

UNIVERSITE SAAD DAHLEB DE BLIDA

Faculté des sciences de l'ingénieur

Département d'Electronique

MEMOIRE DE MAGISTER

Spécialité : signaux et système

**APPLICATION DE « PARTICLE SWARM OPTIMISATION » POUR
L'IDENTIFICATION PARAMETRIQUE DE LA MACHINE
ASYNCHRONE**

Par

Ahmed OULD-AISSA

Devant le jury composé de :

- | | | |
|---------------------|---|------------|
| - Mr: A. GUESSOUM | Professeur à l'université de Blida | Président |
| - Mr : A. FERDJOUNI | Maître de Conférences à l'université de Blida | Examineur |
| - Mr: M. HADJ SADOK | Maître de Conférences à l'université de Blida | Examineur |
| - Mr : R. TAHMI | Maître de Conférences à l'ENP Alger | Examineur |
| - Mr: M. BOUNEKHLA | Maître de Conférences à l'université de Blida | Rapporteur |

Blida, Juillet 2009

A la mémoire de mon père

A ma famille

A mes amis

RESUME

ملخص

كرس هذا العمل لتطبيق طريقة جديدة للتوفيق "PSO" لحساب قيم عناصر ال آلة لا متزامنة. التقرب المقترح يمكننا من إيجاد في آن واحد قيم العناصر الكهربائية و الميكانيكية للآلة وهذا باستعمال طريقة النموذج المرجعي. والتي تستخدم تيار الإقلاع والتوتر البسيط المناسب كإشارات دخول وخروج.

في النموذج الأول تبيننا التمعنط الخطي للآلة. استخدمت ثلاثة طرق " " لخفض الفارق بين القياس والحساب لإثبات المقاربة. نبين بأن هذه التقنيات المتبناة لتحديد عناصر الآلة لديها نتيجة حسنة.

في النموذج الثاني تم الأخذ بعين الاعتبار التشبع المغناطيسي للآلة. هذه المقاربة استعملت لإيجاد بالإضافة لقيم العناصر الكهربائية و الميكانيكية, ال خاصية المغناطيسية للآلة. الأخطاء ما بين شدة التيار المحسوبة وشدة التيار المقاس ضعيفة جدا مقارنة بالنتائج المتحصل عليها في النموذج الخطي وهذا يثبت فائدة الأخذ بعين الاعتبار عدم خطية التمعنط.

Abstract

This work consists to the application of the new method of optimisation «Particle Swarm Optimisation" for the parametric identification of the asynchronous machine. The proposed approach permits to determine the electric and mechanical parameters of the machine simultaneously; with the reference models method using the current of starting together with the corresponding simple tension like input output signals. A model of the machine that considers the linearity of the inductance is first adopted. Three algorithms of PSO are proposed for the minimisation of the quadratic error between the measured current and the calculated current to validate the approach. One shows that these algorithms are adapted to the identification of the machine. A model taking into account the magnetic saturation in the machine is then developed. The approach is then used to determine, in addition to the electric and mechanical parameters, the magnetic characteristic of the machine. We show that by taking into account this no linearity the difference between the calculated and measured currents is much lower than those obtained by the linear model.

Résumé

Ce travail consiste à l'application de la nouvelle méthode d'optimisation « Particle Swarm Optimisation » pour l'identification paramétrique de la machine asynchrone. L'approche proposée permet de déterminer simultanément les paramètres électriques et mécaniques de la machine à l'aide de la méthode du modèle de référence, en utilisant le courant de démarrage ainsi que la tension simple correspondante comme signaux d'entrée-sortie.

Un modèle de la machine qui considère la linéarité de l'inductance est d'abord adopté. Trois algorithmes de PSO sont proposés pour la minimisation de l'erreur quadratique entre le courant mesuré et le courant calculé pour valider l'approche et vérifier l'identification de la machine. On montre que ces algorithmes sont adaptés à l'identification de la machine.

Un modèle tenant compte de la saturation magnétique dans la machine est ensuite développé. L'approche est alors utilisée pour déterminer, en plus des paramètres électriques et mécaniques, la caractéristique magnétique de la machine. On montre l'intérêt de la prise en compte de cette non linéarité les erreurs entre les courants calculés et les courants mesurés sont beaucoup plus faibles que celles obtenues par le premier modèle.

REMERCIEMENTS

Ce travail de mémoire a été réalisé au sein de l'équipe « Commande Moteur » du laboratoire «LABSET » du département d'électronique.

Je tiens tout d'abord à remercier Monsieur BOUNEKHLA M'hamed, Maître de conférence à l'université de Blida, pour m'avoir accordé sa confiance, pour sa disponibilité, sa bonne humeur, sa rigueur scientifique et sa patience qui m'ont permis de mener ce travail à son terme.

Je tiens à remercier les membres du jury qui ont accepté de juger mon travail de mémoire: Monsieur A. GESSOUM, Professeur à l'université de Blida, Monsieur M. HADJ SADOK Maître de Conférences à l'université de Blida, Monsieur A. FERDJOUNI Maître de Conférences à l'université de Blida, Monsieur R. TAHMI Maître de Conférences à l'ENP Alger.

Je ne pourrais clore ces remerciements sans une pensée très personnelle à ma petite famille pour leurs soutient et leurs sacrifices durant toutes ces longues journées de travail.

TABLE DES MATIERES

RESUME	
REMERCIEMENTS	
TABLE DES MATIERES	
LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX	
INTRODUCTION	8
1. LES METHODES D'OPTIMISATION	11
1.1 Introduction	11
1.2 Classification des problèmes d'optimisation	11
1.3 Les méthodes d'optimisation des problèmes continu non Linéaires	12
1.4 Les méthodes d'optimisation des problèmes discrets (combinatoire) :	14
1.5. Les méta heuristiques	16
1.5.1 Définition	16
1.5.2 Domaines d'application des Méta-heuristiques	16
1.5.3 Résolution d'un problème avec une Méta heuristique	18
1.5.4 Les méta-heuristiques à solution unique	19
1.5.5 Les méta-heuristiques à population de solutions	21
1.6. L'optimisation par essaim particulaire	26
1.6.1 Origines	26
1.6.2 Principe	27
1.6.3 Formalisation	27
1.6.4 Configuration de la méthode	28
1.7. Conclusion	33
2. MODELISATION DE LA MACHINE ASYNCHRONE	34
2.1 Introduction	34
2.2 Modélisation de la machine asynchrone	34
2.2.1 Equations électriques	35
2.2.2 Equations des flux	36
2.2.3 Equation du couple.	36
2.2.4 Modèle de Park	37

2.2.5	Modèle à cinq paramètres électriques	40
2.2.6	Modèle à quatre paramètres électriques	41
2.3.	Technique d'identification	42
2.3.2	Définition d'un problème d'identification de paramètres	44
2.3.3	Identification de paramètres par analyse inverse	44
2.3.4	Identification de paramètres par analyse direct	45
2.4.	Conclusion	46
3.	IDENTIFICATION DES PARAMETRES DE LA MACHINE ASYNCHRONE	47
3.1.	Introduction	47
3.2.	Algorithme PSO	47
3.2.1.	Algorithme de standard	47
3.2.2.	Algorithme PSO avec deux structures de voisinage	49
3.2.3.	Algorithme PSO avec coefficient de suivie	51
3.3.	Identification à l'aide des données simulées	52
3.4.	Identification à l'aide des données mesurées	58
3.5.	Comparaison des résultats	64
3.6.	Conclusion	65
4.	IDENTIFICATION DE LA MACHINE ASYNCHRONE AVEC SATURATION	65
4.1.	Introduction	65
4.2.	Présentation de la saturation	65
4.3.	Modélisation de la machine asynchrone	69
4.4.	Identification	73
4.4.1	Le choix de l'algorithme PSO	73
4.4.2.	Résultats par des données simulés	73
4.4.3.	Relevés expérimentaux	76
4.4.4.	Testes et résultats	77
4.5.	Comparaison	82
4.6.	Conclusion	84
	CONCLUSION	85
	APPENDICE	87
	BIBLIOGRAPHIES	95

LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX

Figure 1.1 La convergence vers le minimum local	14
Figure 1.2 Le déplacement de la $i^{\text{ème}}$ particule	28
Figure 1.3 Exemples sur les structures de voisinage	30
Figure 1.4 La zone probable du déplacement suivant R1 ou R2.	31
Figure 2.1 Représentation des enroulements de la machine asynchrone triphasée	35
Figure 2.2 Repérage angulaire des systèmes d'axes	38
Figure 2.3 Principe d'identification inverse	44
Figure 2.4 Principe d'identification direct	46
Figure 3.1 Algorithme du PSO standard	48
Figure 3.2 Algorithme PSO a deux structures de voisinage	49
Figure 3.3 recherché avec une structure de voisinage globale	50
Figure 3.4 Recherche avec une structure de voisinage géométriquement proche.	50
Figure 3.5 Décalage de la zone probable du déplacement des particules.	52
Figure 3.6 Courant de démarrage et tension simple correspondante Alimentation sinusoïdale	53
Figure 3.7 L'évolution des paramètres du moteur simulé	55
Figure 3.8 L'évolution de la fonction fitness par les trois algorithmes.	58
Figure 3.9 Courant de démarrage et tension simple correspondante au Moteur M1	59
Figure 3.10 Courant de démarrage et tension simple correspondante du Moteur M2	59
Figure 3.11 La superposition des deux courants : mesuré et calculé.	61
Figure 3.12 L'évolution des paramètres du moteur M1	62
Figure 3.13 Superposition des deux courants : mesuré et calculé du moteur M2	63
Figure 3.14 Un zoom sur le courant statorique mesuré du moteur M1	64
Figure 3.15 Un zoom sur le courant statorique mesuré du moteur M2	64
Figure 4.1 Allure du champ dans l'entrefer en régime saturé ; Le fondamental et l'harmonique 3 sont représentés.	67
Figure 4.2 Schéma de mise en évidence de la saturation croisée	68
Figure 4.3 Représentation du courant magnétisant dans le repère dq	71

Figure 4.4 La courbe de magnétisation : simulé et calculé.	74
Figure 4.5 L'évolution des paramètres du moteur simulé	75
Figure 4.6 Courant de démarrage et tension simple correspondante au Moteur M1	76
Figure 4.7 Courant de démarrage et tension simple correspondante au Moteur M2	76
Figure 4.8 L'évolution des paramètres du moteur M1 avec saturation	78
Figure 4.9 Courbe de magnétisation, inductances statique et dynamique du Moteur M1	79
Figure 4.10 Courbe de magnétisation, inductances statique et dynamique du Moteur M2	79
Figure 4.11 La superposition des courants mesuré et calculé pour le moteur M1	80
Figure 4.12 La superposition des courants mesuré et calculé pour le moteur M2	80
Figure 4.13 Erreur absolue entre le courant mesuré et le courant calculé. Moteur M1: Le modèle sans saturation.	81
Figure 4.14 Erreur absolue entre le courant mesuré et le courant calculé. Moteur M1 : Le modèle avec saturation	81
Figure 4.15 Un zoom sur la superposition du courant mesuré et calculé. Moteur M1, en régime permanent sans saturation.	83
Figure 4.16 Un zoom sur la superposition du courant mesuré et calculé. Moteur M1, en régime permanent avec saturation.	83
Tableau 3.1 Identification des paramètres par L'algorithme de PSO standard	54
Tableau 3.2. Identification des paramètres par L'algorithme de PSO à Deux structure de voisinage	56
Tableau 3.3. Identification des paramètres par L'algorithme de PSO avec Coefficient de suivie.	57
Tableau 3.4. Paramètres du moteur M1 estimes par les trois algorithmes	60
Tableau 3.5. Paramètres du moteur M1 estimes par les trois algorithmes	63
Tableau 4.1 Les paramètres obtenu par des données simulées.	74
Tableau 4.2. Paramètres estimés avec saturation des moteurs M1 et M2	77
Tableau 4.3 Comparaison de l'erreur quadratique entre le modèle de magnétisation linéaire et en Saturation.	82

INTRODUCTION

Les machines électriques jouent un rôle fondamental dans les différentes branches de l'industrie. Elles sont utilisées dans les entraînements réglés et principalement comme actionneur dans la grande majorité des servomécanismes. Il est connu, pour ce qui est de la partie technologique, que la productivité et la qualité du produit fini sont en partie dépendantes de la conduite des machines électriques intervenant dans le processus.

Les progrès réalisés en matière de commande et de développements technologiques, tant dans le domaine de l'électronique de puissance que celui de la micro électronique, ont rendu possible l'usage de commandes performantes faisant de la machine asynchrone un concurrent potentiel dans les domaines de la vitesse variable. La mise au point de commandes performantes requière un modèle basé sur des paramètres physiques et une méthode d'identification de la machine asynchrone. Dans les installations de grandes puissances où les essais directs sont onéreux et parfois difficiles à réaliser, la simulation s'impose comme une alternative pour la prédiction des caractéristiques de la machine. Ces caractéristiques peuvent servir à la mise en œuvre d'un système de diagnostic et de surveillance. Dans toutes ces situations, la modélisation et l'identification de la machine ont un impact non négligeable sur la précision des résultats obtenus. Plusieurs approches d'identification sont mises en œuvre pour le cas de la machine asynchrone. Elles se distinguent les unes des autres par la méthode d'identification, les signaux d'entrées sorties et par la technique de minimisation adoptée [62].

- * L'identification classique détermine les paramètres en utilisant le schéma équivalent de la machine asynchrone à l'aide des deux essais classiques suivants: L'essai au synchronisme, l'essai à rotor bloqué.

- * L'identification statique de la machine asynchrone consiste à déterminer les paramètres électriques en utilisant la croissance ou la décroissance du courant continu dans les enroulements de la machine. La réalisation pratique s'effectue par application ou suppression d'une tension continue dans les enroulements tout en maintenant le rotor bloqué.
- * L'identification à l'aide de méthodes numériques de calcul de champs basées sur la méthode des éléments finis peut conduire à des paramètres précis. Cependant, elles nécessitent la connaissance de la géométrie de la machine ainsi que la caractéristique magnétique des matériaux constituant la machine. Ces données ne sont pas toujours disponibles au niveau de l'utilisateur, ce qui restreint le recours à ces méthodes.

Les trois méthodes d'identifications précédentes présentent l'inconvénient de ne pas assurer l'estimation des paramètres mécaniques de la machine. L'obtention de ces paramètres nécessite un essai de ralentissement de vitesse de la machine. La mesure de ralentissement de la vitesse est effectuée à l'aide d'une génératrice tachymétrique ou d'un codeur optique. L'accouplement de ces derniers à la machine n'est pas toujours aisé, ceci constitue un véritable inconvénient.

La méthode d'identification utilisant le régime transitoire des variables d'état mesurables (courants, tensions et vitesse) assure l'identification des paramètres électriques et mécaniques de la machine. Dans le cas de la machine asynchrone triphasée, les mesures sont effectuées au démarrage sous pleine tension et à vide. Dans cette méthode, la mesure de la vitesse constitue un véritable handicap lorsque la machine n'est pas dotée d'un codeur de vitesse (ou génératrice tachymétrique) et surtout lorsque la machine est difficilement accessible, comme c'est le cas des moteurs immergés à des profondeurs de plusieurs centaines de mètres ou des moteurs de chaînes de production difficilement déplaçables.

Compte tenu des problèmes posés précédemment, nous proposons une approche d'identification utilisant uniquement le courant de démarrage et la tension simple correspondante pour estimer simultanément les paramètres

électriques et mécaniques de la machine asynchrone en régime linéaire et saturé. Ainsi, en utilisant la méthode du modèle de référence, les paramètres sont déterminés par minimisation de l'erreur quadratique entre les courants mesurés et les courants calculés à partir du modèle de la machine. Nous proposons une méthode Méta heuristique pour l'optimisation des paramètres adapté au problème posé (domaine des problème contenu, ne nécessite pas de dérivé).

Le premier chapitre, présente un état de l'art sur l'optimisation des problèmes non linéaires avec une classification des différentes méthodes d'optimisation suivant le problème ou la méthode utilisé pour la recherche d'optimum. La méthode de PSO « Particle Swarm Optimisation » est plus détaillée pour sa particularité parmi les derniers méta heuristique utilisé dans l'optimisation des problèmes continu.

Dans le deuxième chapitre, nous commençons par rappeler les équations classiques régissant le fonctionnement de la machine asynchrone triphasée. Nous rappelons le modèle de la machine à cinq puis quatre paramètres électriques, puis nous présentons succinctement différentes techniques d'identification et différentes méthodes d'optimisation.

Dans le troisième chapitre nous mettent en évidence l'application des méta-heuristiques pour optimisé les paramètres de la machine asynchrone en régime linéaire par simulation afin d'assuré la convergence vers un minimum global et en suite par des essai réelles sur deux moteurs asynchrones.

Dans le quatrième chapitre, nous commençons par présenter le phénomène de la saturation puis nous développons un modèle de la machine asynchrone triphasée tenant compte de la saturation, mais se limitant au premier harmonique d'espace. Nous validons ensuite notre modèle et notre approche d'identification à l'aide d'essais réalisés sur plusieurs machines.

Nous terminons ce travail par une conclusion générale et par des perspectives de recherche

CHAPITRE 1

LES METHODES D'OPTIMISATION

1.1 Introduction

Dans ce chapitre nous présentons une classification des méthodes d'optimisation suivant le domaine d'application (Continu ou discret), et suivant la manière de la recherche d'optimum en suite en présentant la nouvelle méta-heuristique « Optimisation par essaim particulaire ».

1.2 Classification des problèmes d'optimisation

Pour tenter de cerner le domaine de l'optimisation, il est nécessaire de faire la distinction entre deux types de problèmes d'optimisation : les problèmes discrets et les problèmes à variables continues. Parmi les problèmes discrets, on trouve le problème du voyageur de commerce : il s'agit de minimiser la longueur de la tournée d'un « voyageur de commerce », qui doit visiter un certain nombre de villes, avant de retourner à la ville de départ. Dans la catégorie des problèmes continus, un exemple classique est celui de la recherche des valeurs à affecter aux paramètres d'un modèle numérique de processus, pour que ce modèle reproduise au mieux le comportement réel observé. En pratique, on rencontre aussi des « problèmes mixtes », qui comportent à la fois des variables discrètes et des variables continues [1].

Des efforts ont longtemps été menés, séparément, pour résoudre ces deux types de problèmes. Dans le domaine de l'optimisation continue, il existe ainsi un arsenal important de méthodes classiques dites d'optimisation globale [1], mais ces techniques sont souvent inefficaces si la fonction objectif ne possède pas une propriété structurelle particulière, telle que la convexité.

Dans le domaine de l'optimisation discrète, un grand nombre d'heuristiques, qui produisent des solutions proches de l'optimum, ont été développées; mais la plupart d'entre elles ont été conçues spécifiquement pour un problème donné.

1.3 Les méthodes d'optimisation des problèmes continu non Linéaires

Les problèmes d'optimisation non linéaires sont divisés en deux grandes classes:

– Problèmes sans contraintes: aucune contrainte d'inégalité ou d'égalité n'est présentée. La forme du problème sans contrainte est :

Minimise $f(x)$. $x \in R^n$, où f est une fonction régulière sur R^n .

– Problèmes avec contraintes, qui impliquent au moins une contrainte d'inégalité ou d'égalité.

Minimise $f(x)$. $x \in R^n$,

Sous les contraintes : $g_i(x) \leq 0 \quad i = 1, 2, \dots, m$

Les problèmes avec contraintes, à leur tour, sont subdivisés en plusieurs classes, selon qu'il y a des contraintes non linéaires, contraintes d'inégalité, et ainsi de suite.

Selon la classification décrite des problèmes d'optimisation, les méthodes d'optimisation sont principalement divisées en celles pour l'optimisation sans contraintes et celles pour l'optimisation sous contraintes. Bien que les problèmes sans contraintes plus simples ne sont pas très fréquents dans les applications, les méthodes d'optimisation sans contrainte jouent un rôle très important : elles sont employées directement pour résoudre des problèmes sans contrainte et indirectement, comme modules, dans beaucoup de méthodes de minimisation sous contraintes.

Les méthodes de résolution numérique des problèmes d'optimisation non linéaire sont, en leur essence, des routines itératives : une méthode ne peut pas typiquement trouver la solution exacte en temps fini. En fait, la méthode génère une suite infini $\{X_t\}$ de solutions approximatives. L'itération suivante X_{t+1} est formée, selon certaines règles, sur la base de l'information locale sur le problème,

collectée à l'itération précédente (De l'objectif et des contraintes en X_t et celles des gradients ou même des dérivées supérieures de ces fonctions en X_t). Notez que la liste d'arguments de X_t est composée des $(t-1)$ parts d'information locale ; en particulier, la liste d'arguments de la toute première règle de recherche X_1 est vide, de sorte que cette fonction soit simplement un vecteur fixe donné par la description de la méthode (point initial).

Il découle du schéma général décrit ci-dessus d'une routine itérative que les méthodes d'optimisation peuvent être classifiées non seulement selon les types de problèmes que les méthodes résolvent, mais également selon le type d'information locale qu'elles emploient. De ce point de vue d'information, les méthodes sont divisées en :

- Routines d'ordre zéro, qui utilisent seulement des valeurs de l'objectif et des contraintes et pas leurs dérivées ;
- Routines du premier ordre, celles qui utilisent les valeurs et les gradients de l'objectif et des contraintes.
- Routines du second ordre, qui utilisent les valeurs, les gradients et les Hessien de l'objectif et des contraintes.

En principe, nous pourrions parler aussi des méthodes d'ordre plus élevé; ces méthodes, cependant, ne sont jamais employées dans la pratique. En effet, pour employer une méthode d'ordre k , on devrait avoir une possibilité de calculer les dérivées partielles de l'objectif et des contraintes jusqu'à l'ordre k . Dans le cas multidimensionnel, ce n'est pas facile même pour $k = 2$ et même quand les fonctions sont données par des expressions analytiques explicites, ce qui n'est pas toujours le cas. Et il y a une explosion de difficultés dans le calcul des dérivées d'ordre supérieur : pour une fonction de n variables, il y a n premières dérivées à calculer, $n(n+1)/2$ dérivées secondes, $n(n+1)(n+2)/(2 \times 3)$ de dérivées troisièmes, etc.; en conséquence, même dans le cas d'un problème d'échelle moyenne avec $n \sim$ quelques dizaines, les difficultés avec la programmation, temps de calcul et la mémoire requise pour traiter les dérivées d'ordre élevées rend excessivement chère l'exploitation de ces dérivées. Par ailleurs, des méthodes d'ordre plus élevé que 2 ne possède aucun avantage théorique, ainsi il n'y a aucune compensation pour l'effort de calcul de ces dérivées.

La difficulté intrinsèque dans l'Optimisation non linéaire est que nous ne pouvons pas nous attendre à ce qu'une méthode numérique d'optimisation approche une solution globalement optimale du problème [2]. Cette difficulté a ses racines en la nature locale d'information disponible aux méthodes. N'importe quelle méthode d'optimisation non linéaire lancée sur le problème en question avec le point de départ dans un petit voisinage du minimum local x_1 , convergera vers x_1 (figure 1.1).

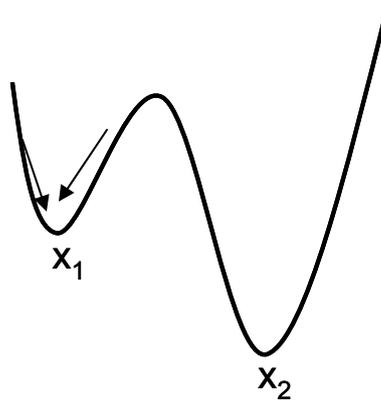


Figure 1.1 La convergence vers le minimum local

1.4 Les méthodes d'optimisation des problèmes discrets (combinatoire) :

L'optimisation combinatoire occupe une place très importante en recherche opérationnelle, en mathématiques discrètes et en informatique. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation [3] et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation combinatoire [4]. Bien que les problèmes d'optimisation combinatoire soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes NP-complet et ne possèdent donc pas à ce jour de solution algorithmique efficace valable pour toutes les données [5].

Etant donnée l'importance de ces problèmes, de nombreuses méthodes de résolution ont été développées en recherche opérationnelle (RO) et en intelligence artificielle (IA). Ces méthodes peuvent être classées sommairement en deux grandes catégories :

- Les méthodes exactes (complètes) qui garantissent la complétude de la résolution
- Les méthodes approchées (incomplètes) qui perdent la complétude pour gagner en efficacité.

Le principe essentiel d'une méthode exacte consiste généralement à énumérer, souvent de manière implicite, l'ensemble des solutions de l'espace de recherche. Pour améliorer l'énumération des solutions, une telle méthode dispose de techniques pour détecter le plus tôt possible les échecs (calculs de bornes) et d'heuristiques spécifiques pour orienter les différents choix. Parmi les méthodes exactes, on trouve la plupart des méthodes traditionnelles (développées depuis une trentaine d'années) telles les techniques de séparation et évaluation progressive (SEP) ou les algorithmes avec retour arrière. Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille moyenne. Malgré les progrès réalisés (notamment en matière de la programmation linéaire en nombres entiers), comme le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème, les méthodes exactes rencontrent généralement des difficultés face aux applications de taille importante.

Les méthodes approchées constituent une alternative très intéressante pour traiter les problèmes d'optimisation de grande taille si l'optimalité n'est pas primordiale. En effet, depuis une dizaine d'années, des progrès importants ont été réalisés avec l'apparition d'une nouvelle génération de méthodes approchées puissantes et générales, souvent appelées méta heuristiques [6][7]. Une méta heuristique est constituée d'un ensemble de concepts fondamentaux (par exemple, la liste tabou et les mécanismes d'intensification et de diversification pour l'heuristique tabou), qui permettent d'aider à la conception de méthodes heuristiques pour un problème d'optimisation. Ainsi les méta-heuristiques sont adaptables et applicables à une large classe de problèmes [8].

Les méta-heuristiques sont représentées essentiellement par les méthodes de voisinage comme le recuit simulé et la recherche tabou, et les algorithmes évolutifs comme les algorithmes génétiques et les stratégies d'évolution. Grâce à

ces méta-heuristiques, on peut proposer aujourd'hui des solutions approchées pour des problèmes d'optimisation classiques de plus grande taille et pour de très nombreuses applications qu'il était impossible de traiter auparavant [9]. On constate, depuis ces dernières années, que l'intérêt porté aux méta-heuristiques augmente continuellement en recherche opérationnelle et en intelligence artificielle.

1.5 Les méta-heuristiques

1.5.1 Définition : Une Méta-heuristiques est une stratégie (règle) de choix pilotant une heuristique (introduite initialement dans le contexte de la méthode taboue) [7].

Une heuristique : est une méthode approchée conçue pour un problème particulier pour produire des solutions non nécessairement optimales avec un temps de calcul raisonnable.

La recherche d'optimum par une heuristique se base sur deux principes:

- Intensification (ou exploitation) : permet d'examiner en profondeur une zone particulière de l'espace de recherche.
- Diversification (ou exploration) : permet d'orienter la recherche vers de nouvelles zones prometteuses dans l'espace de recherche.

Une recherche heuristique efficace nécessite un bon compromis entre intensification et diversification.

1.5.2 Domaines d'application des méta-heuristiques

L'arrivée des méta-heuristiques marque une réconciliation des deux domaines : contenu et discret, en effet, celles-ci s'appliquent à toutes sortes de problèmes discrets, et elles peuvent s'adapter aussi aux problèmes continus. Ces méthodes ont en commun, en outre, les caractéristiques suivantes :

- Elles sont, au moins pour partie, stochastiques : cette approche permet de faire face à l'explosion combinatoire des possibilités ;
- Souvent d'origine discrète (à l'exception notable de Particle Swarm Optimisation).
- Elles ont l'avantage, décisive dans le cas continu, d'être directes, c'est-à-dire qu'elles ne recourent pas au calcul des gradients de la fonction objectif ;

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

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

Ces méthodes ne s'excluent pas mutuellement : en effet, dans l'état actuel de la recherche, il est le plus souvent impossible de prévoir avec certitude l'efficacité d'une méthode donnée, quand elle est appliquée à un problème donné. Un sujet théorique de recherche concerne les possibilités d'analyse systématique des méta-heuristiques du point de vue de la convergence, la complexité, la robustesse et les garanties de qualité.

