimal parameter estimation for Muskingum model based on Gray-encoded accelerating genetic algorithm", *Communications in Nonlinear Science and Numerical Simulation* 12; pp: 849–858, Elsevier Science, (2007).
- [27] Wei-Der Chang, "Nonlinear system identification and control using a real-coded genetic Algorithm", *Applied Mathematical Modelling*; 31, pp: 541–550, Elsevier Science, (2007).
- [28] Nedim Tutkun, "Parameter estimation in mathematical models using the real coded genetic algorithms", *Expert Systems with Applications*, 36, pp : 3342–3345, Elsevier Science, (2009).
- [29] Li Liu, David A. Cartes and Wenxin Liu, " Particle Swarm Optimization Based Parameter Identification Applied to PMSM",*Proceedings of the American Control Conference IEEE*, pp :2955-2960, New York City, USA, July 11-13, (2007).
- [30] Alireza Alfi, Hamidreza Modares, "System identification and control using adaptive

- particle swarm optimization”, *Applied Mathematical Modelling* , 35 ; pp : (2011) 1210–1221, Elsevier Science, (2011).
- [31] C. Darwin, ” On the Origin of Species by Means of Natural Selection”, John Murray, London, U.K, (1958).
- [32] T. Back, ”Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms (Hardcover) ”, Oxford University Press, USA, (1996).
- [33] D. Whitley, ” Foundations of Genetic Algorithms 2 ”, US Edition, California, USA, (1993).
- [34] L. Fogel., A.J. Owens, M.J. Walsh, ”Artificial Intelligence through Simulated Evolution”, Wiley, Chichester, UK, (1966).
- [35] I. Rechenberg, ” Cybernetic Solution Path of an Experimental Problem ”, Royal Aircraft Establishment Library Translation, Farnborough, U.K, (1965).
- [36] J.H. Holland, ”Adaptation in Natural and Artificial Systems”, Ann Arbor: The University of Michigan Press, USA, (1975).
- [37] P.Preux, ” Les Algorithmes Evolutifs , Publication LIL 94-1. Laboratoire d’Informatique Fondamentale de Lille, Octobre (1995).
- [38] HOUSROUM Haiyan, ” Une approche génétique pour la résolution du problème VRPTW dynamique” these de doctorat (2005)
- [39] Man, K.F., K.S. Tang et S. Kwong. Genetic algorithms: Concepts and applications. *IEEE Transaction on industrial electronics* 43, pp: 519–534,(1996)..
- [40] Wu, C.J. et G.Y. Lin). Design of fuzzy logic controllers using genetic algorithms. In: *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*. Vol.6. SAAA. pp. 104–109. (1999)
- [41] T. Vallée et Murat Yildizoglu ‘Présentation des algorithmes génétiques et de leurs applications en économie’ , sep (2001).

- [42] O. Guenounou ‘Méthodologie de conception de contrôleurs intelligents par l’approche génétique- application à un bioprocédé’, thèse doctorat université de Toulouse,(2009).
- [43] L. Tamine, ‘Optimisation de requêtes dans un système de recherche d’information approche basée sur l’exploitation de techniques avancées de l’algorithmique génétique’, Thèse de doctorat (2000)
- [44] Michalewicz, ”Z. Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag. . (1992).
- [45] Cordon, O., F. Herrera et P. Villar. , ” Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base”,. IEEE Transactions on Fuzzy Systems 9(4), 667–674. (2001)
- [46] Hwang, H.S, ”Automatic design of fuzzy rule base for modelling and control using evolutionary programming”,. IEE Proceedings on Control Theory and Applications, pp. 9–16. , (1999).
- [47] Goldberg D.E , ” Genetic Algorithms in Search, Optimisation and Machine Learning, ”, Edition Addison Wesley (1989)
- [48] Ankenbrandt C, ” An Extension to the Theory of Convergence and a Proof of the Time Complexity of Genetic Algorithms”, FOGA90, pp 53-58, (1990)
- [49] R. Cerf, ” Une Théorie Asymptotique des Algorithmes Génétiques ”, Thèse Phd, Université de Montpellier II, Mars (1994)
- [50] Hartman W.& Belew R.K, ” Optimizing An Arbitrary Function is Hard for the Genetic Algorithm, ICGA4 pp 190-195, (1991)
- [51] J. Kennedy et R.C. Eberhart “Particle swarm Optimisation”, Proceedings of the IEEE International Conference on Neural Networks, pp. 1942-1948, IEEE Press.(1995).
- [52] C.W. Reynolds. “flocks, herds and schools: a distributed behavioral model ”, computer journal (1993)