De plus, la tendance actuelle est l'émergence de méthodes hybrides, qui s'efforcent de tirer parti des avantages spécifiques d'approches différentes en les combinant. La coopération entre méta-heuristiques peut aussi prendre la forme de systèmes multi agents, capables en principe de changer automatiquement d'outil de recherche, en fonction des difficultés rencontrées.

Enfin, dans leur démarche unificatrice de la programmation à mémoire adaptative, E. Taillard remarque que les méta-heuristiques reposent finalement sur un ensemble commun de concepts peu nombreux [10]: la mémoire, qui sauvegarde l'information recueillie par l'algorithme, l'intensification, qui tente d'améliorer la pertinence des informations disponibles au moyen de recherche locale, et la diversification, qui vise à accroître la quantité de ces informations, en explorant de nouvelles régions de l'espace de recherche. Les associations multiples de ces concepts peuvent conduire à une grande variété de méta-heuristiques : la filiation à une métaphore donnée n'est alors plus nécessairement explicite.

On peut souligner une autre richesse des méta-heuristiques : elles se prêtent à toutes sortes d'extensions. Citons, en particulier :

- L'optimisation multi objective, où il s'agit d'optimiser simultanément plusieurs objectifs contradictoires. Il n'existe pas, dans ce cas, un optimum unique;

on cherche, en revanche, une gamme de solutions. Ces solutions peuvent être soumises à l'arbitrage final de l'utilisateur.

- L'optimisation multimodale, où l'on s'efforce de repérer tout un jeu d'optimums globaux ou locaux. Les algorithmes évolutionnaires sont particulièrement bien adaptés à cette tâche, de par leur nature distribuée. Les variantes de type « multi population » exploitent en parallèle plusieurs populations, qui s'attachent à repérer des optimums différents.

- Le recours à des implémentations parallèles. De multiples modes de parallélisme ont été proposés pour les différentes méta-heuristiques. Certaines techniques se veulent générales, d'autres, en revanche, tirent parti de particularités du problème.

- L'optimisation dynamique, qui fait face à des variations temporelles de la fonction objectif au cours de l'optimisation : il faut alors approcher au mieux la solution optimale à chaque pas de temps. Les problèmes d'optimisation dynamique semblent être la cible de choix pour les algorithmes de colonies de fourmis et d'essaims particulières, en particulier quand on ne dispose que d'informations locales. La structure distribuée et le caractère auto organisé de ces algorithmes leur procurent en effet la flexibilité nécessaire pour évoluer dans un environnement changeant.

On voit que ces contextes particuliers requièrent, de la part des méthodes de résolution, des propriétés spécifiques qui ne sont pas présentées dans toutes les méta-heuristiques, de là peut découler un moyen de guidage de l'utilisateur dans le choix d'une méta-heuristique.

1.5.3 Résolution d'un problème avec une méta-heuristique :

Les étapes à suivre globalement sont :

- Analyse et modélisation du problème à traiter
- Choix d'une méta heuristique selon :
 - la qualité des solutions recherchées
 - la disponibilité des connaissances sur le problème
- Adaptation d'un méta heuristique au problème
 - configuration (espace de recherche)

- voisinage
 - traitement des contraintes
- Évaluation de l'algorithme :
- Qualité de la meilleure solution trouvée ou profile de recherche
 - Rapidité, *i.e.* l'effort nécessaire pour trouver une solution (le temps de calcul, le nombre d'itération...).

1.5.4. Les méta heuristiques à solution unique

Les méthodes itératives à solution unique sont toutes basées sur un algorithme de recherche de voisinage qui commence avec une solution initiale, puis l'améliore pas à pas en choisissant une nouvelle solution dans son voisinage [9].

L'intérêt de ces méthodes est qu'il existe une preuve de la convergence asymptotique. Ainsi, lorsque certaines conditions sont vérifiées (schéma de décroissance particulier), on a la garantie d'obtenir la solution optimale. Malheureusement, le paramétrage recommandé par la théorie n'est pas réaliste et il faut beaucoup de temps pour arriver à paramétrer ses méthodes.

1.5.4.1. Les méthodes de descente (Hill Climbing)

Les méthodes de descente sont assez anciennes et doivent leur succès à leur rapidité et leur simplicité [11]. A chaque pas de la recherche, cette méthode progresse vers une solution voisine de meilleure qualité. La descente s'arrête quand tous les voisins candidats sont moins bons que la solution courante; c'est-à-dire lorsqu'un optimum local est atteint. On distingue différents types de descente en fonction de la stratégie de génération de la solution de départ et du parcours du voisinage : la descente déterministe, la descente stochastique et la descente vers le premier meilleur.

1.5.4.2. Le recuit simulé (Simulated Annealing)

Le recuit simulé est une technique d'optimisation de type Monte-Carlo généralisé à laquelle on introduit un paramètre de température qui sera ajusté pendant la recherche. Elle s'inspire des méthodes de simulation en mécanique statistique.

L'analogie historique s'inspire du recuit des métaux en métallurgie : un métal refroidi trop vite présente de nombreux défauts microscopiques, c'est l'équivalent d'un optimum local pour un problème d'optimisation combinatoire. Si on le refroidit lentement, les atomes se réarrangent, les défauts disparaissent, et le métal a alors une structure très ordonnée, équivalente à un optimum global. La méthode du recuit simulé, appliquée aux problèmes d'optimisation, considère une solution initiale et recherche dans son voisinage une autre solution de façon aléatoire. Il est possible de se diriger vers une solution voisine de moins bonne qualité avec une probabilité non nulle. Ceci permet d'échapper aux optima locaux. Au début de l'algorithme, un paramètre T , est déterminé et décroît tout au long de l'algorithme pour tendre vers 0. De la valeur de ce paramètre va dépendre la probabilité d'acceptation des solutions détériorant (plus la température T est élevée, plus cette probabilité sera forte). La performance du recuit simulé dépend, entre autres, de la règle de refroidissement que l'on utilise. Un refroidissement trop rapide mènerait vers un optimum local pouvant être de très mauvaise qualité. Un refroidissement trop lent serait très coûteux en temps de calcul.

1.5.4.3. La recherche Tabou (Tabu search)

La recherche Tabou a été introduite par F. Glover [12] et a montré sa performance sur de nombreux problèmes d'optimisation. Les idées de bases de la recherche Tabou se retrouvent également dans le travail de P. Hansen [13]. Elle n'a aucun caractère stochastique et utilise la notion de mémoire pour éviter de tomber dans un optimum local. Le principe de l'algorithme est le suivant : à chaque itération, le voisinage (complet ou sous-ensemble de voisinage) de la solution courante est examiné et la meilleure solution est sélectionnée. En appliquant ce principe, la méthode autorise de remonter vers des solutions qui semblent moins intéressantes mais qui ont peut être un meilleur voisinage.

Le risque est de cycler entre deux solutions. Pour éviter les phénomènes de cyclage, la méthode a l'interdiction de visiter une solution récemment visitée. Pour cela, une liste taboue contenant les attributs des dernières solutions visitées est tenue à jour. Chaque nouvelle solution considérée enlève de cette liste la solution la plus anciennement visitée.

Cette méthode ne s'arrête pas d'elle-même et il faut déterminer un critère d'arrêt en fonction du temps de recherche que l'on s'octroie. Ce critère peut être, par exemple, l'exécution d'un certain nombre d'itérations ou la non amélioration de la meilleure solution pendant un certain nombre d'itérations. Il existe de nombreuses variantes de recherches Tabou impliquant des techniques plus ou moins sophistiquées pour intensifier ou diversifier la recherche.

1.5.5 Les méta-heuristiques à population de solutions :

Les méthodes d'optimisation à population de solutions améliorent, au fur et à mesure des itérations, une population de solutions. L'intérêt de ces méthodes est d'utiliser la population comme facteur de diversité.

1.5.5.1 La recherche par dispersion (Scatter Search)

La recherche par dispersion est une méthode d'optimisation relativement ancienne décrite par Glover [14]. Cette approche évolutionnaire a pour origine les stratégies de création de règles de décision composée et de contrainte de remplacement.

La recherche par dispersion opère sur un ensemble de solutions appelé l'ensemble de référence, en les combinant pour en créer des nouvelles. À la différence d'une "population" dans les algorithmes génétiques, l'ensemble de référence de solutions dans la recherche par dispersion est relativement petit.

La recherche par dispersion comprend les cinq méthodes suivantes :

- Une méthode de génération de diversification pour produire une collection de solutions diverses, en utilisant une solution d'essai arbitraire (ou solution initiale) comme entrée.

- Une méthode d'amélioration pour transformer une solution initiale en une ou plusieurs solutions d'essai amélioré.

- Une méthode de mise à jour de l'ensemble de référence pour construire et maintenir un ensemble de référence comprenant les meilleures solutions trouvées, organisé pour fournir l'accès efficace par d'autres parties de la méthode. L'incorporation de solutions à l'ensemble de référence est effectuée selon leur qualité ou leur diversité.

- Une méthode de génération d'un sous-ensemble pour opérer sur l'ensemble de référence, pour produire un sous-ensemble de ses solutions comme une base pour créer des solutions combinées.

- Une méthode de combinaison de solutions pour transformer un sous-ensemble donné de solutions produit par la méthode de génération d'un sous-ensemble en un ou plusieurs vecteurs combinés de solutions.

1.5.5.2 Les algorithmes génétiques (Genetic Algorithms)

Il existe de nombreux algorithmes évolutionnaires et nous ne pouvons pas parler des algorithmes génétiques sans mentionner les méthodes de programmation évolutionnaire développée par Fogel [15] et les stratégies évolutionnaires développées indépendamment par Rechenberg [16] et Schwefel [17]. Ils ont contribué énormément à l'intérêt porté aux algorithmes évolutionnaires.

Les algorithmes génétiques sont des méthodes basées sur les mécanismes biologiques tels que les lois de Mendel et sur le principe fondamental (sélection) de Charles Darwin. Holland exposa les principes de ces algorithmes pour permettre aux ordinateurs "d'imiter les êtres vivants en évoluant" pour rechercher la solution à un problème [18]. Les principes fondamentaux des algorithmes génétiques :

- la capacité de représentations élémentaires, comme les chaînes de bits, à coder des structures complexes.

- le pouvoir de transformations élémentaires à améliorer de telles structures.

Plus récemment, Goldberg enrichit la théorie des algorithmes génétiques en s'appuyant sur le parallèle suivant [19]:

- Un individu est lié à un environnement par son code d'ADN,

- Une solution est liée à un problème par son indice de qualité,

- Une bonne solution à un problème donné peut être vue comme un individu susceptible de survivre dans un environnement donné.

Les algorithmes génétiques simulent le processus d'évolution d'une population. A partir d'une population de N solutions du problème représentant des individus, on applique des opérateurs simulant les interventions sur le génome telle que le

croisement (cross-over) ou la mutation pour arriver à une population de solutions de mieux en mieux adaptée au problème. Cette adaptation est évaluée grâce à une fonction coût.

1.5.5.3 Programmation génétique (Genetic Programming)

Récemment, un nouveau paradigme génétique fait son apparition, c'est la Programmation Génétique. L'idée sous-jacente à cette approche est qu'il n'est pas nécessaire d'utiliser des codages linéaires (les simples vecteurs de l'algorithme génétique traditionnel) pour soumettre des programmes à une évolution génétique. L'algorithme consiste à faire évoluer une population constituée d'un grand nombre de programmes. La plupart des algorithmes de programmation génétique travaillent avec une population modélisée sous forme d'arbres. Lors de la recherche, la profondeur de ces arbres peut augmenter fortement en taille. Au départ, la population est constituée de programmes créés aléatoirement. Chaque programme est évalué selon une méthode propre au problème posé. A chaque itération (génération), on classe les programmes en fonction des notes qu'ils ont obtenues, et on crée une nouvelle population, où les meilleurs programmes auront une plus grande chance de survivre ou d'avoir des enfants que les autres. Ce principe est le même qu'en algorithmique génétique classique, mais les opérateurs de croisement et de mutation sont différents. En effet, ils travaillent directement sur la structure d'arbre du programme.

1.5.5.4 Les colonies de fourmis (Ants System)

L'histoire de l'intelligence en essaim remonte à l'étude du comportement de fourmis à la recherche de nourriture au départ de leur nid, par Goss, Deneubourg et leur équipe [20]. En se déplaçant du nid à la source de nourriture et vice-versa (ce qui, dans un premier temps, se fait essentiellement d'une façon aléatoire), les fourmis déposent au passage sur le sol une substance odorante appelée phéromone, ce qui a pour effet de créer une piste chimique. Les fourmis peuvent sentir ces phéromones qui ont un rôle de marqueur de chemin, quand les fourmis choisissent leur chemin, elles ont tendance à choisir la piste qui porte la plus forte concentration de phéromones. Cela leur permet de retrouver le chemin vers leur

nid lors du retour. D'autre part, les odeurs peuvent être utilisées par d'autres fourmis pour retrouver les sources de nourriture détectées par leurs consœurs.

Il a été démontré expérimentalement que ce comportement permet l'émergence des chemins les plus courts entre le nid et la nourriture, à condition que les pistes de phéromones soient utilisées par une colonie entière de fourmis.

Le système de fourmis (Ants System - AS) est une méthode d'optimisation basée sur ces observations proposées par Dorigo [20]. Le système de fourmis a été employé avec succès sur des nombreux problèmes (voyageur de commerce, affectation quadratique, ...) mais les auteurs ont remarqué que l'AS n'a pas un comportement très exploratoire ce qui a conduit les auteurs à utiliser des hybridations du système de fourmis avec des recherches locales.

1.5.5.5 Systèmes immunitaires artificiels (Artificial Immune Systems)

Les systèmes immunitaires artificiels (AIS) sont apparus dans les années 90 et sont inspirés du fonctionnement du système immunitaire humain qui est un mécanisme de défense capable d'apprendre. Beaucoup de propriétés des systèmes immunitaires ont un grand intérêt pour les informaticiens par exemple :

- chaque individu possède son propre système immunitaire avec ses forces et vulnérabilités,
- les molécules qui n'appartiennent pas à l'individu sont reconnues et éliminées,
- le système immunitaire peut détecter et réagir aux pathogènes (appelés antigène *Ag*) que le corps n'a jamais rencontrés,

Un des types de réponse immunitaire est la sécrétion d'anticorps. Les anticorps sont des molécules réceptrices permettant de reconnaître l'antigène et de le bloquer. Cette métaphore est utilisée par les AIS où un anticorps va représenter une solution potentielle au problème.

1.5.5.6 Algorithmes à Estimation de Distribution (EDA)

Les stratégies évolutionnaires mettent en œuvre des opérateurs de mutation et de croisement mais il est difficile pour un utilisateur inexpérimenté de choisir l'opérateur approprié à son problème. Les algorithmes à estimation de distribution reprennent les principes des algorithmes à population mais utilisent des modèles probabilistes à la place d'opérateurs de mutation et de croisement pour construire de nouveaux individus. Le modèle de fonctionnement de l'algorithme est le suivant :

- Génération de la population initiale.
- Sélection des individus prometteurs.
- Estimation de la distribution de ces individus (construction d'un modèle probabiliste).
- Génération de nouvelles solutions à partir du modèle probabiliste.
- Retour en 2 jusqu'à ce que le critère d'arrêt soit satisfait.

Il existe de nombreux algorithmes à estimation de distribution qui peuvent être classés selon le modèle utilisé pour la construction des nouveaux individus :

- Compact GA (produit de distribution de Bernouilli),
- Population Based Incremental Learning (règle de Hebian),
- Univariate Marginal Distribution Algorithm, extended compact GA (produit de distribution marginale),
- Bayesian Optimisation Algorithm (réseau bayésien)

1.6 L'optimisation par essaim particulaire

1.6.1 Origines

L'optimisation par essaim particulaire est une méthode née en 1995 aux Etats Unis sous le nom de Particle Swarm Optimisation (PSO) [21]. Initialement, ses deux concepteurs, Russel Eberhart et James Kennedy, cherchaient à modéliser des interactions sociales entre des «agents» devant atteindre un objectif donné dans un espace de recherche commun, chaque agent ayant une certaine capacité de mémorisation et de traitement de l'information. La règle de base était qu'il ne devait y avoir aucun chef d'orchestre, ni même aucune connaissance par les agents de l'ensemble des informations, seulement des connaissances locales. Un modèle simple fut alors élaboré. Dès les premières simulations, le comportement collectif de ces agents évoquait celui d'un essaim d'êtres vivants convergeant parfois en plusieurs sous essaims vers des sites intéressants. Ce comportement se retrouve dans bien d'autres modèles, explicitement inspirés des systèmes naturels. Ici, la métaphore la plus pertinente est probablement celle de l'essaim d'abeilles, particulièrement du fait qu'une abeille ayant trouvé un site prometteur sait en informer certaines de ses consœurs et que celles ci vont tenir compte de cette information pour leur prochain déplacement . Finalement, le modèle s'est révélé être trop simple pour vraiment simuler un comportement social, mais par contre très efficace en tant qu'outil d'optimisation.

Le fonctionnement de PSO fait qu'elle peut être rangée dans les méthodes itératives (on approche peu à peu de la solution) et stochastiques (on fait appel au hasard). Sous ce terme un peu technique, on retrouve un comportement très sociale qui est: améliorer sa situation en se déplaçant partiellement au hasard et partiellement selon des règles prédéfinies.

Par ailleurs, on notera que cette méta heuristique se démarque des autres méthodes évolutionnaires (typiquement, les algorithmes génétiques) sur deux points essentiels :

- Elle met l'accent sur la coopération plutôt que sur la compétition

– Il n’y a pas de sélection (au moins dans les versions de base), l’idée étant qu’une particule même actuellement médiocre mérite d’être conservée, car c’est peut être justement elle qui permettra le succès futur.

1.6.2 Principe

L’optimisation par essaim de particules repose sur un ensemble d’individus originellement disposés de façon aléatoire et homogène, que nous appellerons dès lors des particules, qui se déplacent dans l’hyperespace de recherche et constituent, chacune, une solution potentielle.

Chaque particule dispose d’une mémoire concernant sa meilleure solution visitée ainsi que la capacité de communiquer avec les particules constituant son entourage.

À partir de ces informations, la particule va suivre une tendance faite, d’une part, de sa volonté à retourner vers sa solution optimale, et d’autre part, de son mimétisme par rapport aux solutions trouvées dans son voisinage.

À partir d’optimums locaux et empiriques, l’ensemble des particules va converger vers la solution optimale globale du problème traité.

1.6.3 Formalisation

Un essaim de particule est caractérisé par :

- N : le nombre de particules ;
- V_{max} : la vitesse maximale d’une particule ;
- La topologie et la taille du voisinage d’une particule.
- $C1$: l’inertie d’une particule ;
- $C2, C3$: les coefficients de confiance qui pondèrent le comportement conservateur (ie. la tendance à retourner vers la meilleure solution visitée) et le panurgisme (ie. la tendance à suivre le voisinage).

Une particule est caractérisée, à l’instant t par :

- $X_i(t)$: sa position dans l’espace de recherche
- $V_i(t)$: sa vitesse
- X_{pbest_i} : la position de la meilleure solution par laquelle elle est passée.
- X_{vbest_i} : la position de la meilleure solution connue de son voisinage.

- $pbest_i$: la valeur de fitness de sa meilleure solution.
- $vbest_i$: la valeur de fitness de la meilleure solution connue du voisinage.

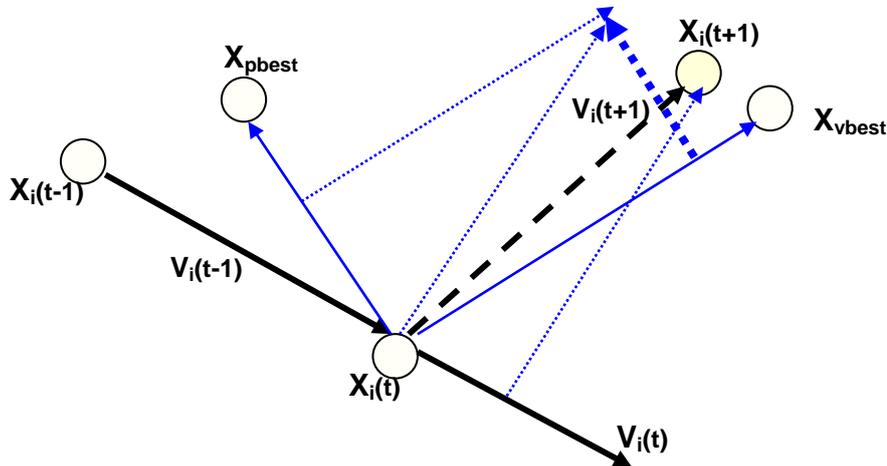


Figure 1.2 Le déplacement de la $i^{\text{ème}}$ particule

L'équation de la vitesse :

$$V_i(t+1) = C1 * V_i(t) + C2 * rand * (X_{pbest} - X_i) + C3 * rand * (X_{vbest} - X_i) \quad (1.1)$$

L'équation du déplacement :

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (1.2)$$

1.6.4 Configuration de la méthode

1.6.4.1 Initialisation de l'essaim et Nombre de particules

La position des particules ainsi que leur vitesse initiale doivent être initialisés aléatoirement selon une loi uniforme sur $[0::1]$. La quantité de particules allouées à la résolution du problème dépend essentiellement de deux paramètres : la taille de l'espace de recherche (D) et le rapport entre les capacités de calcul de la machine et le temps maximum de recherche. Il n'y a pas de règle pour déterminer ce paramètre, faire de nombreux essais permet de se doter de l'expérience nécessaire à l'appréhension de ce paramètre.

1.6.4.2 Topologie du voisinage

La topologie du voisinage défini avec qui chacune des particules va pouvoir communiquer pour la transmission des coordonnées de sa meilleure position déjà visitée. Le choix du voisinage défini le choix de la stratégie de recherche [22] [23].

Il y a deux classes de voisinage : le voisinage globale et le voisinage locale.

➤ Le voisinage globale : C'est le voisinage adopté dans la version de base du PSO, où chaque particule considère la totalité de l'essaim comme voisinage, donc la meilleure particule représente le minimum global de l'essaim et sa position va être transmis vers toutes les autres particules. Le problème avec ce voisinage c'est le risque de converger vers un minimum local si l'essaim n'est pas suffisamment grand [30].

➤ Le voisinage local : Pour éviter qu'une seul particule possède une grande influence sur toutes les autres, le voisinage locale limite la communication entres les particules et définit le canal de communication. Il existe de nombreuses combinaisons proposé [56] dont les suivantes sont les plus utilisées dans les version (Figure 1.1):

- Topologie en étoile : chaque particule est reliée à toutes les autres, ie. l'optimum du voisinage est l'optimum global du l'essaime
- Topologie en anneau : chaque particule est reliée à n particules (en général, $n = 3$), c'est la topologie la plus utilisée
- Topologie en rayon : les particules ne communiquent qu'avec une seule particule centrale
- Topologie de Von Neumann (topologie en cube) : chaque particule communique avec les autres particules placé sur les sommets des arrêts adjacentes.
- Topologie de sous ensembles : chaque sous ensemble désigne une particule pour la communication avec les autres sous ensembles.

Chaque structure possède des propriétés propres à un problème bien déterminé. L'hybridation des structures améliore la convergence de l'algorithme.

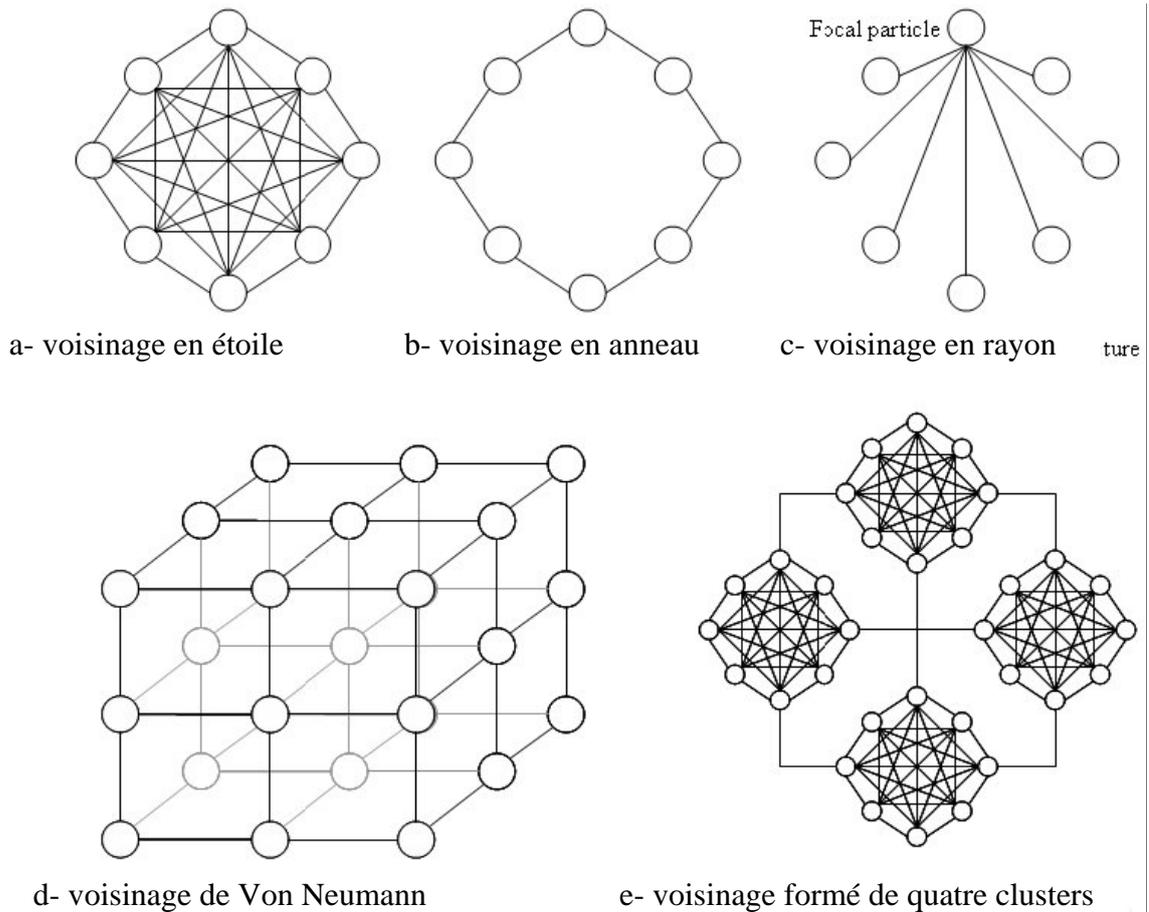


Figure 1.3 Exemples sur les structures de voisinage local

Le voisinage géographique n'est pas nécessairement pertinent car, d'une part, il s'agirait d'un voisinage trop local, et d'autre part car la sociabilité des particules tend à rendre tout voisinage social en voisinage géographique. Enfin, c'est un voisinage très lourd en terme de calculs car nécessitant de recalculer le voisinage à chaque itération.

1.6.4.3 Facteur d'inertie

Le facteur d'inertie $C1$ est très important car c'est lui qui fixe le compromis entre la recherche locale et la recherche globale.

La taille du facteur d'inertie influence directement la taille de l'hyperespace exploré. Les études faites sur ce problème [24] proposent des valeurs entre 0.4 et 1.2. La valeur de $C1$ peut être constante ou dynamique pendant l'évolution du problème.

1.6.4.4 Coefficients de confiance

Les variables de confiance pondèrent les tendances de la particule à vouloir suivre son instinct de conservation ou son panurgisme.

C2 et C3 sont des constantes positives déterminées de façon empirique et suivant la relation :

$$C2 + C3 \leq 4. \quad (1.3)$$

On notera ici un point important, à savoir que les deux coefficients aléatoires sont recalculés pour chaque dimension. Ceci va nous donner l'interprétation géométrique du choix de la nouvelle vitesse.

- Le premier rectangle (Figure 1.4) (plus précisément hyper parallélépipède), appelé $p_proximité$, nous donne toutes les positions possibles pour le terme aléatoire correspondant au retour à proximité de la meilleure performance de la particule. L'une d'elles est choisie au hasard, selon une distribution uniforme.

- De la même manière, une autre position est choisie dans un hyper parallélépipède ($v_proximité$) autour de meilleure performance des voisines.

L'utilisation d'hyper parallélépipèdes implique en effet un biais, puisque le domaine des choix possibles dépend du système de coordonnées. De plus, il s'agit d'une forme assez arbitraire, qui résulte seulement de la facilité de codage d'une distribution uniforme sur un intervalle.

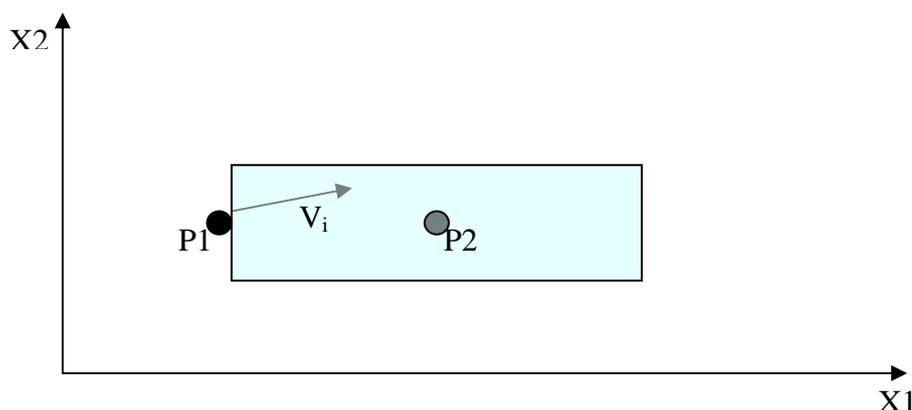


Figure 1.4 La zone probable du déplacement suivant R1 ou R2.

1.6.4.5 Vitesse maximale

Afin d'éviter que les particules ne se déplacent trop rapidement dans l'espace de recherche, passant éventuellement à côté de l'optimum, il peut être nécessaire de fixer une vitesse maximale (notée V_{max}).

1.6.4.6 Notion de confinement :

Lors de l'évolution de l'essaim, il peut arriver qu'une particule sorte de l'espace de recherche initialement défini. C'est sans importance si la valeur de sa position est encore calculable sans « planter » le programme informatique, mais il suffit, par exemple, d'avoir à évaluer la racine d'un nombre négatif ou une division par zéro pour que cela pose problème. Plus généralement, on souhaite souvent rester dans un espace de recherche fini donné. Par conséquent, on ajoute un mécanisme pour éviter qu'une particule sorte de cet espace. Le plus fréquent est le confinement d'intervalle. De plus, on complète souvent le mécanisme de confinement par une modification de la vitesse, soit en remplaçant la composante qui pose problème par son opposée, souvent pondérée par un coefficient inférieur à 1, soit, tout simplement, en l'annulant.

1.6.4.7 Critères d'arrêt

Comme indiqué précédemment, la convergence vers la solution optimale globale n'est pas garantie dans tous les cas de figure même si les expériences dénotent la grande performance de la méthode. De ce fait, il est fortement conseillé de doter l'algorithme d'une porte 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 Max a été atteint.
- La variation de la vitesse est proche de 0
- Le fitness de la solution est suffisant.

1.7 Conclusion

Dans ce chapitre, nous avons présenté les différentes méthodes d'optimisation avec une classification suivant les problèmes traités et suivant les techniques de recherche.

Nous avons étudié les différentes méta heuristiques présentées dans la littérature du domaine d'optimisation, avec une étude détaillée de la nouvelle méta heuristique nommée : Optimisation par Essaim Particulaire « Particle Swarm Optimisation » (PSO) développé initialement pour les problèmes d'optimisation continus.

CHAPITRE 2

MODELISATION DE LA MACHINE ASYNCHRONE

2.1 Introduction

Dans ce chapitre nous présentons la modélisation de la machine asynchrone triphasée ainsi que les techniques d'identification. Après un changement de variables approprié au niveau du vecteur d'état, nous déduisons un modèle de la machine asynchrone à quatre paramètres électriques [60] ayant l'avantage de ne faire apparaître que des paramètres électriques mesurables.

2.2 Modélisation de la machine asynchrone

Le schéma de principe de la machine asynchrone triphasée est représenté par la figure 2.1. L'angle θ repère l'axe de la phase rotorique R_a par rapport à l'axe de la phase statorique de référence S_a . Pour établir les équations régissant le fonctionnement de la machine [61] [62], les hypothèses simplificatrices suivantes sont habituellement utilisées :

- On suppose les circuits magnétiques non saturés et les pertes fer soient négligeables. Les relations entre les flux et les courants sont donc linéaires.
- On considère une densité de courant uniforme dans les sections des conducteurs. L'effet de peau est négligé.
- On ne tient compte que du premier harmonique d'espace de la distribution de la force magnétomotrice de chaque phase du rotor et du stator. L'entrefer étant constant, les inductances propres sont constantes. Les inductances mutuelles sont des fonctions sinusoïdales de l'angle entre les axes des enroulements rotoriques et statoriques.
- On suppose que la machine est symétrique

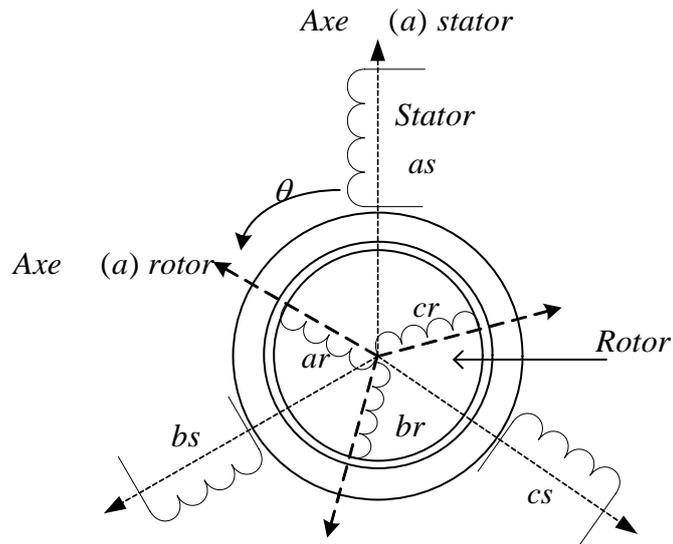


Figure 2.1. Représentation des enroulements de la machine asynchrone triphasée

2.2.1 Equations électriques

Les équations de tension des trois phases statoriques et des trois phases rotoriques sont données par les expressions suivantes :

$$[V_s] = [R_s][I_s] + \frac{d}{dt}[\Phi_s] \quad (2.1)$$

$$[V_r] = [R_r][I_r] + \frac{d}{dt}[\Phi_r] \quad (2.2)$$

Où les indices s et r désignent respectivement les grandeurs relatives au stator et au rotor.

$$[V_s] = \begin{bmatrix} V_{sa} \\ V_{sb} \\ V_{sc} \end{bmatrix}; [I_s] = \begin{bmatrix} I_{sa} \\ I_{sb} \\ I_{sc} \end{bmatrix}; [\Phi_s] = \begin{bmatrix} \Phi_{sa} \\ \Phi_{sb} \\ \Phi_{sc} \end{bmatrix} \quad (2.3)$$

$$[V_r] = \begin{bmatrix} V_{ra} \\ V_{rb} \\ V_{rc} \end{bmatrix}; [I_r] = \begin{bmatrix} I_{ra} \\ I_{rb} \\ I_{rc} \end{bmatrix}; [\Phi_r] = \begin{bmatrix} \Phi_{ra} \\ \Phi_{rb} \\ \Phi_{rc} \end{bmatrix} \quad (2.4)$$

2.2.2 Equations des flux

L'absence de saturation et la limitation au premier harmonique d'espace nous permettent d'écrire les expressions des flux statorique et rotoriques comme suit :

$$\begin{bmatrix} [\Phi_s] \\ [\Phi_r] \end{bmatrix} = \begin{bmatrix} [L_s] & [M_{sr}] \\ [M_{rs}] & [L_r] \end{bmatrix} \cdot \begin{bmatrix} [I_s] \\ [I_r] \end{bmatrix} \quad (2.5)$$

Les quatre sous matrices d'inductances s'écrivent :

$$[L_s] = \begin{bmatrix} l_s & m_s & m_s \\ m_s & l_s & m_s \\ m_s & m_s & l_s \end{bmatrix} \quad [L_r] = \begin{bmatrix} l_r & m_r & m_r \\ m_r & l_r & m_r \\ m_r & m_r & l_r \end{bmatrix} \quad (2.6)$$

$$[M_{rs}] = [M_{sr}] = m_{sr} \begin{bmatrix} \cos(\theta) & \cos\left(\theta + \frac{2\pi}{3}\right) & \cos\left(\theta - \frac{2\pi}{3}\right) \\ \cos\left(\theta - \frac{2\pi}{3}\right) & \cos(\theta) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ \cos\left(\theta + \frac{2\pi}{3}\right) & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos(\theta) \end{bmatrix} \quad (2.7)$$

Les différentes inductances désignent:

l_s : inductance propre d'une phase statorique

l_r : inductance propre d'une phase rotorique.

m_s : inductance mutuelle entre phases statoriques.

m_r : inductance mutuelle entre phases rotoriques.

m_{sr} : inductance mutuelle entre stator et rotor.

θ : écart angulaire entre stator et rotor.

2.2.3 Equation du couple.

L'expression du couple électromagnétique peut être obtenue à partir de la dérivée de la coénergie magnétique par rapport à l'angle. La machine étant à structure lisse, le couple s'écrit :

$$C_{em} = [I_s]^t \frac{\partial [M_{sr}(\theta)]}{\partial \theta} [I_r] \quad (2.8)$$

I_s et I_r désignent respectivement les courants statoriques et rotoriques.

2.2.4 Modèle de Park

Les termes figurant dans les équations des flux (2.7) sont fonction de l'angle θ , ce qui complique considérablement l'étude des régimes dynamiques. Pour pallier ce problème, on opère sur l'ensemble des grandeurs (tensions, courants et flux) un changement de base qui simplifie notablement les relations. Ce changement de base est appelé transformation de Park. Ainsi, la machine sera régie par un système d'équations à coefficients constants.

2.2.4.1 Transformation de Park

La transformation de Park implique la transformation des enroulements statorique et rotorique de la machine originelle en des enroulements équivalents du point de vue électrique et magnétique disposés selon les axes en quadrature d et q comme le montre la figure 2.2. Cette transformation a pour effet de rendre les inductances mutuelles du modèle de la machine indépendante de la rotation, c'est à dire de l'angle θ . La transformation de Park est donnée par l'expression suivante:

$$P(\theta) = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta - \frac{4\pi}{3}\right) \\ -\sin(\theta) & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta - \frac{4\pi}{3}\right) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \quad (2.9)$$

Le changement de variables relatif aux grandeurs X_i (courants, tensions ou flux) est défini par l'expression suivante:

$$\begin{bmatrix} xd \\ xq \\ xo \end{bmatrix} = P \begin{bmatrix} xa \\ xb \\ xc \end{bmatrix} \quad (2.10)$$

X_o représente la grandeur homopolaire.

La matrice inverse P^{-1} est telle que $P^{-1} = {}^tP$.

$$P^{-1}(\theta) = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & \frac{1}{\sqrt{2}} \\ \cos\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta - \frac{2\pi}{3}\right) & \frac{1}{\sqrt{2}} \\ \cos\left(\theta - \frac{4\pi}{3}\right) & -\sin\left(\theta - \frac{4\pi}{3}\right) & \frac{1}{\sqrt{2}} \end{bmatrix} \quad (2.11)$$

d'où les transformations inverse des variables :

$$\begin{bmatrix} xa \\ xb \\ xc \end{bmatrix} = P^{-1} \begin{bmatrix} xd \\ xq \\ xo \end{bmatrix} \quad (2.12)$$

2.2.4.2 Equations de la machine dans le modèle de Park

On applique la transformation de Park aux courants, aux tensions et aux flux. Un changement de variable faisant intervenir l'angle entre l'axe des enroulements et les axes d et q.

Les angles θ_s et θ_r qui interviennent respectivement pour les grandeurs statoriques et rotoriques sont représentés sur la figure 2.2 .

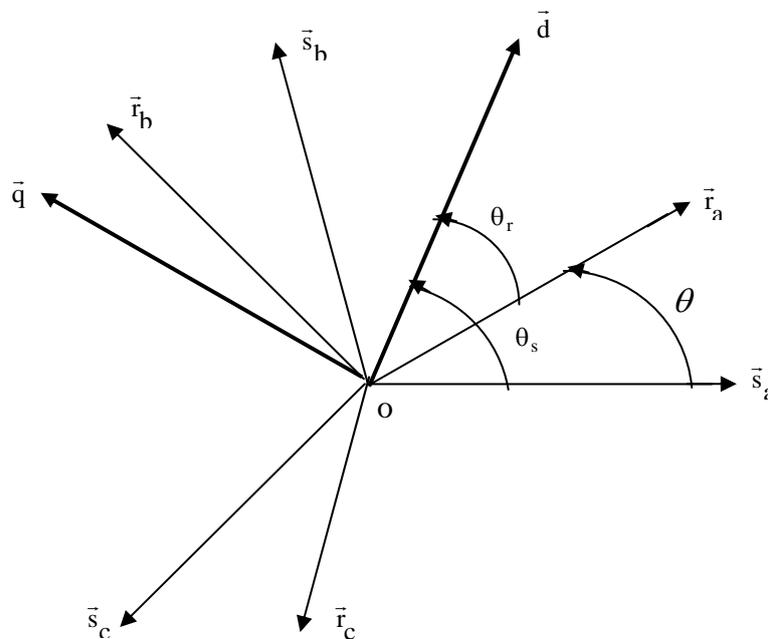


Figure 2.2. Repérage angulaire des systèmes d'axes

2.2.4.3 Equations de tension

Dans un repère dq tournant à la vitesse ω_a par rapport au stator, les équations (2.1) et (2.2) s'écrivent après application de la transformation de Park aux différentes grandeurs comme suit:

$$\left\{ \begin{array}{l} V_{ds} = R_s I_{ds} + \frac{d}{dt} \Phi_{ds} - \omega_a \Phi_{qs} \\ V_{qs} = R_s I_{qs} + \frac{d}{dt} \Phi_{qs} + \omega_a \Phi_{ds} \\ V_{dr} = R_r I_{dr} + \frac{d}{dt} \Phi_{dr} - (\omega_a - \omega_m) \Phi_{qr} \\ V_{qr} = R_r I_{qr} + \frac{d}{dt} \Phi_{qr} + (\omega_a - \omega_m) \Phi_{dr} \end{array} \right. \quad (2.13)$$

ω_a et ω_m désignent respectivement la vitesse du repère dq et la vitesse du rotor par rapport au stator. Selon l'application envisagée, le repère dq peut être lié au stator ($\omega_a=0$), rotor ($\omega_a=\omega_m$) ou au champs tournant ($\omega_a = \omega_s$).

2.2.4.4 Equations des flux

Les différentes expressions des flux statoriques et rotoriques s'écrivent dans le modèle de Park sous la forme suivante:

$$\left\{ \begin{array}{l} \Phi_{ds} = L_s I_{ds} + L_m I_{dr} \\ \Phi_{qs} = L_s I_{qs} + L_m I_{qr} \\ \Phi_{dr} = L_r I_{dr} + L_m I_{ds} \\ \Phi_{qr} = L_r I_{qr} + L_m I_{qs} \end{array} \right. \quad (2.14)$$

L_s , L_r et L_m ont pour expressions :

$$\left\{ \begin{array}{l} L_s = l_s - m_s \\ L_r = l_r - m_r \\ L_m = \frac{3}{2} m_{sr} \end{array} \right. \quad (2.15)$$

les différentes inductances désignent:

- L_s : inductance cyclique propre d'une phase statorique
 L_r : inductance cyclique propre d'une phase rotorique.
 L_m : inductance mutuelle cyclique entre stator et rotor.

2.2.4.5 Equation du couple

Dans le modèle de Park, l'expression du couple s'écrit de la manière suivante:

$$C_e = P * L_m (I_{qs} * I_{dr} - I_{ds} * I_{qr}) \quad (2.16)$$

P désigne le nombre de paire de pôles de la machine

2.2.5 Modèle à cinq paramètres électriques

Dans un repère lié au stator, les équations électriques de la machine sont déduites des équations (2.13) dans lesquelles ω_a est égale à zéro. En ajoutant l'équation mécanique, la machine sera entièrement décrite par le système suivant:

$$\begin{cases} V_{ds} = R_s I_{ds} + L_s \frac{dI_{ds}}{dt} + L_m \frac{dI_{dr}}{dt} \\ V_{qs} = R_s I_{qs} + L_s \frac{dI_{qs}}{dt} + L_m \frac{dI_{qr}}{dt} \\ V_{dr} = R_r I_{dr} + L_r \frac{dI_{dr}}{dt} + L_m \frac{dI_{ds}}{dt} + P\Omega L_r I_{qr} + P\Omega L_m I_{qs} \\ V_{qr} = R_r I_{qr} + L_r \frac{dI_{qr}}{dt} + L_m \frac{dI_{qs}}{dt} - P\Omega L_r I_{dr} - P\Omega L_m I_{ds} \\ J \frac{d\Omega}{dt} = PL_m (I_{qs} I_{dr} - I_{ds} I_{qr}) - f_r \Omega \end{cases} \quad (2.17)$$

En réarrangeant les équations précédentes de façon à faire apparaître le vecteur d'état $[I_{ds} \ I_{qs} \ I_{dr} \ I_{qr} \ \Omega]^t$, nous obtenons les équations suivantes:

$$\begin{cases} \frac{dI_{ds}}{dt} = -\frac{L_r R_s}{A} I_{ds} + \frac{P\Omega L_m^2}{A} I_{qs} + \frac{L_m R_r}{A} I_{dr} + \frac{P\Omega L_m L_r}{A} I_{qr} + \frac{L_r V_{ds}}{A} \\ \frac{dI_{qs}}{dt} = -\frac{P\Omega L_m^2}{A} I_{ds} - \frac{L_r R_s}{A} I_{qs} - \frac{P\Omega L_m L_r}{A} I_{dr} + \frac{L_m R_r}{A} I_{qr} + \frac{L_r V_{qs}}{A} \\ \frac{dI_{dr}}{dt} = \frac{L_m R_s}{A} I_{ds} - \frac{P\Omega L_m L_s}{A} I_{qs} - \frac{L_s R_r}{A} I_{dr} - \frac{P\Omega L_s L_r}{A} I_{qr} - \frac{L_m V_{ds}}{A} \\ \frac{dI_{qr}}{dt} = +\frac{P\Omega L_m L_s}{A} I_{ds} + \frac{L_m R_s}{A} I_{qs} + \frac{P\Omega L_s L_r}{A} I_{dr} - \frac{L_s R_r}{A} I_{qr} - \frac{L_m V_{qs}}{A} \\ \frac{d\Omega}{dt} = \frac{PL_m}{J} (I_{qs} I_{dr} - I_{ds} I_{qr}) - \frac{f_r}{J} \Omega \end{cases} \quad (2.18)$$

où $A = L_s L_r - L_m^2$

L'équation (2.18), montre que le fonctionnement de la machine dépend de cinq paramètres électriques [R_s R_r L_s L_r L_m] et de trois paramètres mécaniques [J f_r C_s]. Le rotor d'une machine asynchrone à cage n'est pas accessible directement à l'expérimentation et il est impossible de mesurer séparément les vraies valeurs de R_r L_r , et L_m qui figurent dans le système d'équations (2.18). Après un changement de variables du vecteur d'état, nous allons déduire un modèle à quatre paramètres électriques mesurables.

Le système (2.18) semble surdimensionné. En effet, nous avons constaté que les différentes techniques d'optimisation que nous exposerons dans le chapitre suivant, ne convergent pas lorsque l'on désire identifier simultanément les 5 paramètres électriques de la machine. Cette convergence n'est obtenue que si l'on adopte une condition supplémentaire sur les paramètres électriques, comme par exemple celle utilisée par P.L. ALGER [63] qui admet l'égalité des inductances de fuite au rotor et au stator. Nous adoptons alors un changement de variables du vecteur d'état pour déduire un modèle à quatre paramètres électriques mesurables.

2.2.6 Modèle à quatre paramètres électriques

En fonctionnement normal de la machine, les tensions V_{dr} et V_{qr} sont nulles, ce qui a pour conséquence de réduire le nombre de paramètres à quatre lorsqu'on effectue le changement de variables suivant:

$$I_{dr} = \frac{L_m}{L_r} I'_{dr} \quad \text{et} \quad I_{qr} = \frac{L_m}{L_r} I'_{qr}$$

En introduisant le coefficient de dispersion $\sigma = 1 - \frac{L_m^2}{L_s L_r}$, les constantes de temps rotorique $T_r = \frac{L_r}{R_r}$ et statorique $T_s = \frac{L_s}{R_s}$, les équations (2.17) deviennent:

$$\begin{cases}
V_{ds} = R_s I_{ds} + L_s \frac{dI_{ds}}{dt} + (1-\sigma)L_s \frac{dI'_{dr}}{dt} \\
V_{qs} = R_s I_{qs} + L_s \frac{dI_{qs}}{dt} + (1-\sigma)L_s \frac{dI'_{qr}}{dt} \\
V_{dr} = \frac{I'_{dr}}{T_r} + \frac{dI'_{dr}}{dt} + \frac{dI_{ds}}{dt} + P\Omega I'_{qr} + P\Omega I_{qs} \\
V_{qr} = \frac{I'_{qr}}{T_r} + \frac{dI'_{qr}}{dt} + \frac{dI_{qs}}{dt} - P\Omega I'_{dr} - P\Omega I_{ds} \\
J \frac{d\Omega}{dt} = P(1-\sigma)L_s (I_{qs} I'_{dr} - I_{ds} I'_{qr}) - f_r \Omega
\end{cases} \quad (2.19)$$

Et, en faisant apparaître le vecteur d'état $[I_{ds} \ I_{qs} \ I'_{dr} \ I'_{qr} \ \Omega]^t$, nous obtenons les équations suivantes:

$$\begin{cases}
\frac{dI_{ds}}{dt} = -\frac{1}{\sigma T_s} I_{ds} + \frac{1-\sigma}{\sigma} P\Omega I_{qs} + \frac{1-\sigma}{\sigma T_r} I'_{dr} + \frac{1-\sigma}{\sigma} P\Omega I'_{qr} + \frac{V_{ds}}{\sigma L_s} \\
\frac{dI_{qs}}{dt} = -\frac{1-\sigma}{\sigma} P\Omega I_{ds} - \frac{1}{\sigma T_s} I_{qs} - \frac{1-\sigma}{\sigma} P\Omega I'_{dr} + \frac{1-\sigma}{\sigma T_r} I'_{qr} + \frac{V_{qs}}{\sigma L_s} \\
\frac{dI'_{dr}}{dt} = \frac{1}{\sigma T_s} I_{ds} - \frac{P\Omega}{\sigma} I_{qs} - \frac{1}{\sigma T_r} I'_{dr} - \frac{P\Omega}{\sigma} I'_{qr} - \frac{V_{ds}}{\sigma L_s} \\
\frac{dI'_{qr}}{dt} = \frac{P\Omega}{\sigma} I_{ds} + \frac{1}{\sigma T_s} I_{qs} + \frac{P\Omega}{\sigma} I'_{dr} - \frac{1}{\sigma T_r} I'_{qr} - \frac{V_{qs}}{\sigma L_s} \\
\frac{d\Omega}{dt} = \frac{P}{J} (1-\sigma)L_s (I_{qs} I'_{dr} - I_{ds} I'_{qr}) - \frac{f_r}{J} \Omega
\end{cases} \quad (2.20)$$

L'équation (2.20), montre que le fonctionnement de la machine dépend de quatre paramètres électriques $[\sigma \ T_s \ L_s \ T_r]$ et de deux paramètres mécaniques $[J \ f_r]$.

2.3. Technique d'identification

L'identification consiste à rechercher des modèles mathématiques de systèmes à partir de données expérimentales et de connaissances disponibles a priori [46]. Ces modèles doivent fournir une approximation fidèle du comportement du système physique sous-jacent dans le but d'estimer des paramètres physiques ou de concevoir des algorithmes de simulation, de prévision, de surveillance ou de commande. La démarche classique consiste à formaliser les connaissances disponibles a priori, à recueillir des données expérimentales, puis à estimer la structure, les paramètres et les incertitudes d'un modèle, enfin à valider (ou invalider) celui-ci. Ce champ thématique trouve ses applications dans des domaines très variés.

L'utilisation de données expérimentales pour déterminer les paramètres de modèles mathématiques de systèmes dynamiques représente un sujet classique de l'automatique. Les modèles issus de l'identification sont appelés modèles boîte noire, lorsqu'ils sont peu, voire pas du tout, inspirés de la connaissance physique sur le procédé considéré. Durant les quatre dernières décennies, l'identification de modèles boîte noire a connu un développement considérable tant sur le plan de la proposition de nouvelles techniques que sur le plan des applications [64].

En effet, le formidable développement des calculateurs numériques a rendu l'utilisation des modèles à temps discret de plus en plus courante, non seulement en raison de la nature discrète des données acquises, mais surtout à cause de la facilité de l'implantation de l'algorithme d'identification puis de commande ou de diagnostic préventif développés à partir du modèle identifié.

Pour identifier un modèle à temps continu, deux approches principales sont envisageables dans le domaine temporel : l'approche directe et l'approche indirecte. Toutes deux utilisent des données d'entrée/sortie échantillonnées. L'approche indirecte consiste dans un premier temps, à déterminer un modèle à temps discret, puis à convertir ce dernier en un modèle à temps continu. Les propriétés statistiques de ces estimateurs (biais, variance, convergence) sont bien connues et expliquent en grande partie l'attrait pour cette stratégie. Cette approche indirecte fondée sur l'estimation initiale d'un modèle à temps discret fait, en général, appel à des algorithmes d'optimisation itératifs très coûteux en termes de calculs sans garantie de convergence vers l'optimum global. En effet, pour la plupart de ces algorithmes, la procédure d'initialisation conditionne la convergence vers l'optimum global. Les approches directes, ne souffrent pas de cet inconvénient.

Les techniques d'identification de modèles paramétriques linéaires à temps continu reposent principalement sur la minimisation d'un critère fondé soit sur une erreur de sortie, soit sur une erreur d'équation nécessitant l'utilisation d'une transformation linéaire couplée à une méthode issue des moindres carrés. De nombreuses méthodes de type erreur d'équation ont été proposées au cours des trente dernières années [62].

Dans ce contexte, on va essentiellement s'intéresser à l'identification des systèmes à l'aide de modèles à représentation continue. Deux catégories d'algorithmes sont utilisables, que l'on classe suivant la nature des résidus en erreur équation ou en erreur de sortie.