- [54] P.Siarry, J. Dréo, A. Petrowski, E. Taillard , “Meta heuristiques pour l’optimisation difficile”, Eyrolles, 2003.
- [55] Y. Shi, R. Eberhart “ Parameter selection in Particle swarm optimization ” Proceedings of the 7 Annual Conference on Evolutionary Programming,, pp 591-600, LNCS 1447, Springer. (1998)
- [56]] J. Kennedy et R.C. Eberhart, Y. Shi, “ Swarm Intelligence”, Morgan Kaufmann Academic Press,(2001).
- [57] J.Kennedy , “ Swarm Intelligence” Hand book of nature inspired and innovative Computing, pp 187-219, (2006).
- [58] R.C. Eberhart, J, Kennedy “ A new optimiser using particle swarm theory ” , Proceedings of the 6TH International symposium on micromachine and human science, ,pp.3943 IEEE Press. (1995)
- [59] J. Kennedy “Small worlds and mega-minds : effects of neighbourhood topology on particle swarm performance”, Proceedings of the IEEE Congress on Evolutionary computation, pp. 1931-1938, IEEE Press, (1999).
- [60] M.clerk et J.Kennedy, “ The particle swarm : explosion, stability and convergence in multi dimensional complexe space”, IEEE transactions on evolutionary computation, Vol.6, pp. 778-785, (2004).
- [61] R. Krishnan and A. S. Baharwaj, “A review of parameter sensitivity and adaptation in indirect vector controlled induction motor drive system”, IEEE Transactions on Power Electronics, Vol. 6, No. 4, pp. 695–703, October (1991).
- [62] D. J. McKinnon, D. Seyoum and C. Grantham, “Improved parameter characteristics for induction machine analysis and control”, Proceedings of the Australasian Universities Power Engineering Conference, AUPEC2003, 28 Septembre-1 Octobre Christchurch, New Zealand. (2003)
- [63] J. Stephan, M. Bodson, and J. Chiasson, “Real-time estimation of the parameters and

- fluxes of induction motors”, IEEE Trans. Ind. Applicat, vol.30,no.1,pp. 746– 759,Jun (1994).
- [64] P.Vas,“ Sensorless vector and direct torque control : Oxford university press, (1998).
- [65] J. Stephan, M. Bodson, and J. Chiasson, “Real-time estimation of the parameters and fluxes of induction motors”, IEEE Trans. Ind. Applicat, vol. 30,, no.1, pp. 746–759.,Jun. (1994).
- [66] H. Chai and P.Acarney, ‘Induction motor parameter estimation algorithm using spectral analysis’’,Proc.Inst Elect. Eng., Elect. Powe Applicat., vol. 139, no. 3, pp. 165–174. May (1992).
- [67] L.-C. Zai, C. L. DeMarco, T. A. Lipo, “An extended Kalman filter approach to rotor time constant measurement in PWM induction motor drives’’, IEEE Trans Ind.Applicat.,vol.28, pp. 96–104. Feb, (1992).
- [68] Mena, M.; Touhami, O.; Ibtouen, R, “Estimation of the rotor resistance in induction motor by application of the spiral vector theory associate to extended Kalman filter’’, IECON conference of the IEEE, Vol. 1, (2003).
- [69] K. Kristinn et al., ”System identification and control using genetic algorithms”, IEEE Trans. on Systems, Man and Cybernetics, Vol.22, No.5, 1033-1046, (1992).
- [70] A.David ,“ Identification des paramètres du moteur à induction triphasé en vue de sa commande vectorielle’’, thèse de l’université de Laval Canada (2004)
- [71] A.Alhinai, A.Al Badi, “ Parameters identification of 11 phase torus Generator using particle swarm optimization technique’’, Proc. of IEEE Power Engineering General Meeting, Pittsburgh, PA, July 20-24, (2008)