2.3.1 Définition d'un problème d'identification de paramètres :

Soit M un vecteur de $\mathbb{R}^{N_{\text{Mes}}}$ contenant N_{Mes} observations d'un système physique et F une application de $\mathbb{R}^{N_{\text{par}}}$ dans $\mathbb{R}^{N_{\text{Mes}}}$. On suppose que F est obtenue à partir d'une modélisation du système physique observé. Le but est de trouver le jeu de paramètres λ appartenant à $\mathbb{R}^{N_{\text{par}}}$ tel que $F(\lambda) = M$.

2.3.2 Identification de paramètres par analyse inverse

L'intérêt essentiel de ces méthodes, fondées sur l'erreur d'équation, est la formulation d'une solution explicite qui conduit à un optimum unique. Toutefois ces techniques nécessitent l'estimation des dérivées successives des signaux d'entrée/sortie [65].

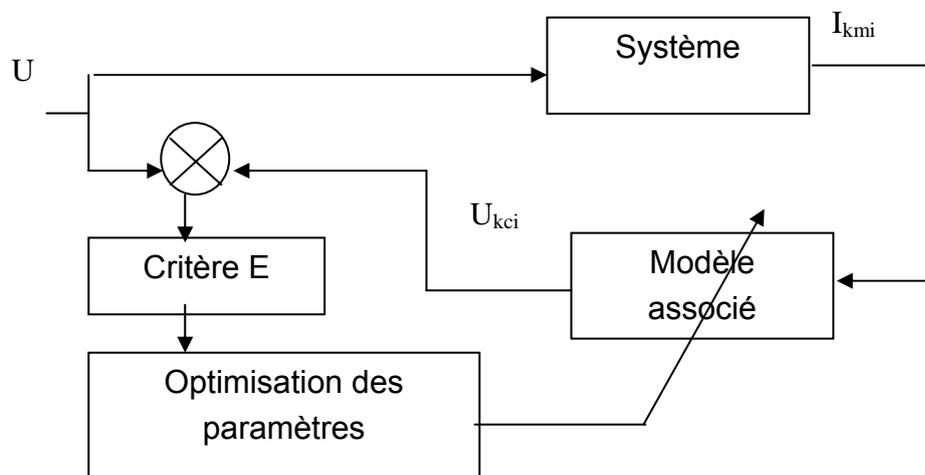


Figure 2.3. Principe d'identification inverse

Les problèmes inverses doivent intégrer des données expérimentales (figure 2.3). Il est nécessaire dans ce cas d'avoir une idée du degré de confiance que l'on accorde aux mesures expérimentales et de connaître la façon dont une erreur de mesure affecte les valeurs des paramètres identifiés.

Le problème inverse est exprimé sous une forme relaxée, et l'on cherche à minimiser une distance entre les données issues d'un modèle et les mesures expérimentales. La résolution du problème inverse tel que défini ci-dessus implique donc que l'application F soit bijective. Une condition nécessaire de bijectivité de l'application F est que $NbMes=Npar$. Généralement, on a plutôt $NbMes \gg Npar$ en identification de paramètres. Donc, une condition nécessaire d'existence et d'unicité de la solution du problème d'identification de paramètres est que $M \in Im(F)$. En pratique, lorsque la mesure expérimentale est entachée d'erreurs de mesure et lorsque l'erreur de modélisation est non négligeable, il est fréquent que $M \notin Im(F)$. Les problèmes d'identification de paramètres sont généralement exprimés sous la forme d'un problème d'optimisation (au sens des moindres carrés) afin de ne pas avoir à considérer l'inverse de l'opérateur F .

Les algorithmes à erreur équation ne sont utilisables en pratique qu'avec des modèles du type équation différentielle à coefficients constants. Pour de tels modèles, il faut exprimer la sortie du système linéairement par rapport à ses paramètres (L.P.). Cette propriété de linéarité permet alors d'utiliser la méthode des moindres carrés dont l'intérêt essentiel est de fournir une expression analytique de l'estimée des paramètres. Malheureusement, on démontre que pour tout modèle L.P. dont le régresseur dépend directement (ou indirectement par filtrage) des valeurs de la sortie, les résidus sont du type erreur équation et en conséquence l'estimateur est biaisé.

Comme en général les machines électriques ne se ramènent pas à des équations différentielles à coefficients constants mais plutôt à des systèmes différentiels non linéaires, on comprendra que cette méthodologie d'identification ne soit pas vraiment adaptée au problème envisagé.

2.3.3 Identification de paramètres par analyse direct :

Dans les problèmes directs, les solutions analytiques ou numériques sont trouvées pour les équations différentielles ordinaires ou partielles avec des conditions initiales et aux limites connue et les constantes dans les équations connues. Le point important dans les problèmes directs, les variables dépendantes dans les équations sont calculés comme fonctions de temps et/ou position. Les mesures n'entrent pas dans la solution, excepté peut-être pour fournir des conditions de la limite (Figure 2.4).

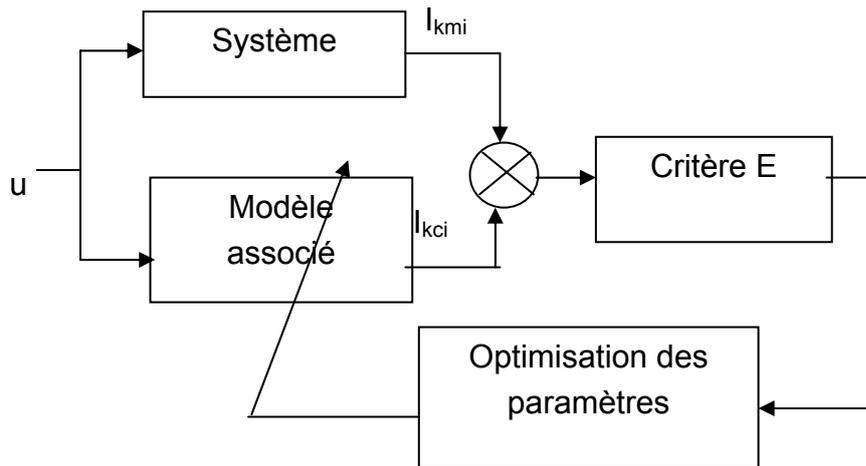


Figure 2.4 Principe d'identification direct

Cette méthode est aussi connue sous l'appellation « Méthode de l'erreur de sortie » ou de méthode du modèle. Elle repose sur la minimisation de l'énergie de l'erreur de sortie. Ce critère se distingue de celui fondé sur une erreur d'équation par sa signification physique et sa sensibilité plus importante aux erreurs de structure de modèles et aux valeurs des paramètres. Elle se différencie également par sa mise en œuvre plus délicate due à la non linéarité de l'erreur de sortie par rapport aux paramètres. Elle consiste à exploiter les propriétés locales du critère au voisinage d'un point courant dans l'espace des paramètres pour en déduire un point suivant meilleur au sens du critère à minimiser.

- Elle permet l'identification de systèmes multi entrées mono sortie représentés par une structure de modèles.
- Le problème principal des méthodes à erreur de sortie est l'existence possible de plusieurs minima locaux vers lesquels peut converger l'algorithme d'optimisation. L'initialisation des algorithmes joue, en général, un rôle important sur la convergence et il est nécessaire de faire appel à des procédures particulières afin d'éviter les optimums locaux.

2.4. Conclusion

Dans ce chapitre on a développé un modèle proche de la réalité physique de la machine asynchrone et qui repose sur des paramètres réels. Les paramètres de ce modèle seront identifier par la technique d'identification directe (erreur de sortie) dans le chapitre suivant.

CHAPITRE 3

IDENTIFICATION DES PARAMETRES DE LA MACHINE ASYNCHRONE

3.1. Introduction

Dans ce chapitre, nous déterminons simultanément les paramètres électriques et mécaniques de la machine en utilisant un modèle dynamique ayant comme vecteur d'état, les courants statoriques, les courants rotoriques et la vitesse. Comme signaux d'entrées-sorties, nous n'utilisons que le courant d'alimentation d'une phase statorique et la tension simple correspondante. Nous explorons l'espace paramétrique de la machine à l'aide de trois variantes de l'algorithme PSO (standard, deux structures de voisinage, Avec coefficient de suivi). Nous validons ensuite les résultats obtenus par simulation par des relevés expérimentaux effectués sur deux moteurs asynchrones triphasés.

3.2. L'algorithme d'optimisation par essaim particulaire (PSO)

Nous allons présenter trois algorithmes de PSO :

- Algorithme PSO standard
- Algorithme PSO avec deux structures de voisinage.
- Algorithme PSO avec coefficient de suivi.

3.2.1. Algorithme PSO standard

Les équations de la vitesse et de déplacement définies dans les PSO standard sont données par [20] :

$$V_i(t+1) = C1 * V_i(t) + C2 * rand * (X_{pbest} - X_i) + C3 * rand * (X_{vbest} - X_i) \quad (3.1)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (3.2)$$

C1 = 0.689 : coefficient de décélération.

C2 = 1.426 : coefficient de confiance en son record.

$C3 = 1.426$: coefficient de confiance en voisinage.

r : Une valeur aléatoire varie entre 0 et 1, recalculé pour chaque dimension.

X_{pbest} : La meilleure position visitée par la particule.

X_{vbest} : La meilleure position transmise par le voisinage.

$N = 40$ particules : La taille de l'essaim.

$K = 7$: Nombre des particules dans le voisinage qui est formé par l'ensemble de l'essaim.

Chaque particule transmet sa meilleure position déjà visitée aux k particules choisi aléatoirement dans le voisinage. Cette méthode implique qu'il y a une probabilité qu'une particule ne reçoit aucune information du voisinage le fait qu'elle n'est jamais sélectionnée. Cette méthode favorise l'exploitation (recherche locale) et elle diminue l'exploration du domaine de recherche (recherche globale).

L'organigramme de la méthode est présenté par la figure 3.1

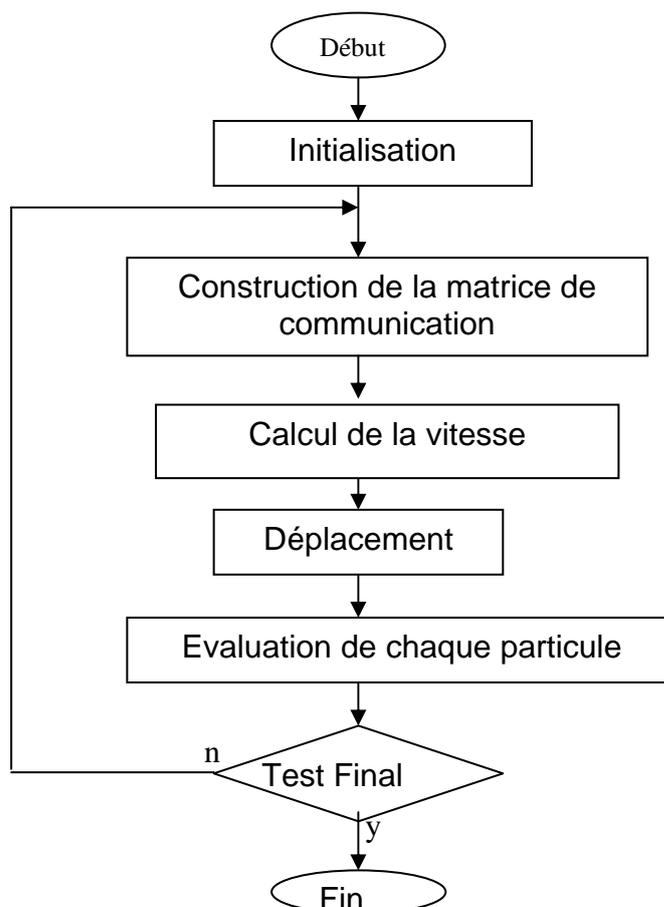


Figure 3.1 Algorithme du PSO standard

3.2.2. Algorithme PSO avec deux structures de voisinage

Cette méthode basé sur une recherche globale et une recherche locale. L'algorithme change le type de recherche chaque 20 itérations, il exécute 15 itérations avec recherche locale et 5 itérations avec recherche globale (Figure 3.2)

Cette technique est illustrée pour deux variables et six particules.

- La recherche globale est présentée dans Figure 3.3 où chaque particule demande information aléatoirement de trois autres particules de son voisinage formé par la totalité d l'essaim. Donc on a plus de probabilité d'avoir l'information d'une particule éloignée, ce qui va augmenté le terme $(X_{bv} - X)$ qui intervient dans le calcul de la vitesse. Cette augmentation permet l'exploration des nouveaux sites ce qui va favoriser la recherche globale.

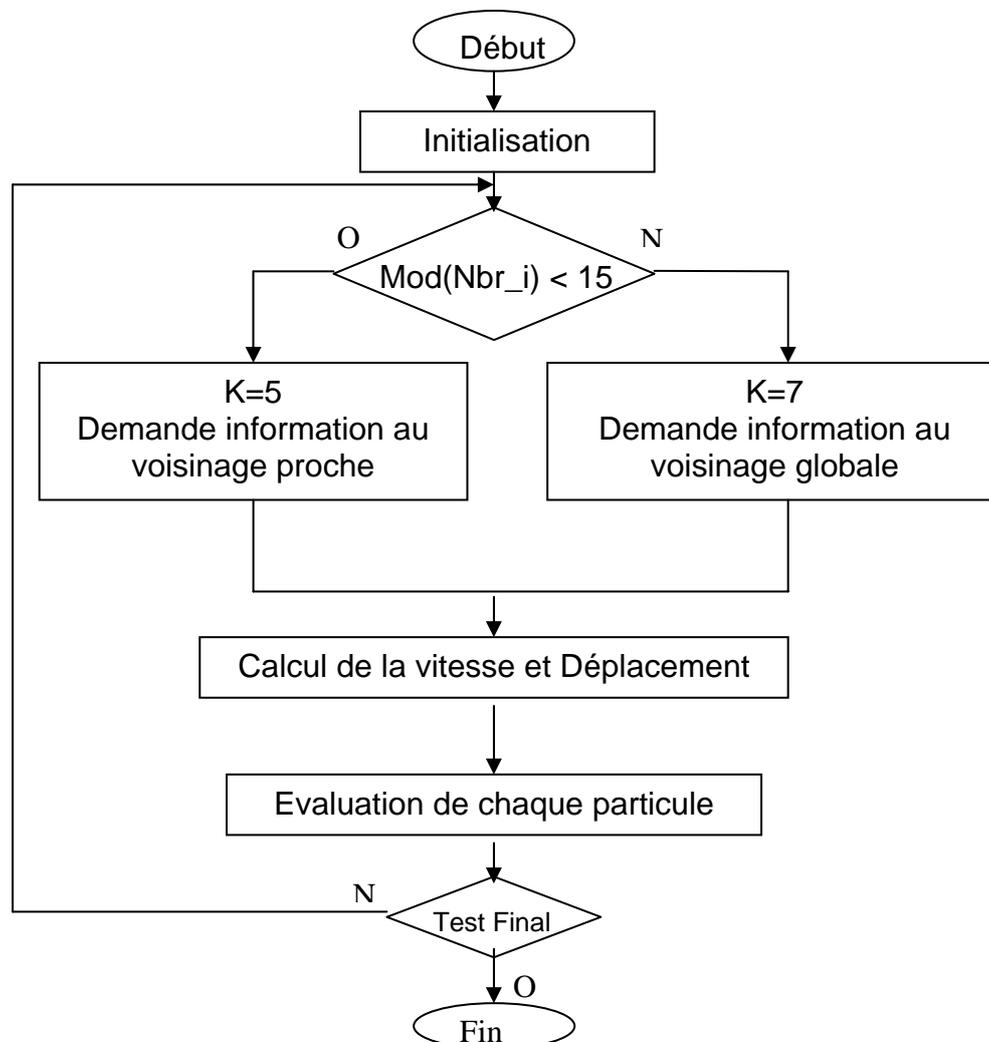


Figure 3.2 Algorithme PSO a deux structures de voisinage

- La recherche locale est illustrée par Figure 3.4 où chaque particule demande de l'information de deux particules géométriquement plus proches. La réduction de la vitesse est proportionnelle au rapprochement des particules. Par conséquent l'exploitation du site locale est mieux réalisée.

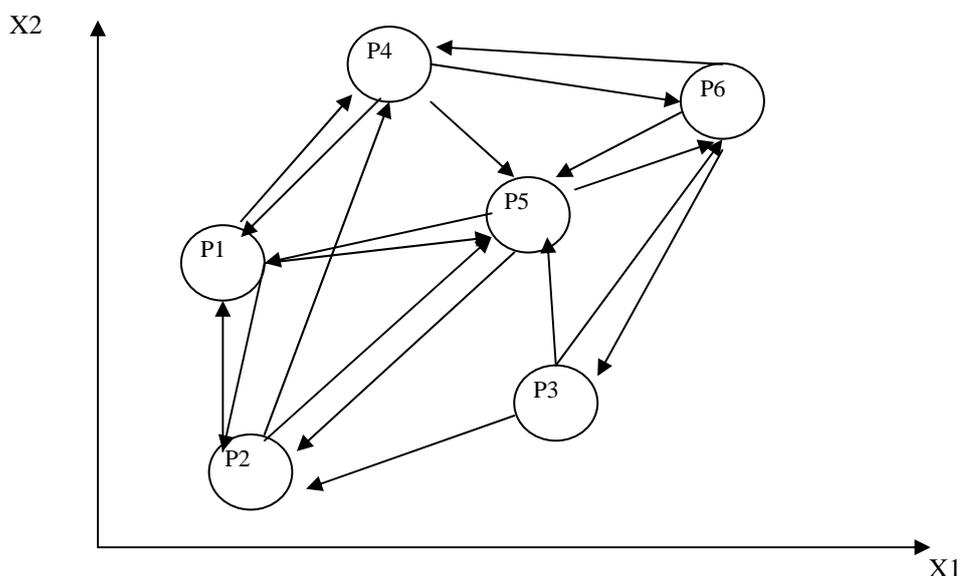


Figure 3.3: recherché avec une structure de voisinage globale
 —→ : Canal de communication avec la particule voisine

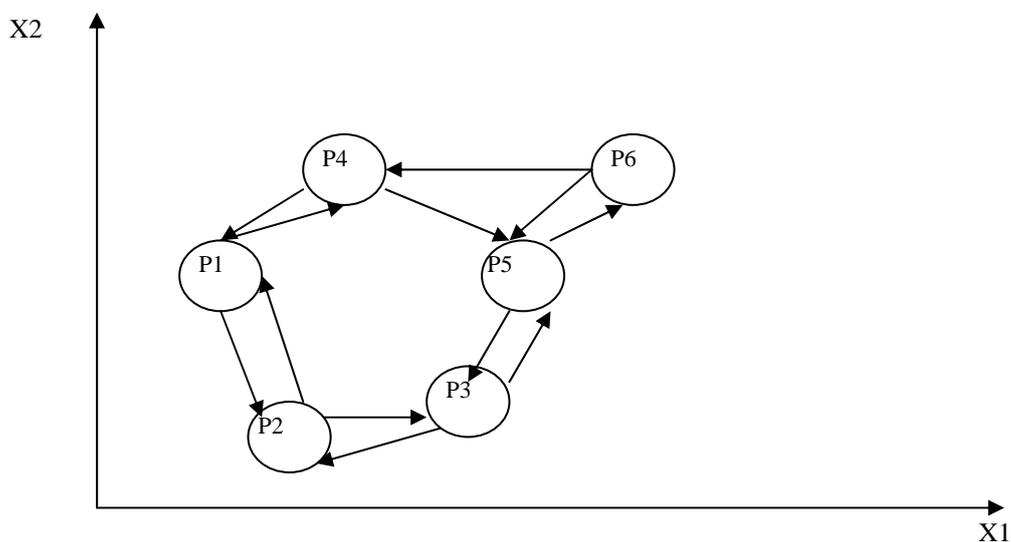


Figure 3.4: Recherche avec une structure de voisinage géométriquement proche.
 —→ : Canal de communication avec la particule voisine

Les paramètres de l'algorithme :

C1 = 0.689 : coefficient d'accélération.

C2 = 1.426 : coefficient de confiance dans son record.

C3 = 1.426 : coefficient de confiance dans le voisinage.

N = 40 particules : La taille de l'essaim

K = 7 pour le voisinage global et K = 5 pour le voisinage local.

3.2.3. Algorithme PSO avec coefficient de suivi

Notre approche vise à diminuer le temps de recherche par l'exploitation non seulement du meilleur record transmis par le voisinage, mais aussi la trajectoire empruntée. Ceci permet l'orientation de la zone probable du déplacement vers une région plus prometteuse. Cette technique permet une augmentation de la probabilité d'améliorer le record réalisé.

L'introduction de la trajectoire dans l'équation de mouvement est représentée par la vitesse (V_m) avec laquelle la particule a réalisé son meilleur record, et un coefficient de suivi nommé C4. La nouvelle formule de la vitesse devient :

$$V_i(t+1) = C1 * V_i(t) + C2 * rand * (X_{pbest} - X_i) + C3 * rand * (X_{vbest} + C_4 * V_m - X_i) \quad (3.3)$$

Cette technique est illustrée par la figure 3.5 pour un problème à deux variables et un essaim à trois particules :

- La particule P1 se trouve dans la position P11 et elle se déplace avec une vitesse V1 vers la position P12.
- La position P12 est la meilleure position transmise à la particule P2 qui se trouve dans la position P21 et elle se déplace avec une vitesse V2.
- à la prochaine itération, la deuxième particule va se déplacer en considérant sa vitesse, son meilleur record réalisé et la meilleure position transmise par le voisinage (P12) mais avancé suivant la vitesse de P1. La zone hachurée dans la figure 3.5 représente l'espace probable de déplacement suivant la probabilité et les coefficients de confiance.

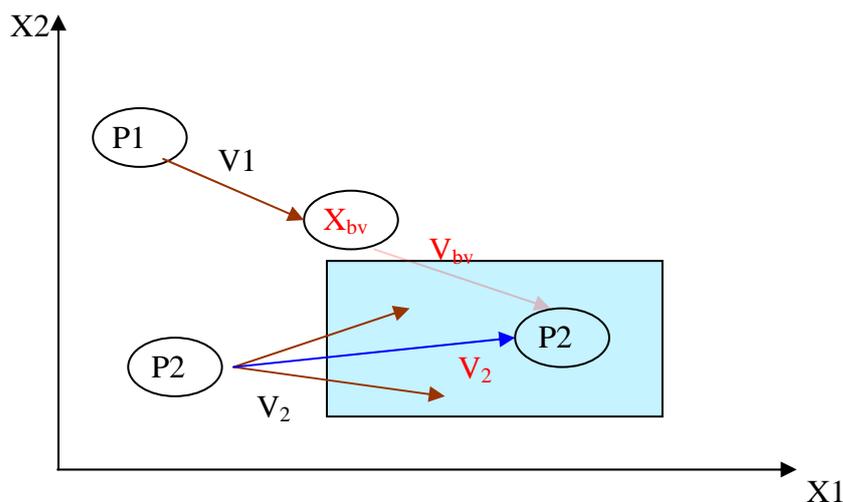


Figure 3.5. Décalage de la zone probable du déplacement .

3.3. Identification à l'aide des données simulées

A l'aide de données simulées nous nous proposons de:

- Valider les programmes développés.
- Montrer que la connaissance du courant et de la tension est suffisante pour déterminer simultanément les paramètres électriques et mécaniques caractérisant la machine.
- Montrer que la fonction représentant l'erreur quadratique entre les grandeurs mesurées et les grandeurs calculées présente un minimum global, en d'autres termes, que le vecteur paramètre représentant le minimum de cette fonction est unique.

Les données simulées sont obtenues par résolution numérique du système non linéaire (2.20) (décrivant le fonctionnement de la machine) par la méthode de Runge-Kutta en utilisant les paramètres d'un moteur asynchrone triphasé connu.

Les données sont obtenues en alimentant le moteur par une tension sinusoïdale (figure 3.6).

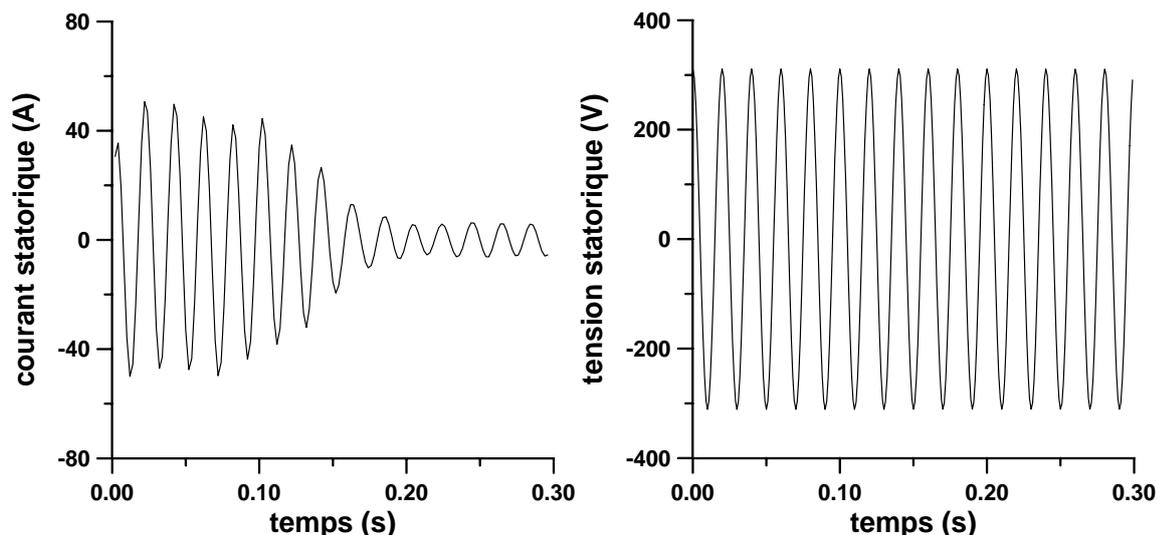


Figure 3.6 Courant de démarrage et tension simple correspondante.

3.3.1 Applications des différents algorithmes à l'identification

En utilisant les données simulées illustrées par les figures 3.7, nous estimons les paramètres du moteur à l'aide des trois algorithmes présentés précédemment. Pour chaque algorithme, nous avons effectué plusieurs tests sur un espace de recherche vaste. Nous présentons l'évolution des paramètres en fonction des itérations.

3.3.1.1 Algorithme PSO standard

Nous appliquons l'algorithme PSO standard pour identifier les paramètres du moteur en utilisant les données représentées par les figures 3.7.

L'intervalle de recherche :

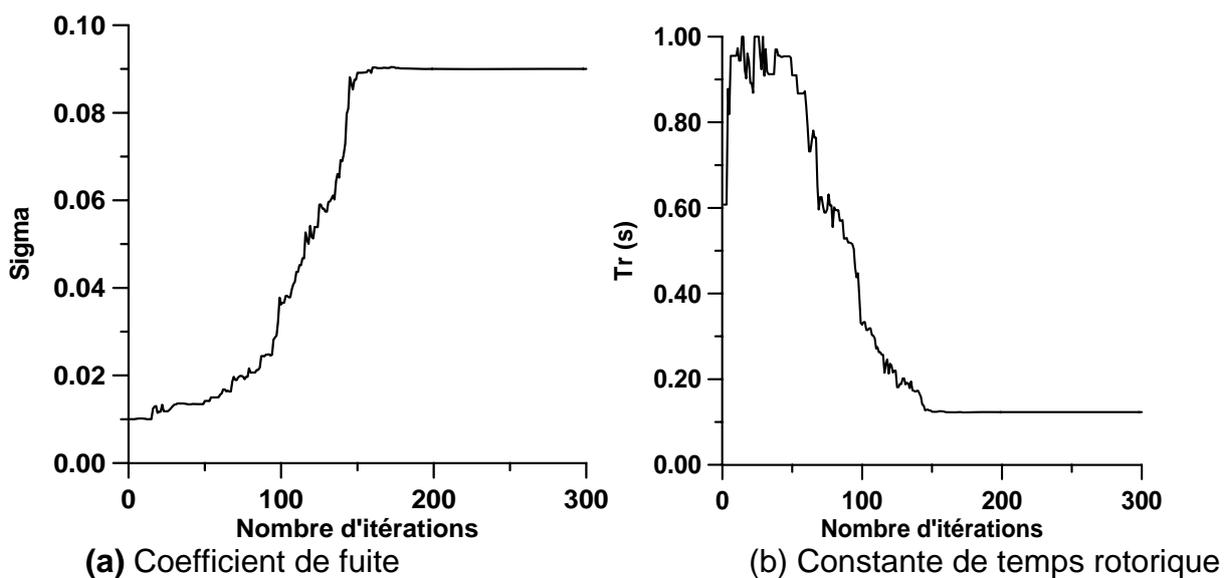
$$\begin{aligned} \text{Sigma} &= [0.000 - 1], \quad T_s = [0.0001 - 1], \quad L_s = [0.001 - 2], \quad T_r = [0.0001 - 1], \\ J &= [0.0001 - 0.1], \quad fr = [0.00001 - 0.1]. \end{aligned}$$

En respectant les conditions d'évaluations dans les méthodes Méta heuristiques, On exécute le programme 1000 fois. Vu que les paramètres sont connus exactement, l'erreurs quadratique doit atteindre la valeur zéro, (réellement inférieur à 10^{-7}). Le tableau 3.1 donne les paramètres estimés suivant l'évolution moyenne des itérations. En remarque qu'après 210 itérations tout les cas d'exécution convergent vers le minimum global. Sachons que à 150 itérations 75% des cas d'exécution convergent vers le minimum global.

Paramètres	Valeur connus	L'évolution des paramètres estimés		
		A 100 itérations	A 150 itérations	A 210 itérations
σ	0.09	0.036258	0.089134	0.089997
Ts(ms)	54	130.355	54.630	54.002
Ls(mH)	159	393.124	160.675	159.005
Tr(ms)	123	327.611	123.803	123.003
J(Kg.m ²)	0.038	0.038182	0.037992	0.038001
Fr(Nm.s/Rd).	0.001	0.000567	0.002671	0.001002

Tableau 3.1. Identification des paramètres par L'algorithme de PSO standard

Les figures de 3.7 (a, b, c, d, e, f) représentent l'évolution des paramètres en fonction des itérations, en remarque bien la convergence de vecteur paramètre ver la solution finale.



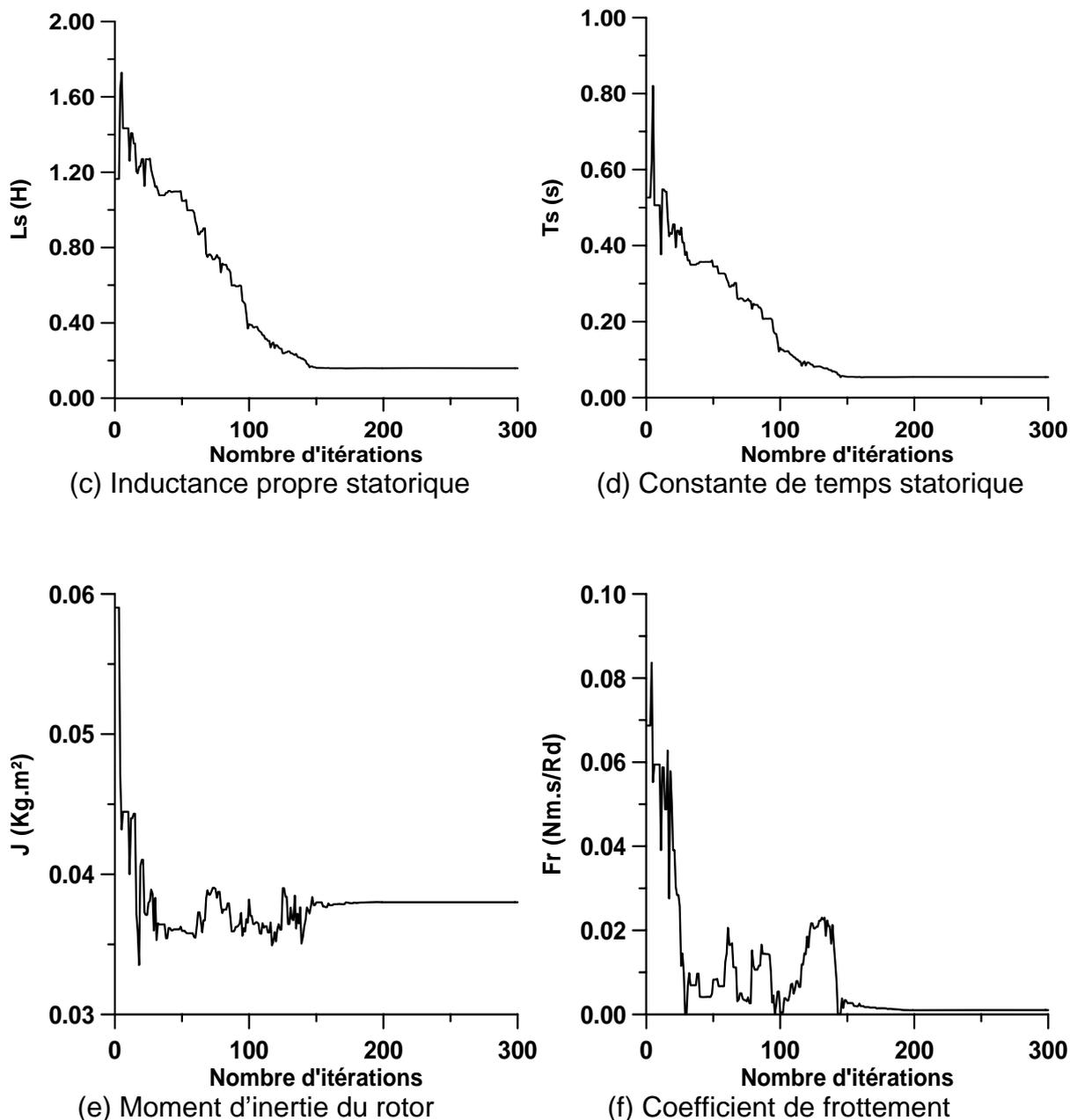


Figure 3.7 L'évolution des paramètres du moteur simulé

3.3.1.2 Algorithme PSO avec deux structures de voisinage

Nous appliquons l'algorithme PSO avec deux structures de voisinage pour identifier les paramètres du moteur en utilisant les données représentées par les figures 3.7, et en respectant les mêmes conditions d'évaluation.

L'intervalle de recherche des paramètres:

$$\text{Sigma} = [0.000, 1]; \text{Ts} = [0.0001, 1]; \text{Ls} = [0.001, 2]; \text{Tr} = [0.0001, 1],$$

$$\text{J} = [0.0001, 0.1]; \text{fr} = [0.00001, 0.1].$$

Le tableau 3.2 donne les paramètres estimés suivant l'évolution des itérations. On remarque qu'après 193 itérations tous les cas d'exécution convergent vers le minimum global. Sachant que à 150 itérations 85% des cas d'exécution convergent vers le minimum global.

Paramètres	Valeur connus	Paramètres estimés		
		A 100 itérations	A 150 itérations	A 193 itérations
σ	0.09	0.083675	0.089984	0.090000
Ts(ms)	54	56.355	54.004	54.001
Ls(mH)	159	172.590	159.038	159.001
Tr(ms)	123	135.290	123.034	123.000
J(Kg.m2)	0.038	0.036105	0.037994	0.038000
Fr(Nm.s/Rd).	0.001	0.006238	0.001024	0.001000

Tableau 3.2. Identification des paramètres par L'algorithme de PSO à Deux structures de voisinages

3.3.1.3 Algorithme PSO avec coefficient de suivie

Nous appliquons l'algorithme PSO avec modification de la vitesse pour identifier les paramètres du moteur en utilisant les données représentées par les figures 3.7, et les mêmes conditions d'évaluation.

L'intervalle de recherche des paramètres:

$$\text{Sigma} = [0.000 - 1], \text{ Ts} = [0.0001 - 1], \text{ Ls} = [0.001 - 2], \text{ Tr} = [0.0001 - 1], \\ \text{J} = [0.0001 - 0.1], \text{ fr} = [0.00001 - 0.1].$$

Le tableau 3.3 donne les paramètres estimés suivant l'évolution des itérations.

On remarque qu'après 194 itérations tout les cas d'exécution convergent vers le minimum global. Sachons que à 150 itérations 88% des cas d'exécution convergent vers le minimum global.

Paramètres	Valeur connus	Paramètres estimés		
		A 100 itérations	A 150 itérations	A 195 itérations
σ	0.09	0.090236	0.089996	0.089998
Ts(ms)	54	53.840	54.003	54.001
Ls(mH)	159	158.481	159.009	159.003
Tr(ms)	123	122.555	123.008	123.002
J(Kg.m ²)	0.038	0.038027	0.037999	0.038000
Fr(Nm.s/Rd).	0.001	0.006238	0.001024	0.001000

Tableau 3.3. Identification des paramètres par L'algorithme de PSO avec Coefficient de suivie.

3.3.2 Comparaison des résultats :

La figure 3.8 donne une comparaison de l'évolution de la fonction fitness en fonction des itérations entre les trois algorithmes. On peut remarquer les points suivants :

- Les trois algorithmes convergent vers le même vecteur paramètres pour tous les cas d'exécution, cela garanti que le minimum trouvé est toujours le minimum global.
- La vitesse de convergence du PSO standard est moins rapide vu que la méthode de communication entre les particules favorise la recherche local.
- Pour les deux autres algorithmes, on remarque deux comportements différents suivant que l'algorithme est loin ou près du solution global : L'algorithme avec coefficient de suivi est plus rapide dans la partie lointaine vu que l'algorithme prévoit la zone de déplacement en avant de la meilleur

position trouvé. Par contre l'algorithme avec deux structures de voisinage améliore sa vitesse de convergence en approchant du minimum global, grâce à sa structure de voisinage local.

En considérant la convergence globale, l'algorithme avec deux structures de voisinage converge plus rapidement que les autres algorithmes.

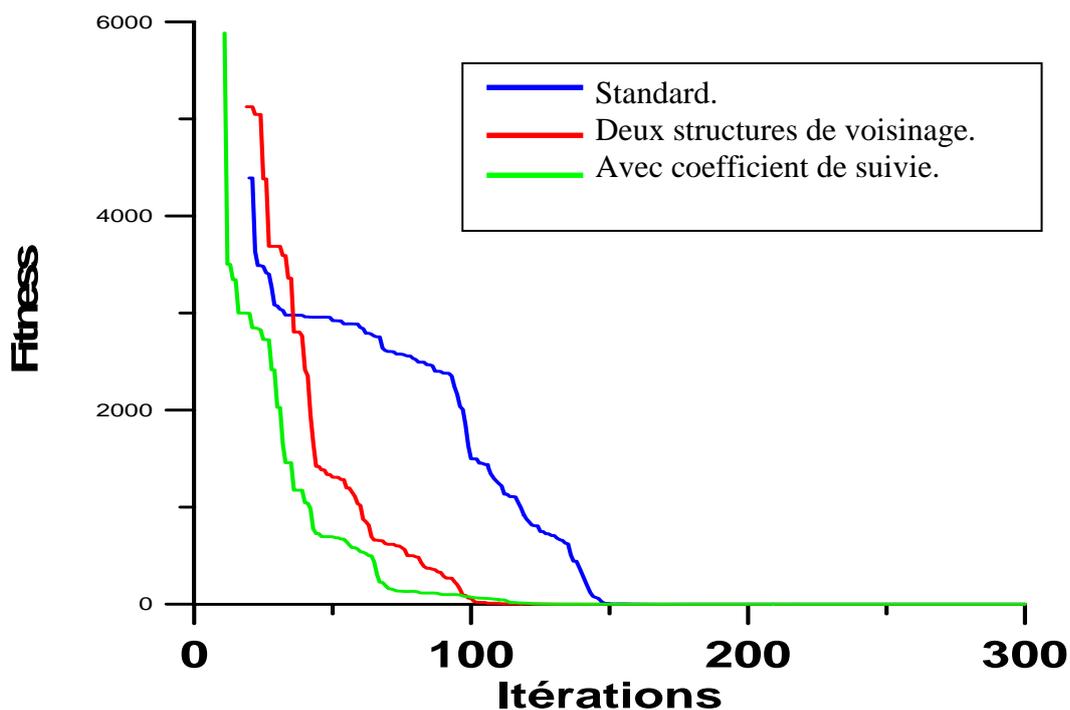


Figure 3.8 L'évolution de la fonction fitness par les trois algorithmes.

3.4. Identification à l'aide des données mesurées

Après les résultats de simulation qui concordent avec les données préétablies et valides les méthodes, nous avons procédé à des relevés expérimentaux sur deux moteurs asynchrones triphasés M1 et M2 et présentant respectivement les caractéristiques suivantes:

Moteur M1: 2 pôles, 220/380V, 3kW.

Moteur M2: 4 pôles, 220/380V, 1.5kW.

Les courbes dans la figure 3.9 et figure 3.10 montrent respectivement le courant de démarrage à vide et la tension simple correspondante mesurés sur une phase.

Ces essais sont utilisés pour estimer le vecteur paramètres de moteur à l'aide des trois algorithmes développés précédemment.

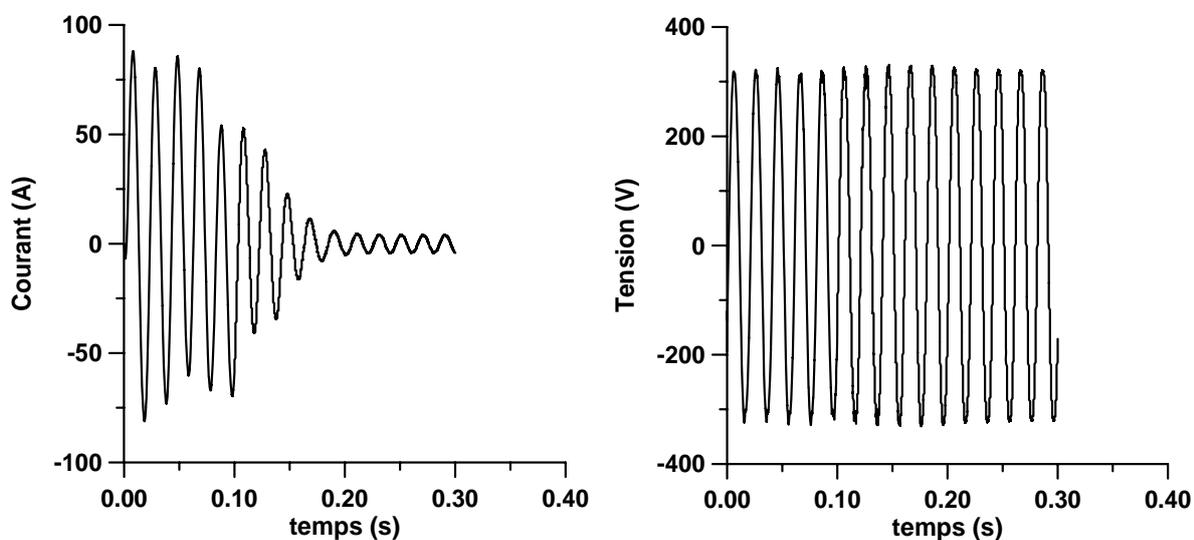


Figure 3.9. Courant de démarrage et tension simple correspondante au moteur M1

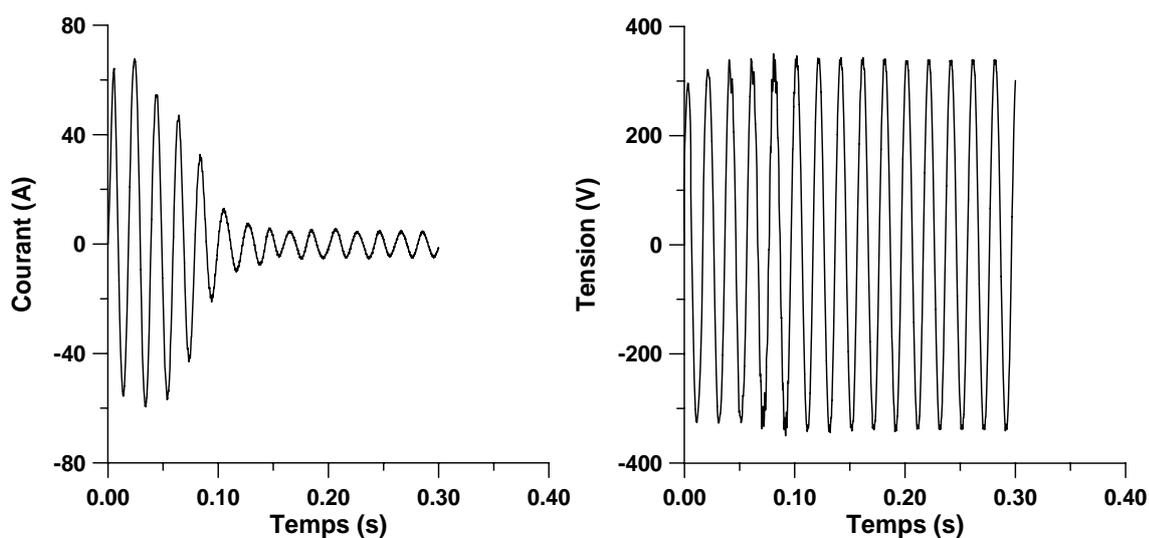


Figure 3.10 Courant de démarrage et tension simple correspondante au moteur M2

Le problème dans le cas du moteur réel est le test d'arrêt de l'algorithme, vu que les paramètres sont inconnus et la valeur minimale de l'erreur quadratique est inconnue aussi.

Vu que les points initiaux des particules sont aléatoires, il est possible que certains points favorisent une méthode par rapport à une autre. Donc le nombre des tests doit être élevé avec des points de démarrages différents.

Pour le nombre d'itérations exigé dans chaque test, il y a plusieurs démarches :

- Fixer le nombre d'itérations plus que le nombre maximum obtenu par simulation
- Arrêter le test si l'erreur quadratique ne diminue pas pendant un certain nombre d'itérations estimé par expériences.

Notre choix est porté sur la deuxième méthode vu que le système réel est différent du système simulé.

3.4.1 Identification du moteur M1

Le tableau 3.4 donne les paramètres estimés pour chaque algorithme et son examen montre que les trois algorithmes convergents vers le même vecteur paramètre.

Concernant la vitesse de convergence, les mêmes remarques obtenues dans la partie simulation sont valables pour les trois algorithmes.

L'erreur quadratique minimale obtenue est de 45712.26 après 200 itérations. Le nombre d'itérations totale de l'algorithme est égal à 400 itérations.

Notation :

PSO1 : algorithme standard

PSO2 : Deux structures de voisinage

PSO3 : Avec coefficient de suivi

Paramètres	Valeurs optimales par PSO1, PSO2, PSO3
σ	0.03379
Ts(ms)	146.7
Ls(mH)	251,661
Tr(ms)	199,641
J(Kg.m ²)	22,297 10 ⁻³
f _r (Nm.s/Rd)	0.002717

Tableau 3.4. Paramètres du moteur M1 estimés par les trois algorithmes

La figure 3.11 montre que le courant calculé avec le vecteur paramètre se superpose avec le courant mesuré aussi bien durant le régime transitoire que le régime permanent avec une erreur due aux problème des mesures et aussi au modèle choisi.

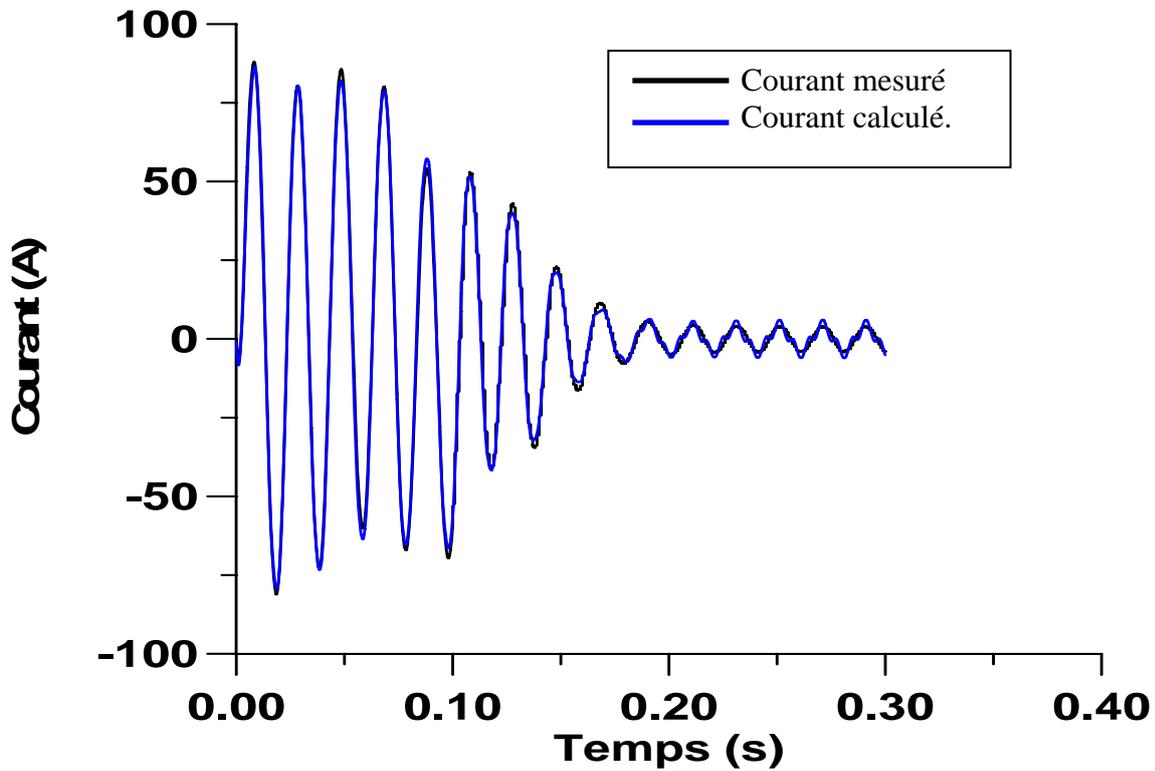
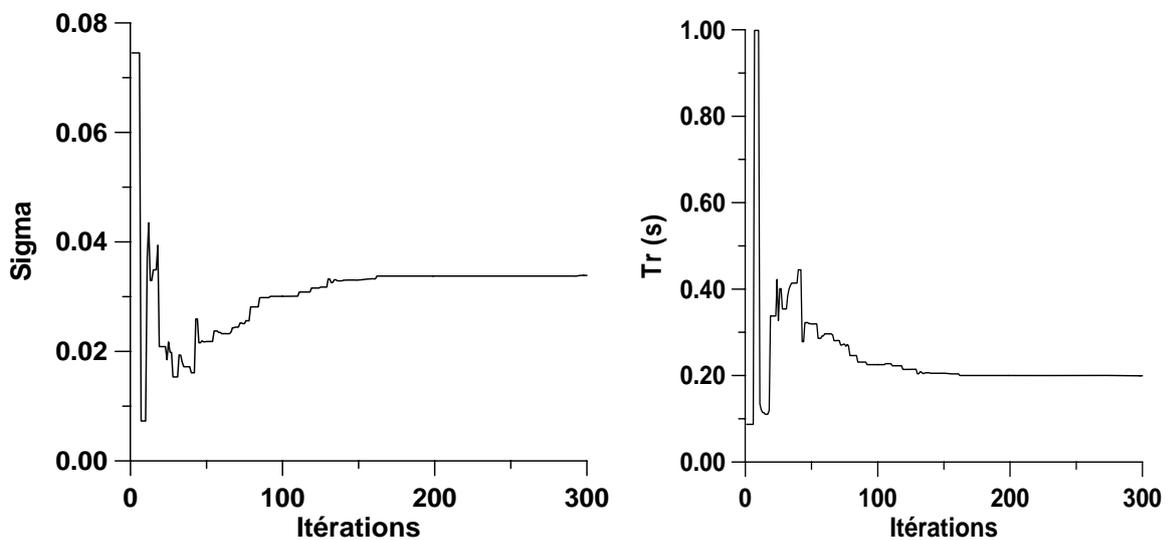


Figure 3.11 La superposition des deux courants : mesuré et calculé.

La figure 3.12 (a,b,c,d,e,f) donnent l'évolution des paramètres du moteur M1 en fonction des itérations.



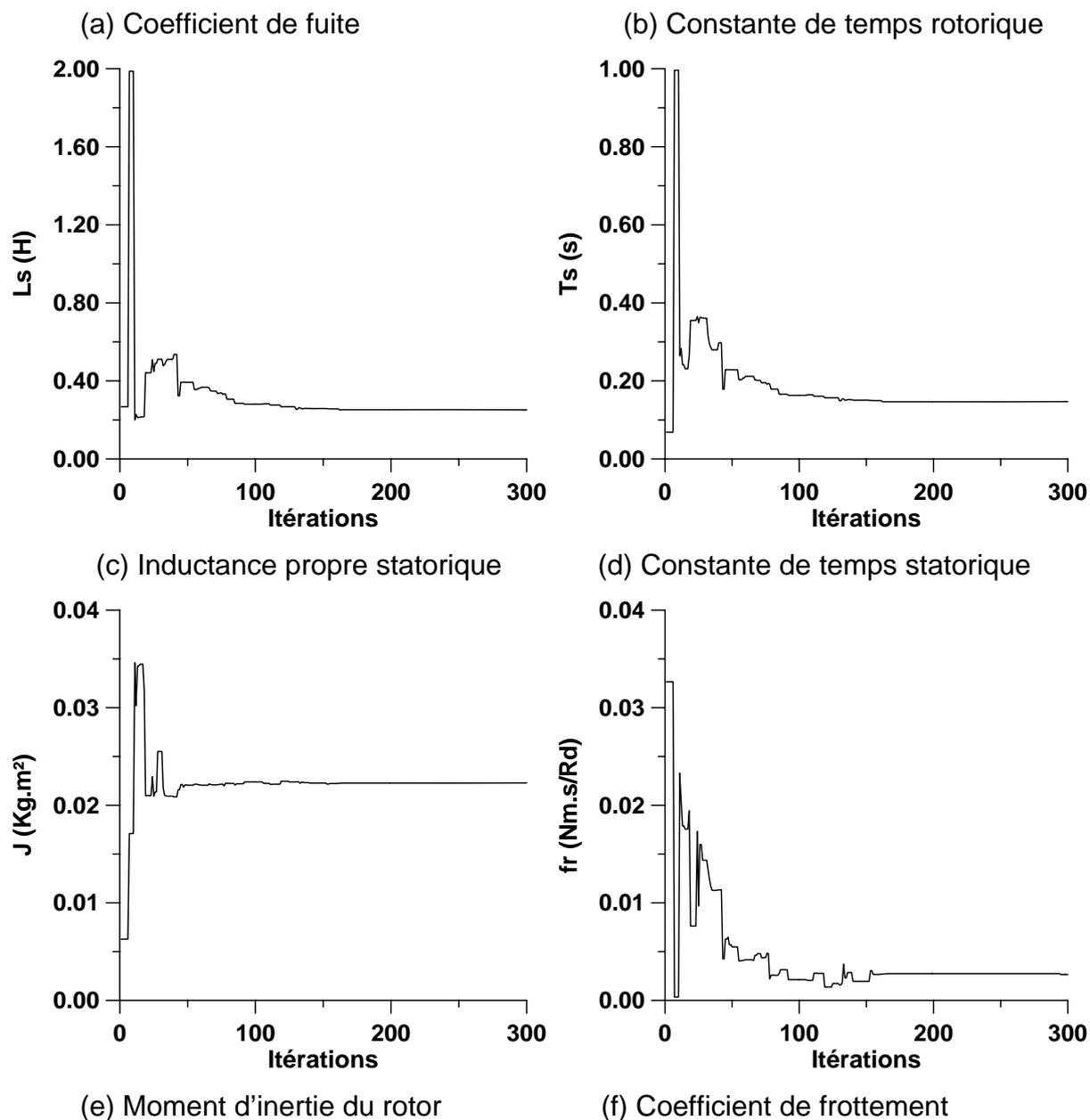


Figure 3.12 L'évolution des paramètres du moteur M1

3.4.2 Identification du moteur M2

Le tableau 3.5 donne les paramètres estimés pour chaque algorithme et son examen montre que les trois algorithmes convergent vers le même vecteur paramètre. Concernant la vitesse de convergence, les mêmes remarques obtenues dans la partie simulation sont valables pour les trois algorithmes. L'erreur quadratique minimale obtenue est de 62412.18 après 230 itérations. Le nombre d'itérations ajouté est égal à 200 itérations.

Paramètres	Valeurs optimales par PSO1, PSO2, PSO3
σ	0.063706
Ts(ms)	94.039
Ls(mH)	201,307
Tr(ms)	143,818
J(Kg.m2)	36,168 10^{-3}
f _r (Nm.s/Rd)	0.011667

Tableau 3.5. Paramètres du moteur M1 estimés par les trois algorithmes

Les figures 3.13 montre que le courant calculé avec le vecteur paramètre se superpose avec le courant mesuré aussi bien durant le régime transitoire que le régime permanent avec une erreur qui apparaît plus dans le régime permanent.

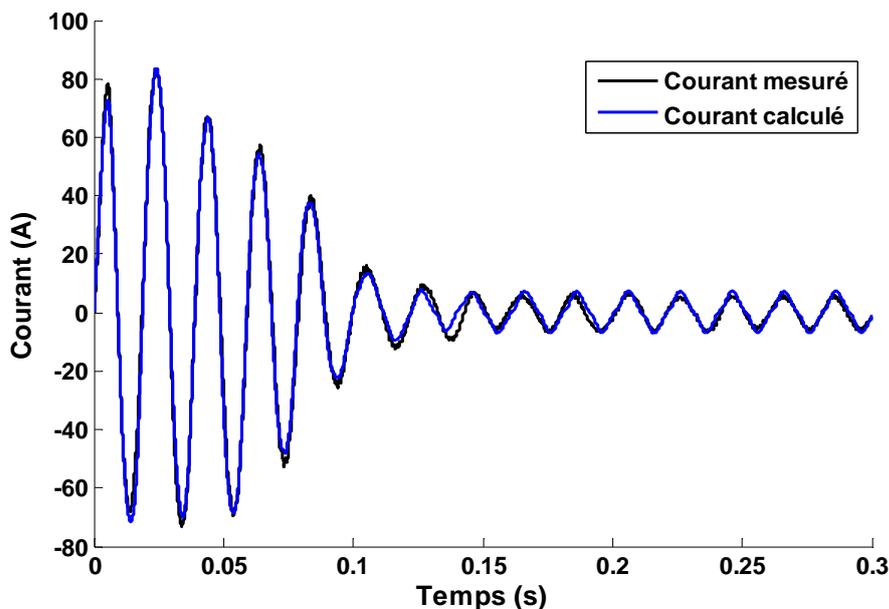


Figure 3.13 Superposition des deux courants : mesuré et calculé du moteur M2

L'évolution des paramètres est montrée en Appendice A1.

3.5. Comparaison des résultats

Les figures 3.14 et 3.15 montrent un agrandissement sur le courant statorique en régime permanent, on remarque bien que le signal est loin d'être uniforme. Le moteur M2 présente plus d'harmoniques que le moteur M1, cela justifie l'erreur quadratique minimale du moteur M2 est plus élevée que l'erreur quadratique minimale du moteur M1. Aussi ces perturbations rendent difficile la recherche du minimum

global, cela justifie le nombre d'itérations élevées pour la recherche du vecteur paramètres du moteur M2.

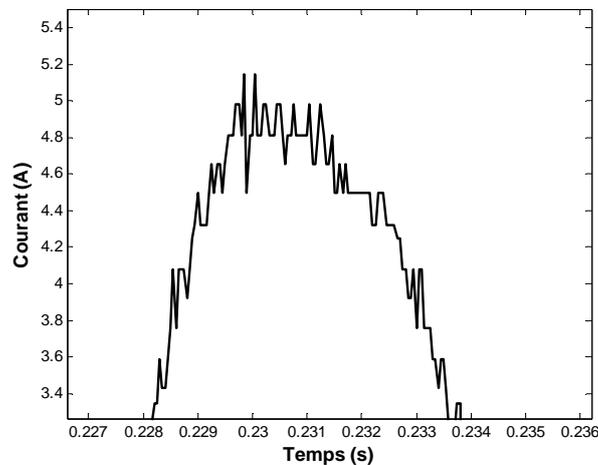


Figure 3.14 Un zoom sur le courant statorique mesuré du moteur M1

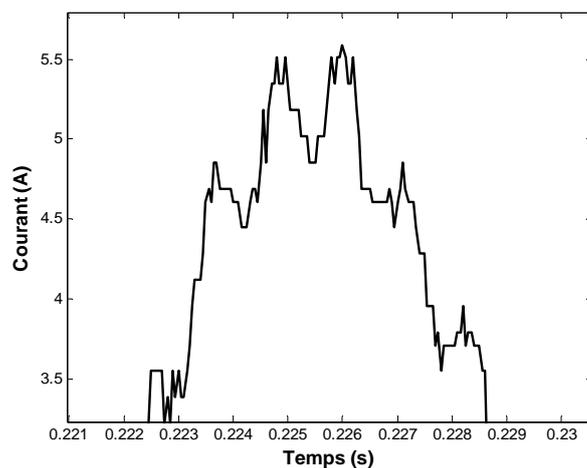


Figure 3.15 Un zoom sur le courant statorique mesuré du moteur M2

3.6. Conclusion :

Dans ce chapitre nous avons présenté trois variantes d'algorithme de PSO permettant la détermination simultanée des paramètres électriques et mécaniques de la machine et cela uniquement à partir de la mesure du courant de démarrage et de la tension simple correspondante. Nous avons validé les algorithmes par des données simulés et un nombre des tests élevé. Le modèle mathématique adopté ainsi que les trois algorithmes sont validés sur des relevés expérimentaux effectués sur deux moteurs différents. Les résultats obtenus sont très satisfaisants, mais une amélioration du modèle mathématique est nécessaire afin de minimiser encore l'erreur quadratique.

CHAPITRE 4

IDENTIFICATION DE LA M.A.S. AVEC SATURATION

4.1. Introduction

Dans ce chapitre, nous commençons par présenter le phénomène de la saturation ainsi que les harmoniques et l'effet croisé qui lui sont associés. Après un rappel sur les différents flux et inductances de fuites, nous donnons les équations régissant le fonctionnement de la machine en régime saturé. Nous validons ensuite notre technique d'identification, en estimant les paramètres électriques, mécaniques ainsi que la courbe de magnétisation de la machine.

4.2. Présentation de la saturation

La saturation des matériaux ferromagnétiques est un phénomène physique complexe et difficile à modéliser. Dans les machines électriques, la saturation apparaît d'abord dans les zones ferromagnétiques les plus rétrécies, habituellement, dans les dents statoriques et rotoriques et également dans les cornes polaires pour les machines à pôles saillants. Il s'agit donc d'un phénomène local. La prise en compte de la saturation dans la modélisation des machines peut se faire de différentes façons. La méthode consiste à définir la géométrie exacte de la machine étudiée. On résout ensuite les équations de Maxwell associées aux lois de comportement des matériaux employés en utilisant des méthodes numériques comme les éléments finis [69].

Cette approche est intéressante en conception de machine (optimisation de la structure). Elle est par contre difficilement applicable pour la simulation des ensembles convertisseurs-machines-commande, car elle nécessite un temps de calcul très important.

La démarche la plus utilisée pour simuler l'ensemble convertisseur-machine consiste alors à modifier le modèle de Park. On considère la saturation du point de

vue global en regardant ses effets sur les grandeurs accessibles à la mesure tels que les courants ou les flux magnétiques. On conserve alors la notion d'inductances en modulant leurs valeurs en fonction des courants circulants dans les enroulements de la machine.

Nous allons rappeler que la saturation magnétique engendre des harmoniques supplémentaires ainsi qu'un couplage magnétique entre bobines en quadrature d'espace (appelé saturation croisée).

4.2.1 Harmoniques de saturation

La saturation crée des harmoniques qui diffèrent des harmoniques temporelles créées par l'alimentation et des harmoniques spatiales dues aux distributions non sinusoïdales des enroulements et des encoche [66]. Pour illustrer ce phénomène, on considère une excitation magnétique $H(\theta_e)$ étant à répartition sinusoïdale. Si l'on choisit une caractéristique magnétique $B(H)$ de la tôle approximée par:

$$H = H_0 \cdot \sin(\theta_e) \quad (4.1)$$

$$B(H) = a_1 H + a_3 H^3 \quad (4.2)$$

Le champ $B(\theta_e)$ dans l'entrefer s'écrit :

$$B(\theta_e) = B_1 \sin(\theta_e) + B_3 \sin(3\theta_e) \quad (4.3)$$

Nous remarquons que la non linéarité de la caractéristique de la tôle magnétique donnée par l'équation (4.2) induit dans B des harmoniques comme le montre l'équation (4.3) Le champ $B(\theta_e)$ dans l'entrefer est représenté sur la figure 4.1

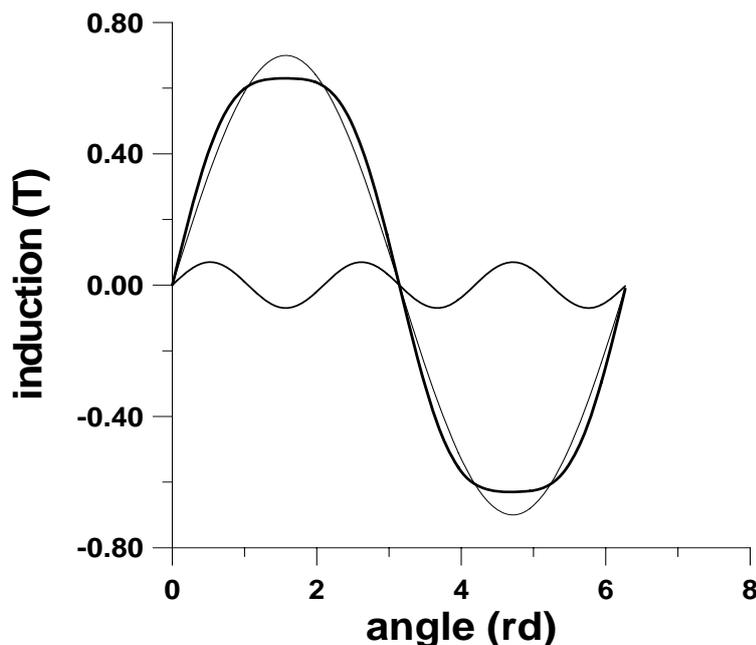


Figure 4.1. Allure du champ dans l'entrefer en régime saturé ;
Le fondamental et l'harmonique 3 sont représentés.

4.2.2 Le phénomène de saturation croisée

La saturation est à l'origine du phénomène de la saturation croisée ou effet croisé (Cross Magnetisation ou Cross Saturation). Il correspond à un couplage d'enroulements stationnaires dans un référentiel d'axes perpendiculaires.

Ce phénomène a été mis en évidence expérimentalement [67]. Les essais ont été réalisés sur une machine asynchrone de 1 kW à rotor bobiné. Les enroulements statoriques et rotoriques sont en quadrature d'espace. Le schéma du montage est donné par la figure 4.2. La machine est à l'arrêt. L'enroulement statorique d'axe d est alimenté par une source de tension continue. L'enroulement statorique d'axe q est également alimenté par une source de tension continue en série avec le secondaire d'un transformateur (tension sinusoïdale de faible amplitude). Les enroulements rotoriques non représentés sur ce schéma sont ouverts. On augmente progressivement les courants continus d'axes d et q. Quand ces courants sont suffisants pour saturer la machine, il apparaît aux bornes du bobinage d'axe d une force électromotrice induite. Cette expérience montre l'existence du couplage magnétique entre deux bobinages d'une machine lisse, placés en quadrature d'espace; ce qui met en évidence le phénomène de la saturation croisée

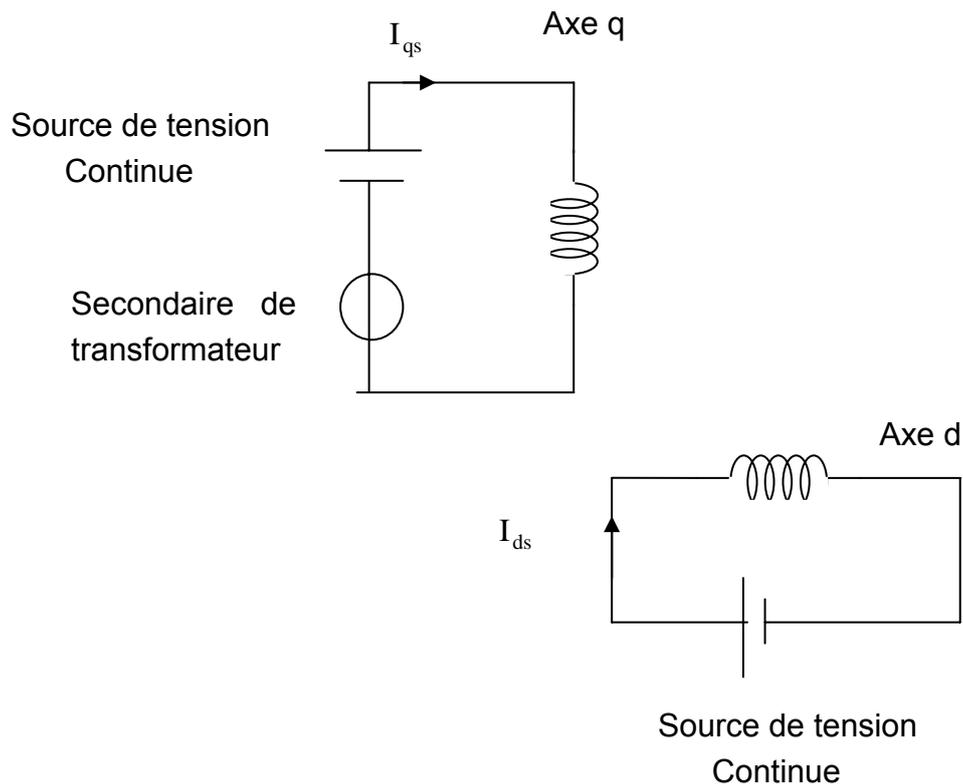


Figure 4.2. Schéma de mise en évidence de la saturation croisée

2.3 Modèle mathématique de la courbe de magnétisation

La courbe de magnétisation de la machine asynchrone issue de tests expérimentaux ou à l'aide de méthodes numériques de calculs de champs comme la méthode des éléments finis est représentée par un ensemble discret de points reflétant la relation entre le flux et le courant magnétisant.

Afin d'éviter les erreurs dues à la différentiation numérique lors du calcul de l'inductance dynamique, il est judicieux d'approcher la courbe de magnétisation par un modèle mathématique [68] [69]. Plusieurs modèles faisant appels à des fonctions polynomiales, des fonctions arc-tangente, des fonctions tangente-hyperboliques et exponentielles sont proposés dans la littérature. Dans la suite de notre travail, nous représentons la courbe de magnétisation par une fonction arc-tangente plus une série de cosinus.

$$\varphi_m = C_a * a \tan(C_b * I_m) + \sum_{i=1}^n C_i * \cos(0.2 * i * \pi * I_m + C_{n+i}) \quad (4.4)$$

4.3 Modélisation de la machine asynchrone

Des modèles saturés au sens du premier harmonique d'espace ont été proposés dans la littérature [70] [71]. Ils séparent le flux utile des fuites magnétiques qui sont distribuées de chaque côté de l'entrefer. Ces modèles font apparaître un flux commun au stator et au rotor qui dépend de l'état magnétique de la machine.

4.3.1 Flux commun et inductances de fuite

La saturation est prise en compte de la façon suivante : On suppose tout d'abord que les inductances de fuite sont constantes (elles correspondent à des lignes de champ magnétique qui traversent une partie importante dans l'air) et on considère ensuite que le flux commun φ_m au stator et au rotor est fonction du courant magnétisant.

On écrit :

$$\varphi_m = L_m(I_m)I_m \quad (4.5)$$

L_m représente l'inductance magnétisante qui est fonction du courant. Les inductances statoriques et rotoriques s'écrivent alors :

$$\begin{cases} L_s = l_{fs} + L_m \\ L_r = l_{fr} + L_m \end{cases} \quad (4.6)$$

Où l_{fs} et l_{fr} désignent respectivement les inductances de fuite statorique et rotorique.

En ne considérant que le fondamental du champ, les expressions des flux totaux statorique et rotorique (l.14) s'écrivent alors.

$$\begin{cases} \varphi_{ds} = l_{fs}I_{ds} + L_m(I_{ds} + I_{dr}) \\ \varphi_{qs} = l_{fs}I_{qs} + L_m(I_{qs} + I_{qr}) \end{cases} \quad (4.7)$$

$$\begin{cases} \varphi_{dr} = l_{fr}I_{dr} + L_m(I_{dr} + I_{ds}) \\ \varphi_{qr} = l_{fr}I_{qr} + L_m(I_{qr} + I_{qs}) \end{cases} \quad (4.8)$$

On voit apparaître dans ces expressions les flux communs aux enroulements statorique et rotorique. Ces flux sont appelés flux magnétisants d'axe d et d'axe q.

$$\varphi_{dm} = L_m I_{dm} \quad \text{Avec} \quad I_{dm} = I_{ds} + I_{dr} \quad (4.9)$$

$$\varphi_{qm} = L_m I_{qm} \quad \text{Avec} \quad I_{qm} = I_{qs} + I_{qr} \quad (4.10)$$

On a :

$$\varphi_m = \sqrt{\varphi_{dm}^2 + \varphi_{qm}^2} \quad \text{Et} \quad I_m = \sqrt{I_{dm}^2 + I_{qm}^2} \quad (4.11)$$

4.3.2 Equations électriques de la machine

Pour la mise en équation, nous partons de l'équation (4.6) où nous faisons apparaître les inductances de fuite l_{fs} et l_{fr} ainsi que les deux composantes du flux magnétisant φ_{dm} et φ_{qm} données par les équations (4.7 et 4.8). Les équations des tensions sont alors données par le système suivant :

$$\begin{cases} V_{ds} = R_s I_{ds} + l_{fs} \frac{dI_{ds}}{dt} + \frac{d\varphi_{dm}}{dt} \\ V_{qs} = R_s I_{qs} + l_{fs} \frac{dI_{qs}}{dt} + \frac{d\varphi_{qm}}{dt} \\ V_{dr} = R_r I_{dr} + l_{fr} \frac{dI_{dr}}{dt} + \frac{d\varphi_{dm}}{dt} + P\Omega(l_{fr} I_{qr} + \varphi_{qm}) \\ V_{qr} = R_r I_{qr} + l_{fr} \frac{dI_{qr}}{dt} + \frac{d\varphi_{qm}}{dt} - P\Omega(L_{fr} I_{dr} + \varphi_{dm}) \end{cases} \quad (4.12)$$

Le calcul de $\frac{d\varphi_{dm}}{dt}$ et $\frac{d\varphi_{qm}}{dt}$ (donné en appendices C et D) conduit aux expressions suivantes :

$$\begin{cases} \frac{d\varphi_{dm}}{dt} = (L_o + L_{2c}) \frac{dI_{dm}}{dt} + L_{2s} \frac{dI_{qm}}{dt} \\ \frac{d\varphi_{qm}}{dt} = L_{2s} \frac{dI_{dm}}{dt} + (L_o - L_{2c}) \frac{dI_{qm}}{dt} \end{cases} \quad (4.13)$$

Avec

$$\begin{cases} L_{2c} = L_2 \cos 2\mu \\ L_{2s} = L_2 \sin 2\mu \end{cases} \quad (4.14)$$

$$\begin{cases} L_2 = \frac{L - L_m}{2} \\ L_0 = \frac{L + L_m}{2} \end{cases} \quad (4.15)$$

L et L_m désignent respectivement l'inductance dynamique et l'inductance magnétisante (inductance statique)

$$\begin{cases} L = \frac{d\varphi_m}{dt} \\ L_m = \frac{\varphi_m}{I_m} \end{cases} \quad (4.16)$$

μ représente l'angle entre le courant magnétisant I_m et l'axe d lié au stator (fig. 4.3).

$$\begin{cases} \cos \mu = \frac{I_{md}}{I_m} \\ \sin \mu = \frac{I_{mq}}{I_m} \end{cases} \quad (4.17)$$

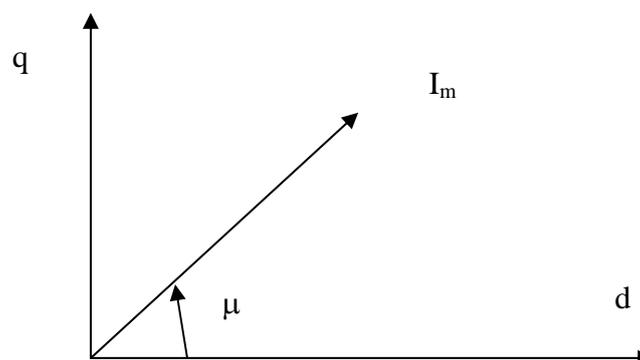


Figure 4.3. Représentation du courant magnétisant dans le repère dq

En considérant le vecteur d'état $[I_{ds} \ I_{qs} \ I_{dr} \ I_{qr} \ \Omega]$ et sachant que :

$$I_{dm} = I_{ds} + I_{dr}$$

$$I_{qm} = I_{qs} + I_{qr}$$

Le système 4.12 s'écrit:

$$\begin{cases} V_{ds} = R_s I_{ds} + (L_{fs} + L_o + L_{2c}) \frac{dI_{ds}}{dt} + L_{2s} \frac{dI_{qs}}{dt} + (L_o + L_{2c}) \frac{dI_{dr}}{dt} + L_{2s} \frac{dI_{qr}}{dt} \\ V_{qs} = R_s I_{qs} + (L_{fr} + L_o + L_{2c}) \frac{dI_{ds}}{dt} + (L_o - L_{2c}) \frac{dI_{qs}}{dt} + L_{2s} \frac{dI_{dr}}{dt} + (L_o - L_{2c}) \frac{dI_{qr}}{dt} \\ V_{dr} = R_r I_{dr} + (L_{fr} + L_o + L_{2c}) \frac{dI_{ds}}{dt} + L_{2s} \frac{dI_{qs}}{dt} + (L_o + L_{2c}) \frac{dI_{dr}}{dt} + L_{2s} \frac{dI_{qr}}{dt} + P\Omega (L_{fr} I_{qr} + L_m I_{qr} + L_m I_{qs}) \\ V_{qr} = R_r I_{qr} + L_{2s} \frac{dI_{ds}}{dt} + (L_o - L_{2c}) \frac{dI_{qs}}{dt} + L_{2s} \frac{dI_{dr}}{dt} + (L_{fr} + L_o - L_{2c}) \frac{dI_{qr}}{dt} - P\Omega (L_{fr} I_{dr} + L_m I_{dr} + L_m I_{ds}) \end{cases} \quad (4.18)$$

Ce système peut se mettre sous la forme

$$U = RX + L \frac{dX}{dt} \quad (4.19)$$

Où R et L désignent respectivement les matrices « résistances » et inductances :

$$R = \begin{bmatrix} R_s & 0 & 0 & 0 \\ 0 & R_s & 0 & 0 \\ 0 & P\Omega L_m & R_r & P\Omega (l_{fr} + L_m) \\ -P\Omega L_m & 0 & -P\Omega (l_{fr} + L_m) & R_r \end{bmatrix} \quad (4.20)$$

$$L = \begin{bmatrix} l_{fs} + L_o + L_{2c} & L_{2s} & L_o + L_{2c} & L_{2s} \\ L_{2s} & l_{fs} + L_o - L_{2c} & L_{2s} & L_o - L_{2c} \\ L_o + L_{2c} & L_{2s} & l_{fr} + L_o + L_{2c} & L_{2s} \\ L_{2s} & L_o - L_{2c} & L_{2s} & l_{fr} + L_o - L_{2c} \end{bmatrix} \quad (4.21)$$

Dans les matrices (4.20) et (4.21) apparaissent les grandeurs L_m , L_o , L_{2c} et L_{2s} qui peuvent être déduites de la courbe de magnétisation donnée par l'équation 4.4. Ainsi, la machine est entièrement caractérisée par le vecteur des paramètres P suivant :

$$P = [L_{fs} \ L_{fr} \ R_s \ R_r \ J \ f_r \ C_a \dots C_n] \quad (4.22)$$

4.4 L'identification

Comme pour le chapitre précédent, nous cherchons à déterminer le vecteur P à partir de la mesure du courant du démarrage et de la tension correspondante. Nous présentons dans ce travail les résultats obtenus par des données simulées et par des relevés expérimentaux.

4.4.1 Le choix de l'algorithme PSO

Dans le chapitre précédent, on a montré que les trois algorithmes convergent, mais avec une supériorité de l'algorithme à deux structures de voisinage vu sa vitesse de convergence.

Dans ce chapitre on va appliquer seulement l'algorithme à deux structures de voisinage.

Les paramètres de l'algorithme :

$$C1 = 0.6$$

$$C2 = C3 = 1.4$$

N = 60 (La taille de l'essaim augmente vu l'augmentation de la taille du vecteur paramètres à optimiser).

K = [5 , 7] (voisinage proche , voisinage global).

4.4.2 Identification à l'aide des données simulées

A fin de valider le modèle mathématique de la courbe de magnétisation on procède par la méthode suivante :

- Fixer les paramètres électriques et mécaniques de la machine asynchrone.
- Considérer le modèle de la courbe de magnétisation avec le terme de l'arctangente seulement.

$$\varphi_m = C_a * a \tan(C_b * I_m)$$

- Chercher le vecteur paramètres en considérant le modèle mathématique de la courbe de magnétisation présenté par l'équation 4.4.

Le moteur utilisé pour la simulation a les caractéristiques suivantes :

- nombres des pôles = 4.
- Période d'échantillonnage dt = 0.0001s. sur un temps de : 0.4 s

Le Tableau 4.1 présente les résultats obtenus par l'algorithme PSO2.

Paramètres	Valeurs connus	Valeurs simulées	Paramètre	Valeurs simulées
L_{fr}	0.005	0.004977	C_1	0.001787
L_{fs}	0.005	0.004977	C_2	0.112165
R_s	2.85	2.821838	C_3	-0.003744
R_r	1.08	1.085840	C_4	2.644821
J	0.031	0.031907	C_5	-0.005226
f_r	0.001	0.001069	C_6	2.219305
C_a	1.0	1.00082	C_7	-0.014390
C_b	0.7	0.70010	C_8	1.0553142

Tableau 4.1 Les paramètres obtenu par des données simulées.

La figure 4.4 donne le tracé des deux courbes de magnétisation. On remarque que le modèle simulé superpose sur le modèle connu avec une légère déformation et une faible erreur sur les paramètres du moteur.

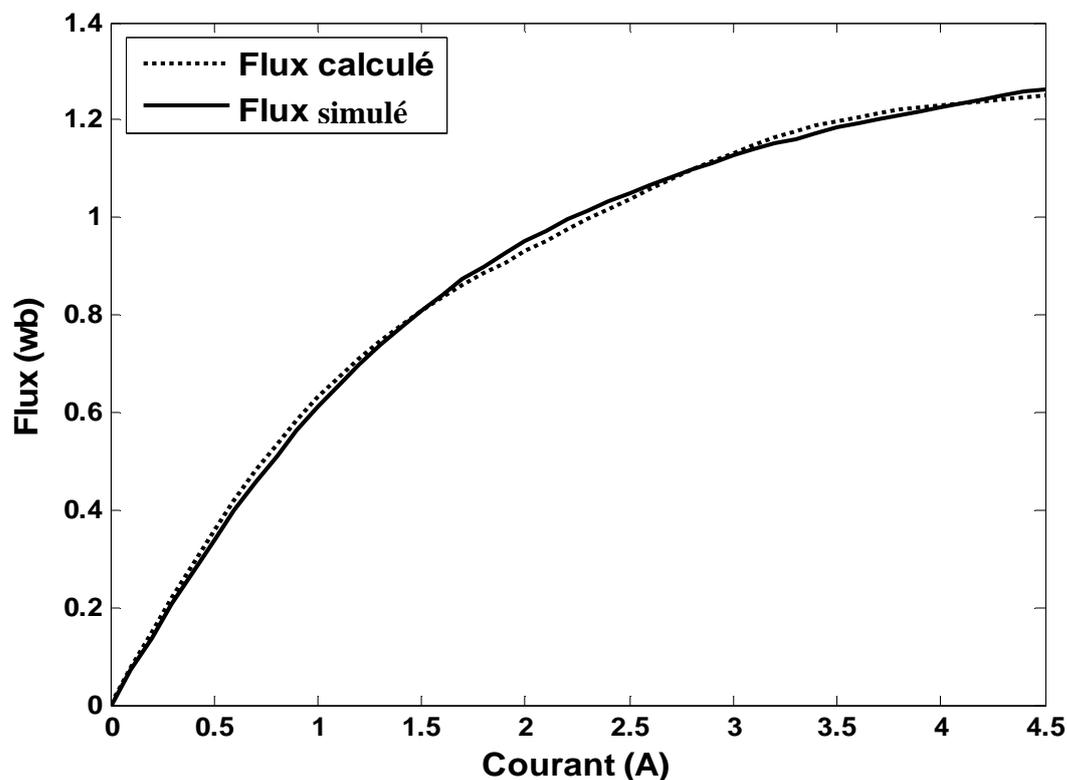
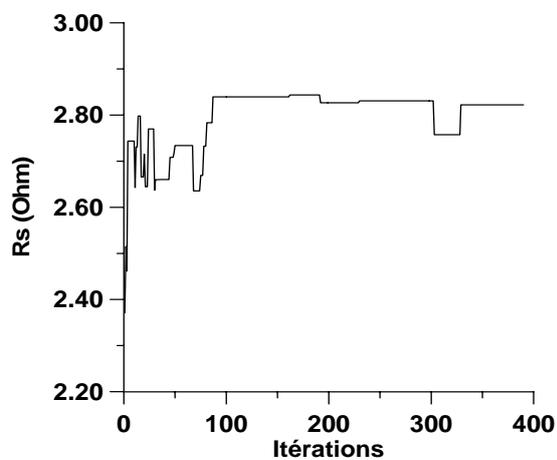
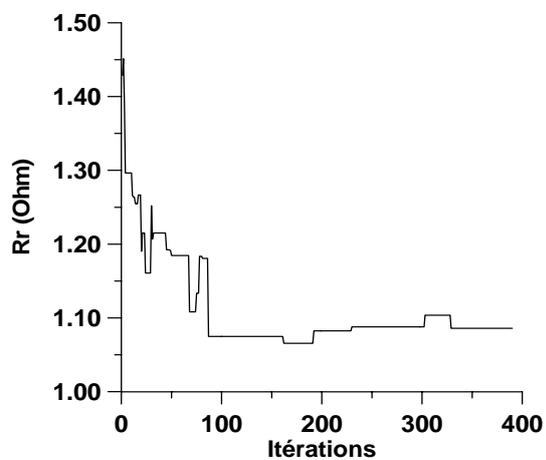


Figure 4.4. Les courbes de magnétisation : Simulé et calculé.

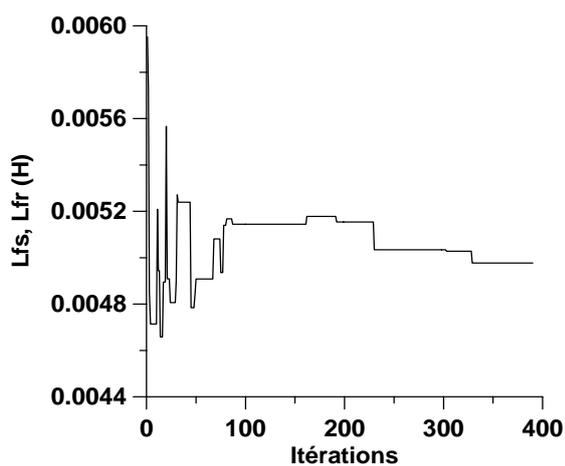
La figure 4.5 (a,b,c,d,e) représentent l'évolutions des paramètres suivant les itérations.



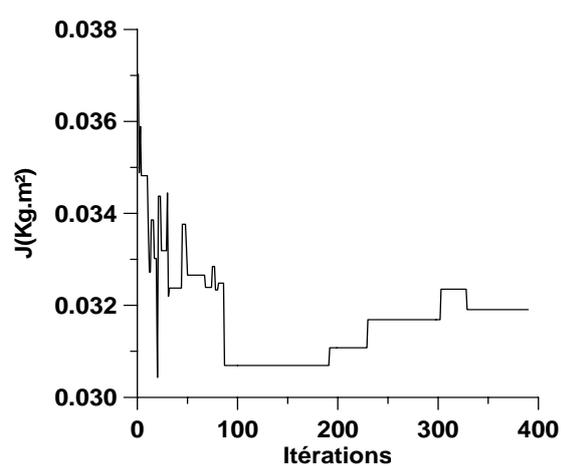
(a) L'évolution de la résistance statorique.



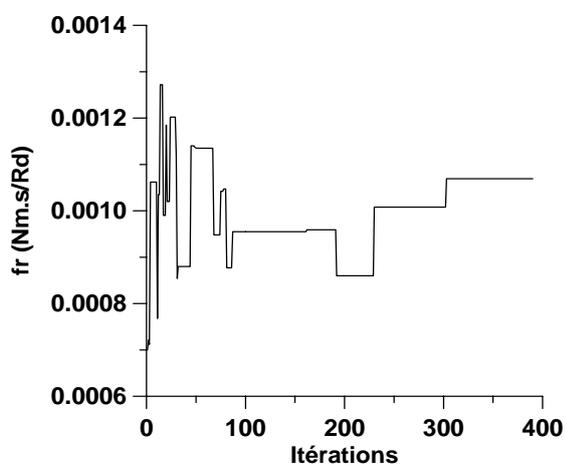
(b) L'évolution de la résistance rotorique.



(c) L'évolution de l'inductance de fuite.



(d) L'évolution du couple magnétique.



(e) L'évolution du coefficient de frottement.

Figure 4.5 L'évolution des paramètres du Moteur simulé

4.4.3 Relevés expérimentaux

Les moteurs M1 et M2 sont ceux utilisé dans le chapitre précédent.

- Moteur M1: 2 pôles, 220/380V, 3kW;
- Moteur M2: 4 pôles, 220/380V, 1.5kW;

Les courbes de 4.6 et 4.71 montrent respectivement le courant de démarrage à vide et la tension simple correspondante mesurés sur chaque moteur.

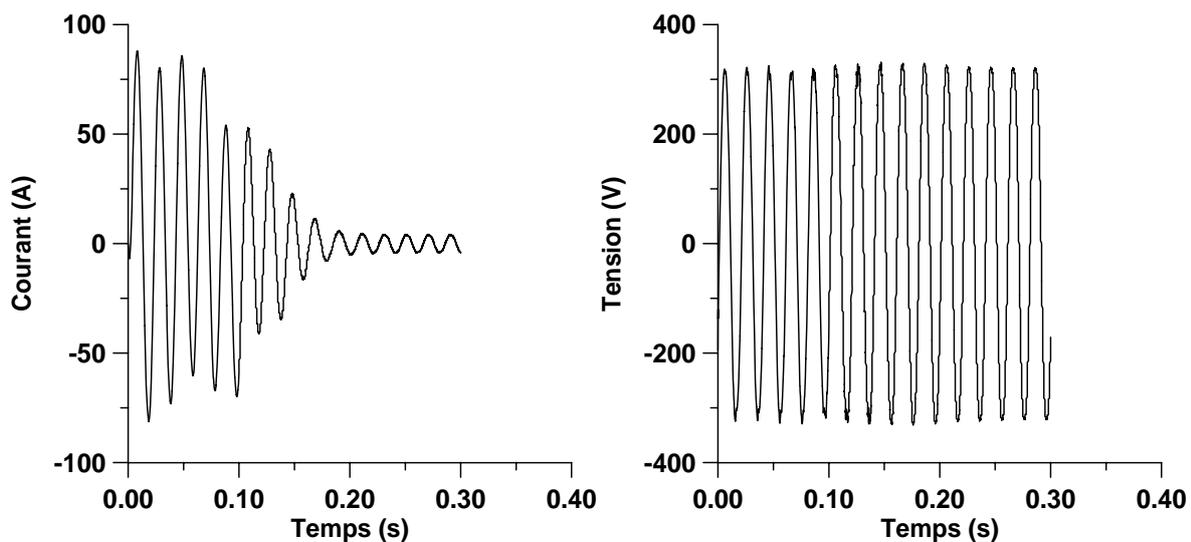


Figure 4.6. Courant de démarrage et tension simple correspondante au moteur M1

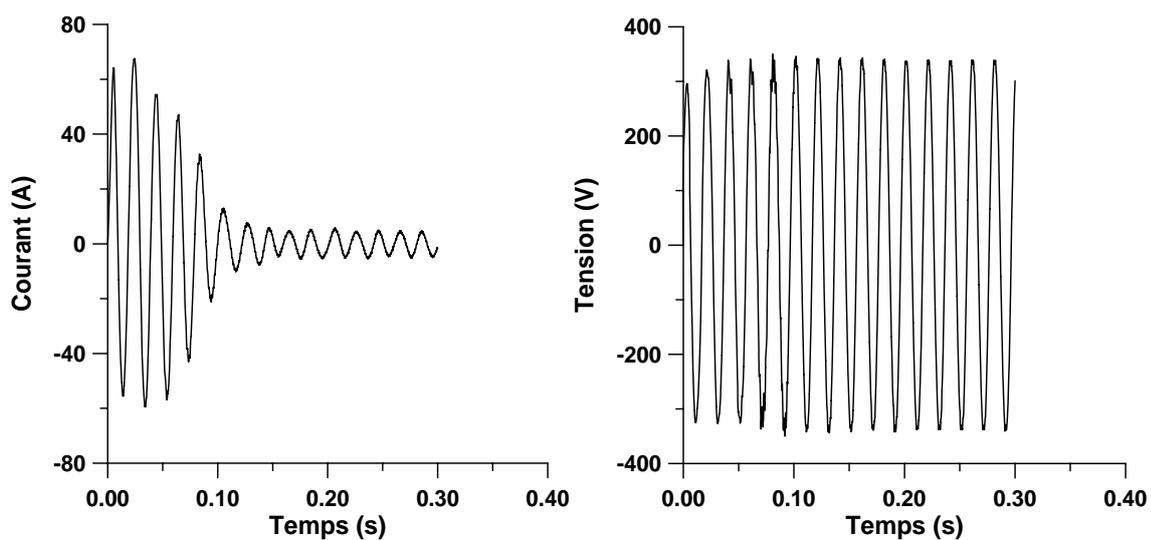


Figure 4.7. Courant de démarrage et tension simple correspondante au moteur M2

4.4.4 Testes et Résultats

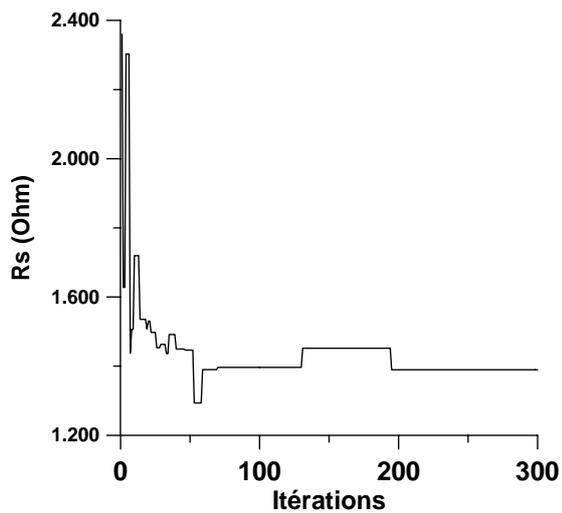
L'identification des moteurs M1 et M2 conduit aux paramètres donnés par le tableau 4.2.

Paramètres	Moteur M1	Moteur M2
l_{fs} (mH)	0.003474	0.005264
l_{fr} (mH)	0.003474	0.005264
R_s (Ω)	1.434748	1.627461
R_r (Ω)	1.004574	1.187608
J (Kg/ms ²)	0.017380	0.031755
f_r (Mn s/rd)	0.002191	0.008449
C_a	2.540421	1.094901
C_b	0.080078	0.235626
C_1	0.051131	0.280636
C_2	1.894404	0.716482
C_3	0.050215	0.006928
C_4	0.759174	1.320077
C_5	0.044577	0.048263
C_6	2.020351	1.574947
C_7	0.055575	0.031222
C_8	1.403229	1.890875

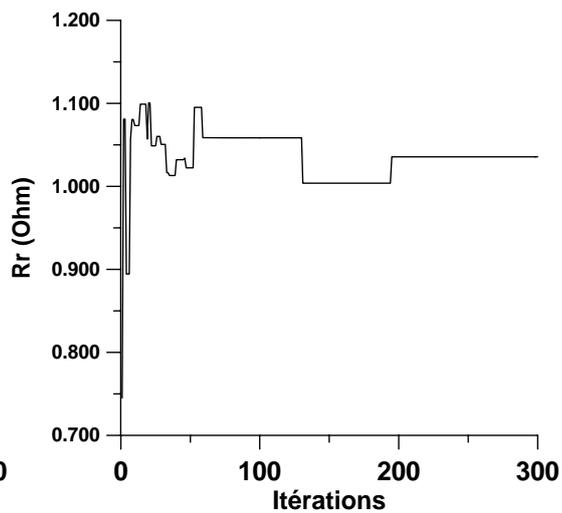
Tableau 4.2. Paramètres estimés avec saturation des moteurs M1 et M2

Les figure de 4.8 (a,b,c,d,e) représentent l'évolution des paramètres du moteur M1 en fonction des itérations.

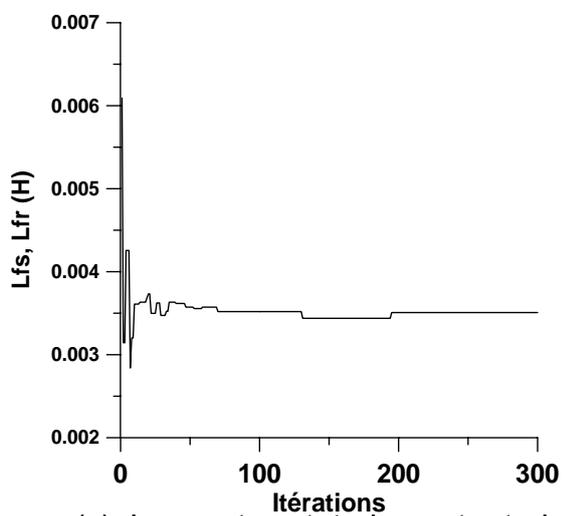
Pour le moteur M2, l'évolution des paramètres sont données en Appendice B.



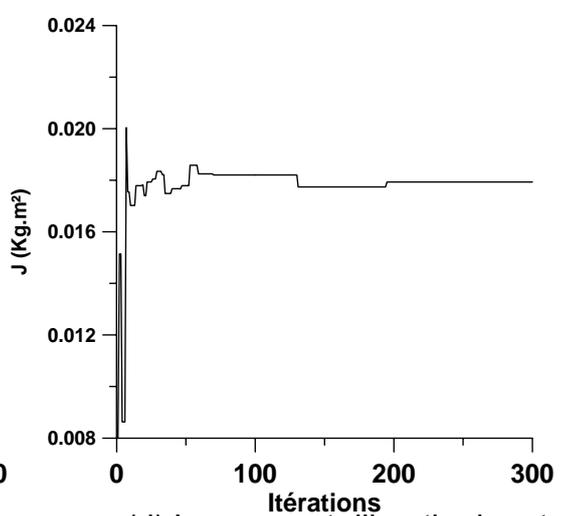
(a) La résistance statorique



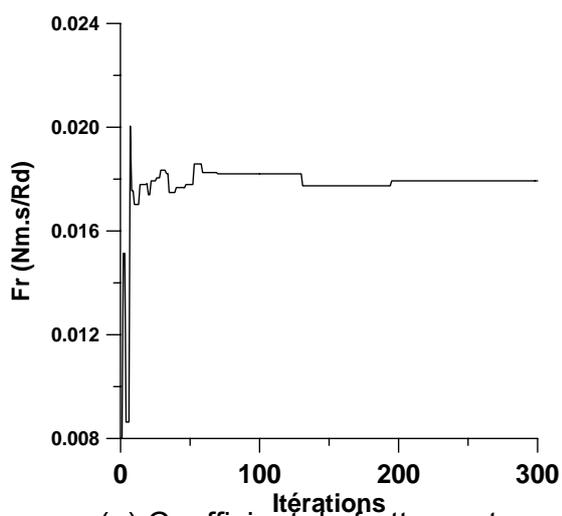
(b) La résistance rotorique



(c) Les pertes statorique et rotorique



(d) Le moment d'inertie du rotor



(e) Coefficient de frottement

Figure 4.8 L'évolution des paramètres du moteur M1 avec saturation

Les figures 4.9 et 4.10 représentent les variations des courbes de magnétisation, des inductances magnétisantes ainsi que des inductances dynamiques des moteurs M1 et M2 en fonction du courant magnétisant. Les inductances dynamiques et magnétisantes sont déduites à partir de la courbe de magnétisation identifiée en utilisant les équations (4.16)

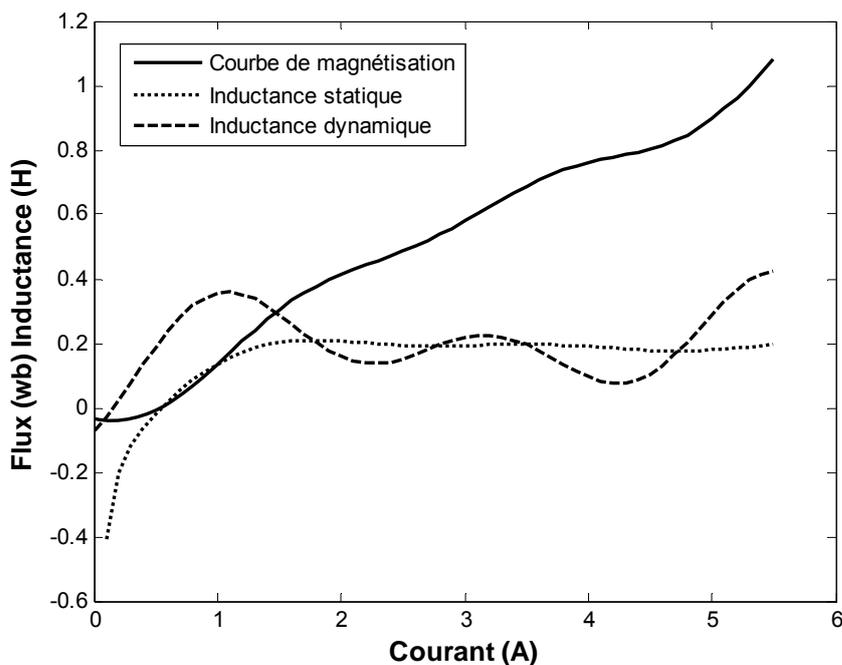


Fig.4.9 Courbe de magnétisation, inductances statique et dynamique du moteur M1

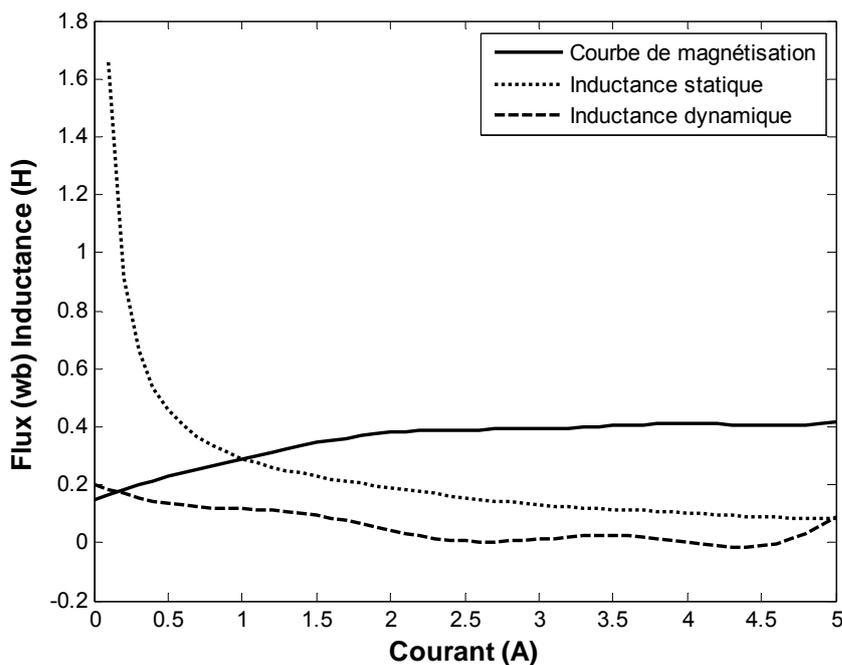


Fig.4.10 Courbe de magnétisation, inductances statique et dynamique du moteur M2

Les figures 4.11 et 4.12 montrent la superposition entre les courants mesurés et calculés.

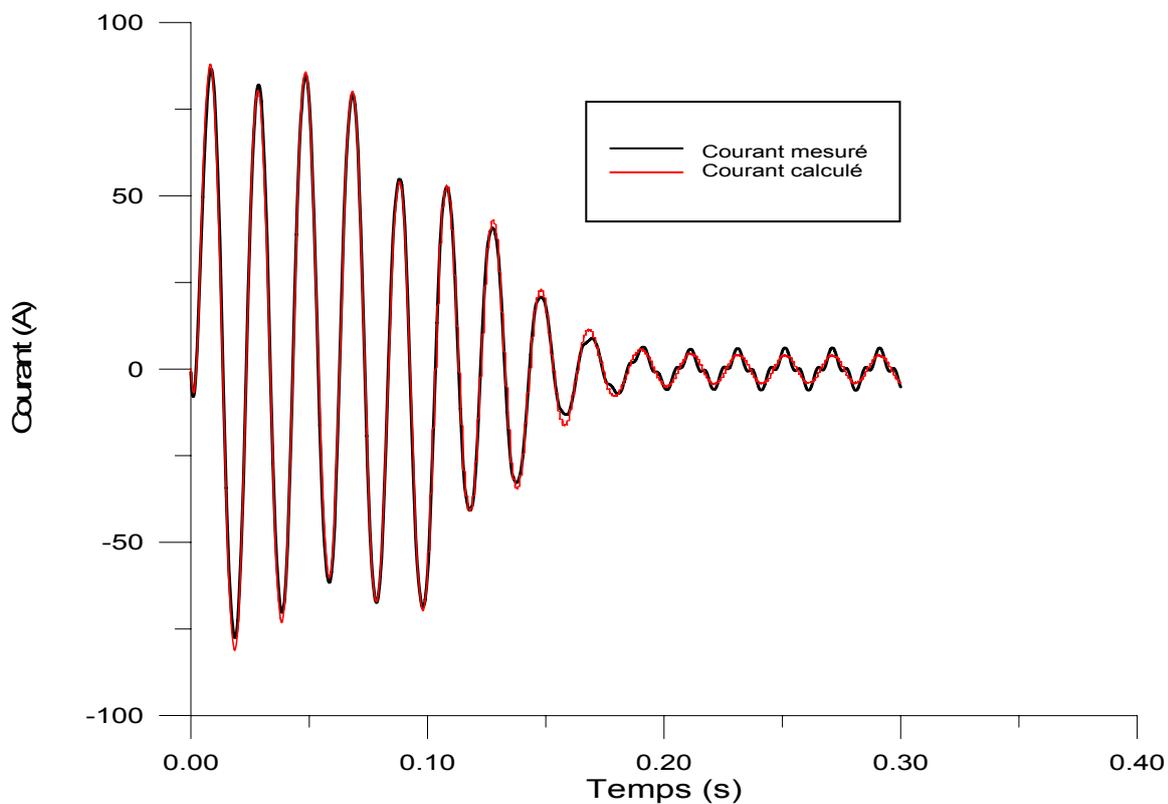


Fig. 4.11 La superposition des courants mesuré et calculé pour le moteur M1

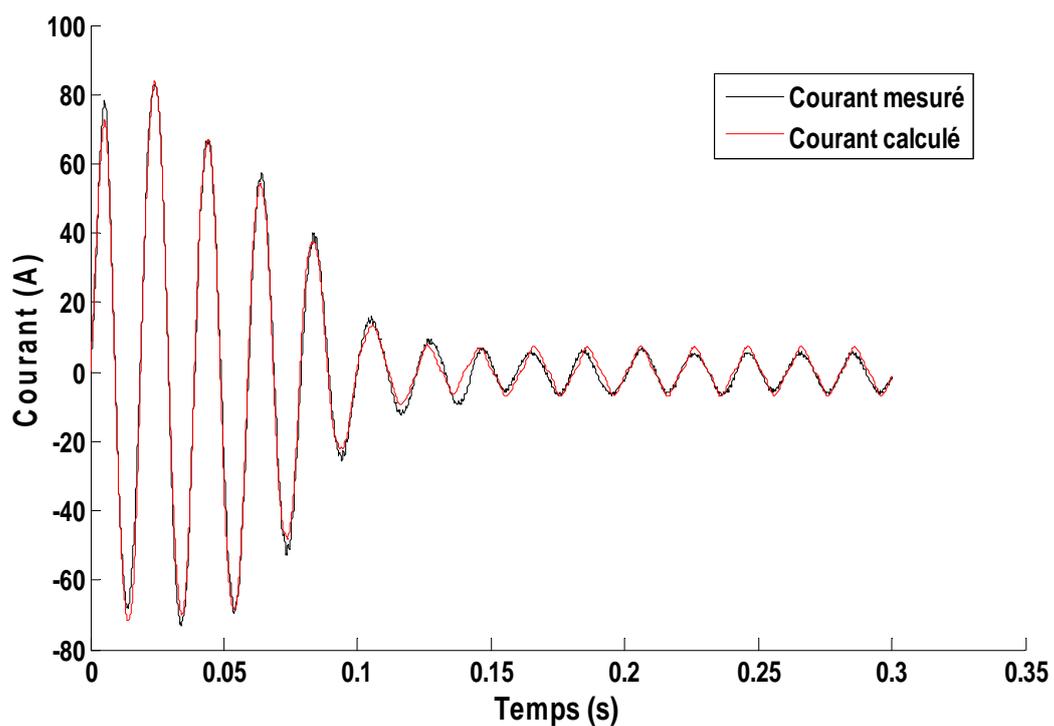


Fig. 4.12 La superposition des courants mesuré et calculé pour le moteur M2

Les erreurs absolues entre les courants mesurés et calculés à l'aide des paramètres estimés en fonction du temps. Les figures 4.13 et 4.14 montrent que l'écart entre les courants mesurés et les courants calculés est plus important en régime transitoire qu'en régime permanent

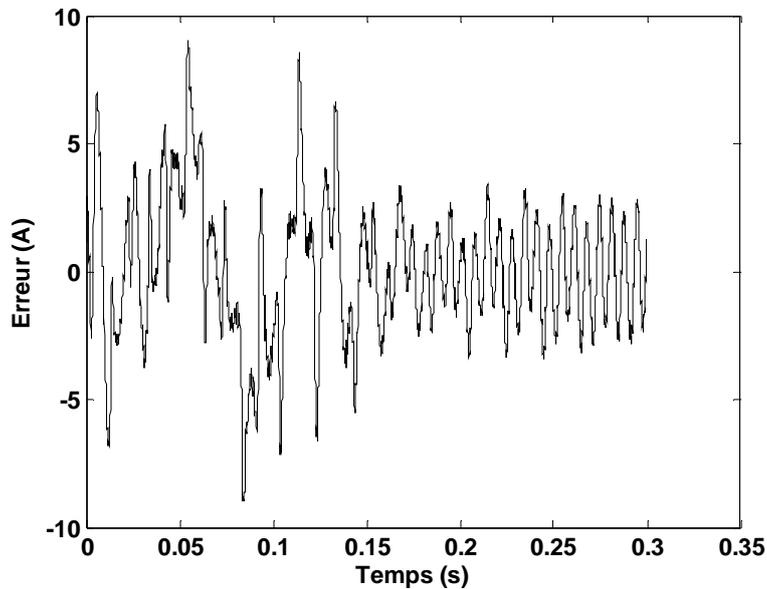


Figure 4.13. Erreur instantané entre le courant mesuré et le courant calculé
Du moteur M1 avec le modèle sans saturation.

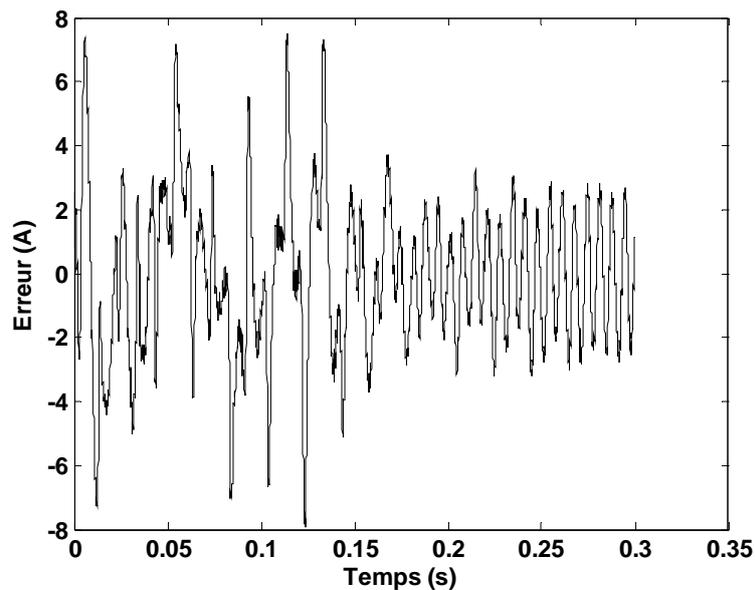


Figure 4.14. Erreur instantané entre le courant mesuré et le courant calculé
Du moteur M1 avec le modèle avec saturation

4.5 Comparaisons

Le tableau 4.3 donne une comparaison de l'erreur quadratique obtenue par le modèle sans saturation et le modèle avec saturation pour les deux moteurs M1 et M2.

	Modèle sans saturation	Modèle avec saturation	Taux d'amélioration
Moteur M1	45715.797	34746.562	24%
Moteur M2	62196.774	38526.450	49%

Tableau 4.3 comparaison de l'erreur quadratique entre le modèle sans saturation et Le modèle avec saturation.

La courbe de magnétisation du moteur M2 montre clairement le phénomène de la saturation, ce résultat est vérifié par une diminution de 49% de l'erreur quadratique pour le moteur M2 en considérant l'effet de la saturation.

Un autre point très important, c'est l'effort du calcul dans les deux modèles. On constate clairement que le deuxième modèle demande un effort de calcul plus grand vu que les inductances statique et dynamique et par la suite les matrices R, L et surtout l'inverse de la matrice L sont recalculées quatre fois dans la méthode de Range Kota, plus encore, la taille de l'essaim dans l'algorithme PSO augmente vu l'augmentation des paramètres à déterminer, tout cela augmentent le temps de calcul par itération (70 secondes sur un processeur P4 2 GHz).

Par contre, le premier modèle ne demande pas des efforts de calcul, donc un temps très réduit par itération (5 secondes sur un processeur P4 2 GHz).

Mais la remarque la plus importante est les valeurs des paramètres trouvés par le premier modèle sont très proches des paramètres trouvés par le deuxième modèle, donc l'hybridation des deux modèles est nécessaire pour une recherche rapide et efficace.

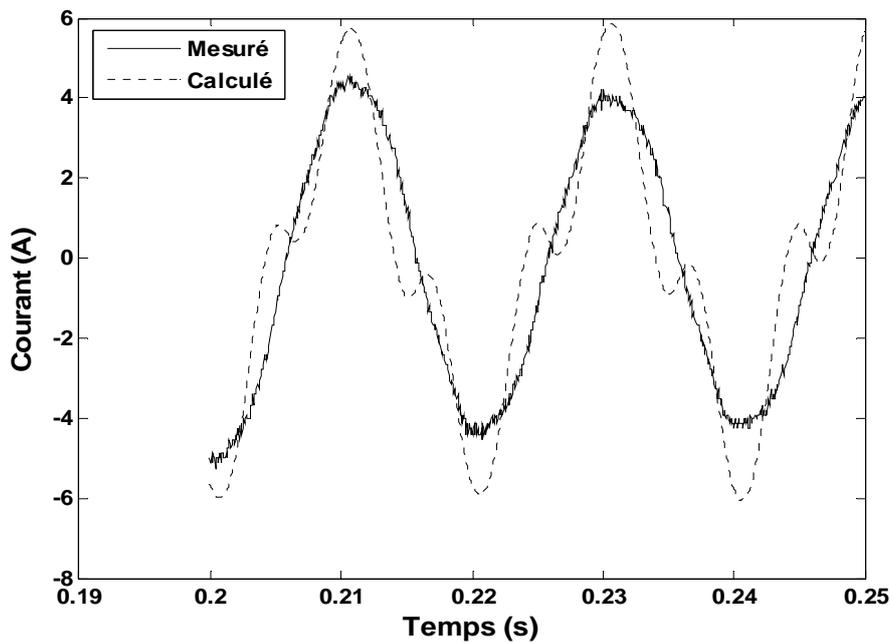


Figure 4.15 Superposition du courant mesuré et calculé sans saturation du moteur M1 en régime permanent

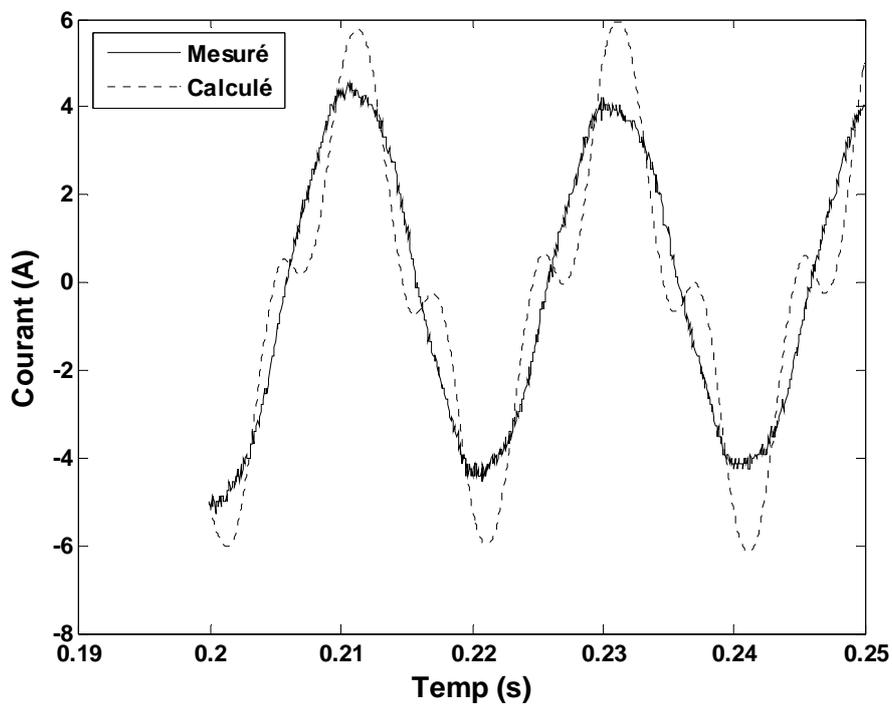


Figure 4.16. Superposition du courant mesuré et calculé avec saturation du moteur M1 en régime permanent

4.6 Conclusion

Après un bref aperçu sur le phénomène de la saturation, nous avons présenté un modèle de la machine asynchrone en régime saturé, mais qui ne tient compte que du premier harmonique d'espace. Nous avons ensuite validé notre approche d'identification en la testant sur deux moteurs asynchrones. Les résultats obtenus sont satisfaisants. Mais un modèle qui tien compte des harmonies d'espace représente mieux la machine asynchrone, comme le montre les résultats obtenus.

CONCLUSION

Le travail présenté est une contribution à l'identification paramétrique de la machine asynchrone. Il s'appuie sur les trois étapes suivantes :

- Le choix de la méthode d'optimisation.
- le choix du modèle de la machine.
- le choix des signaux d'entrée-sortie.

Dans notre approche, les paramètres électriques et mécaniques sont déterminés simultanément à l'aide de la méthode du modèle de référence. Le courant de démarrage et la tension simple correspondante sont considérés comme des signaux d'entrée-sortie, tandis que l'erreur quadratique entre les courants mesurés et ceux calculés est choisie comme critère de similitude. Quant à la machine, nous avons adopté deux modèles ayant quatre paramètres électriques et deux paramètres mécaniques.

Nous avons testés notre approche d'identification en utilisant trois algorithmes:

- PSO standard,
- PSO avec deux structures de voisinage
- PSO avec coefficient de suivi.

Nous avons alors validé notre approche à l'aide des données simulées, puis des données expérimentales pour identifier différents moteurs asynchrones. Nous avons vérifié que les trois algorithmes convergents vers le même vecteur paramètres. Ce résultat nous permet donc de confirmer l'unicité de la solution.

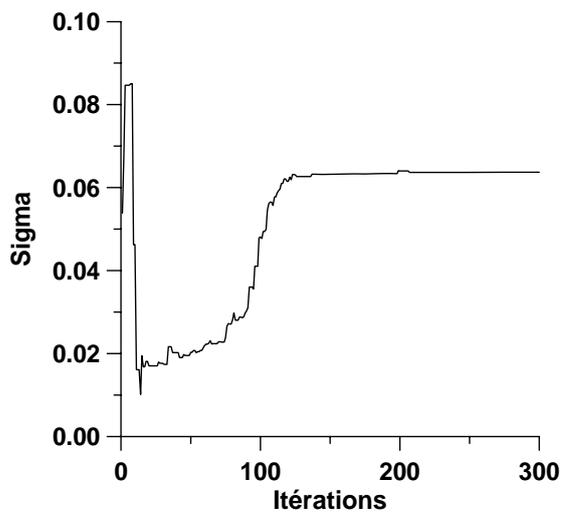
Cette nouvelle méthode d'optimisation (PSO) est bien adaptée à l'identification de la machine asynchrone qui est régie par un système d'équations non linéaires ne permettant pas l'obtention d'une expression mathématique de l'erreur quadratique.

En deuxième étape nous avons développé un modèle qui tient compte de la non linéarité de la courbe de la magnétisation, et nous avons utilisé notre approche d'identification pour déterminer simultanément les paramètres électriques, mécaniques et magnétiques à partir de la seule mesure du courant et de la tension simple correspondante.

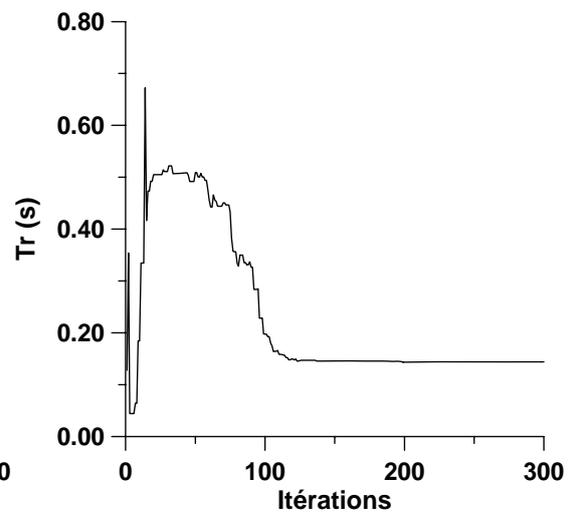
Les résultats d'identification obtenus sur deux moteurs asynchrones triphasés attestent la validité de la méthode ainsi que sa convergence.

L'accessibilité et la facilité des mesures nécessaires à notre approche ainsi que le temps de calcul nous encouragent d'envisager son application pour le diagnostic préventif des machines asynchrone.

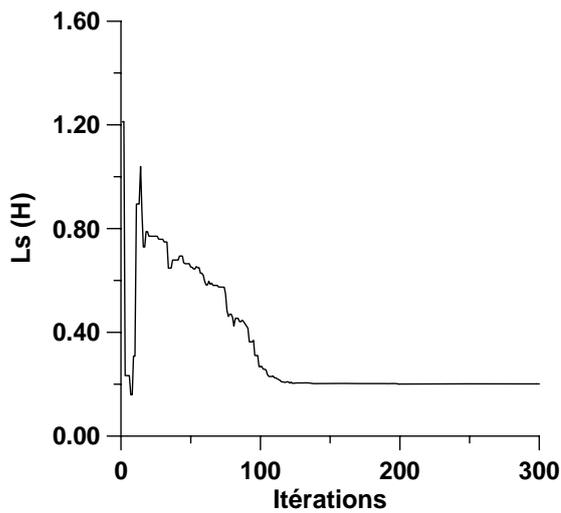
APPENDICE A
EVOLUTION DES PARAMETRES DU MOTEUR M2
SANS SATURATION



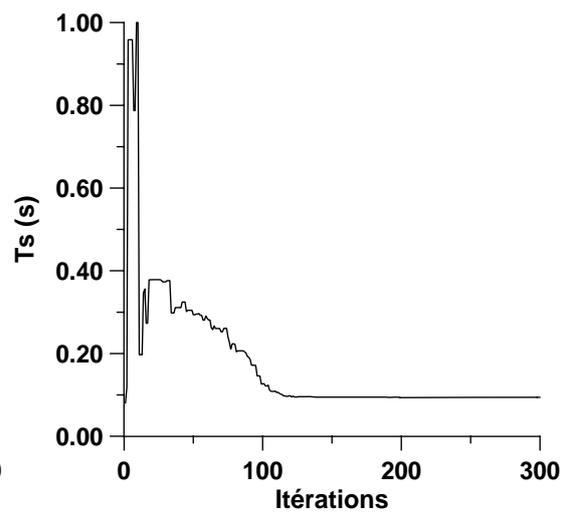
(a) Coefficient de fuite



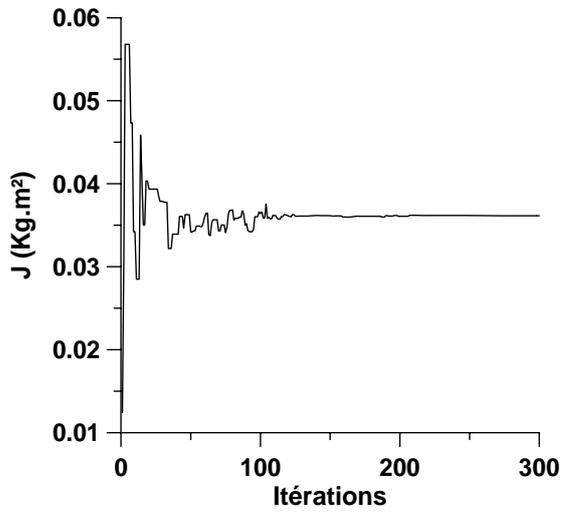
(b) Constante de temps rotorique



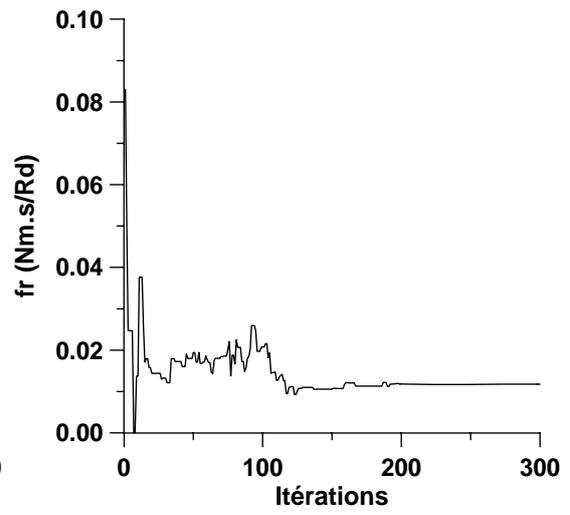
(c) Inductance propre statorique



(d) Constante de temps statorique

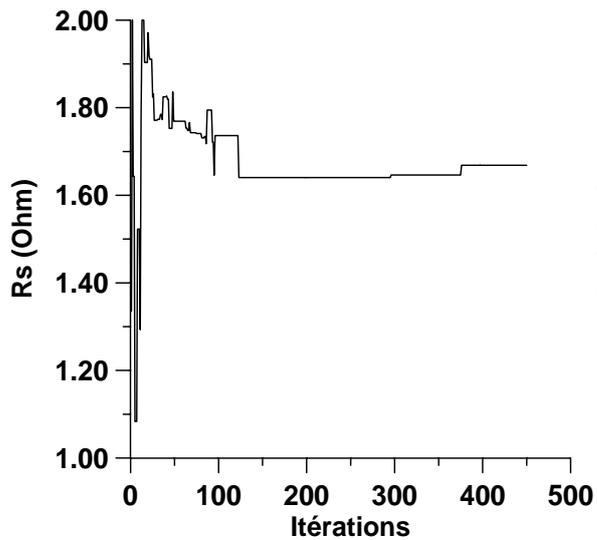


(e) Moment d'inertie du rotor

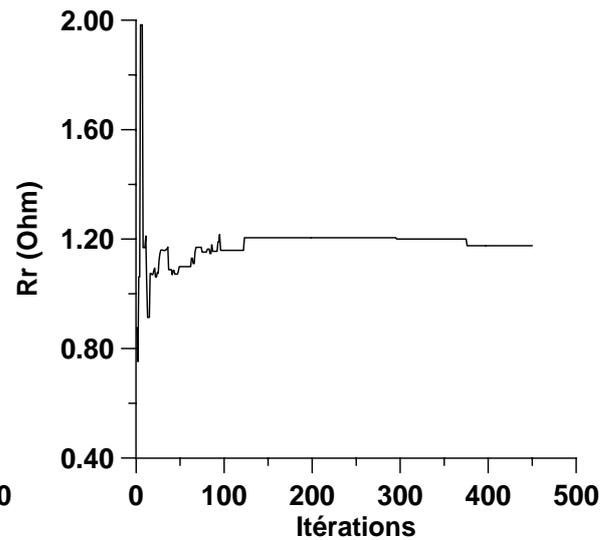


(f) Coefficient de frottement

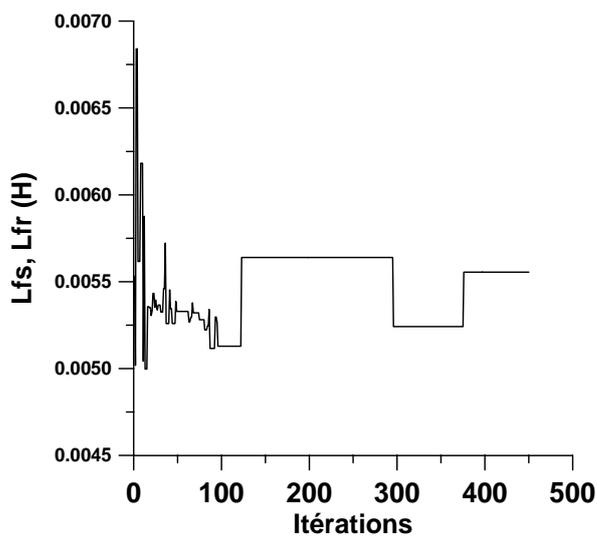
APPENDICE B
EVOLUTION DES PARAMETRES DU MOTEUR M2
AVEC SATURATION



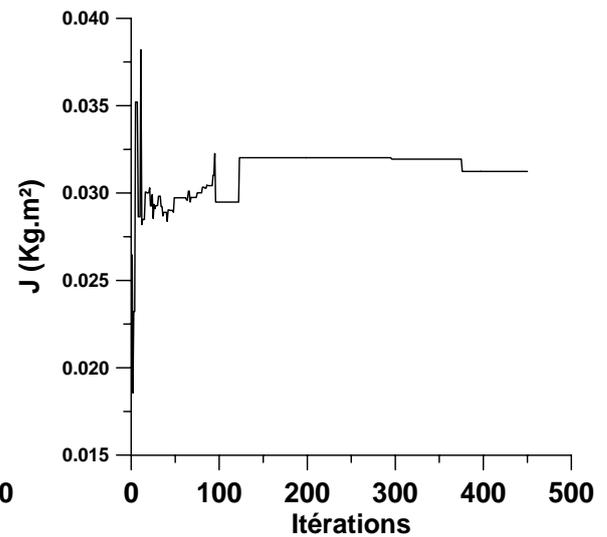
(b) L'évolution de la résistance statorique.



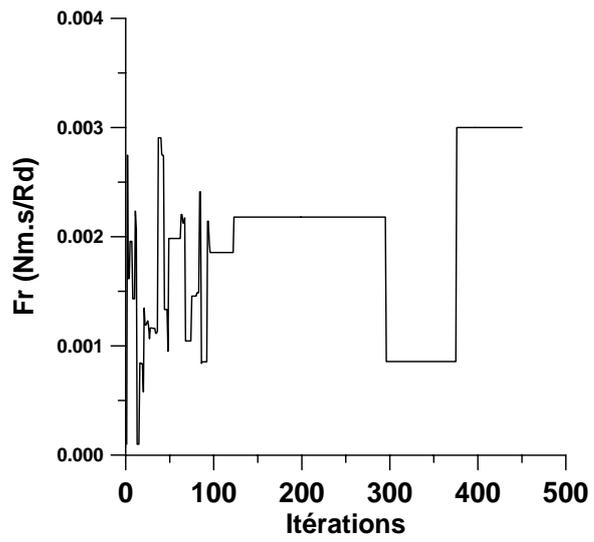
(b) L'évolution de la résistance rotorique.



(c) L'évolution de l'inductance de fuite.



(d) L'évolution du couple magnétique.



(e) L'évolution du coefficient de frottement.

APPENDICE C
CALCUL DE $\frac{d\Phi_{dm}}{dt}$

Sachant que $\Phi_{dm}=L_m I_{dm}$ et $\Phi_m=L_m I_m$ nous pouvons écrire que :

$$\frac{d\Phi_{dm}}{dt} = \frac{d}{dt} \left(\Phi_m \frac{I_{dm}}{I_m} \right) = \frac{d\Phi_m}{dt} \frac{I_{dm}}{I_m} + \Phi_m \frac{d}{dt} \left(\frac{I_{dm}}{I_m} \right) \quad (\text{C.1})$$

L'expression précédente peut s'écrire sous la forme :

$$\frac{d\Phi_{dm}}{dt} = \frac{d\Phi_m}{dI_m} \frac{dI_m}{dt} \frac{I_{dm}}{I_m} + \Phi_m \frac{\frac{dI_{dm}}{dt} I_m - \frac{dI_m}{dt} I_{dm}}{I_m^2} \quad (\text{C.2})$$

Sachant que $I_m = \sqrt{I_{dm}^2 + I_{qm}^2}$, donc $\frac{dI_m}{dt} = \frac{I_{dm} \frac{dI_{dm}}{dt} + I_{qm} \frac{dI_{qm}}{dt}}{I_m}$ (C.3)

Posons :
$$\begin{cases} L = \frac{d\phi_m}{dI_m} \\ L_m = \frac{\phi_m}{I_m} \end{cases} \quad (\text{C.4})$$

L et L_m représentent respectivement l'inductance dynamique et l'inductance statique (ou magnétisante).

$$\frac{d\Phi_{dm}}{dt} = L \left(\frac{I_{dm}}{I_m} \frac{dI_{dm}}{dt} + \frac{I_{qm}}{I_m} \frac{dI_{qm}}{dt} \right) \frac{I_{dm}}{I_m} + L_m \left(\frac{dI_{dm}}{dt} - \frac{I_{qm}}{I_m} \frac{dI_m}{dt} \right) \quad (\text{C.5})$$

$$\frac{d\Phi_{dm}}{dt} = L \left(\frac{I_{dm}}{I_m} \frac{dI_{dm}}{dt} + \frac{I_{qm}}{I_m} \frac{dI_{qm}}{dt} \right) \frac{I_{qm}}{I_m} + L_m \frac{dI_{qm}}{dt} - L_m \frac{I_{qm}}{I_m} \left(\frac{I_{dm}}{I_m} \frac{dI_{dm}}{dt} + \frac{I_{qm}}{I_m} \frac{dI_{qm}}{dt} \right) \quad (\text{C.6})$$

Désignons par μ désigne l'angle entre le courant magnétisant I_m et l'axe d.

$$\begin{cases} \cos\mu = \frac{I_{md}}{I_m} \\ \sin\mu = \frac{I_{mq}}{I_m} \end{cases} \quad (\text{C.7})$$

$$\frac{d\Phi_{dm}}{dt} = L \left(\cos\mu \frac{dI_{dm}}{dt} + \sin\mu \frac{dI_{qm}}{dt} \right) \sin\mu + L_m \frac{dI_{qm}}{dt} - L_m \sin\mu \left(\cos\mu \frac{dI_{dm}}{dt} + \sin\mu \frac{dI_{qm}}{dt} \right) \quad (\text{C.8})$$

$$\frac{d\Phi_{dm}}{dt} = (L - L_m) \cos\mu \sin\mu \frac{dI_{dm}}{dt} + (L - L_m) (\sin\mu)^2 \frac{dI_{qm}}{dt} + L_m \frac{dI_{qm}}{dt} \quad (\text{C.9})$$

$$\frac{d\Phi_{dm}}{dt} = (L - L_m) \cos\mu \sin\mu \frac{dI_{dm}}{dt} + \left(L (\sin\mu)^2 + L_m (\cos\mu)^2 \right) \frac{dI_{qm}}{dt} \quad (\text{C.10})$$

L'expression précédente peut se mettre sous la forme suivante :

$$\frac{d\Phi_{dm}}{dt} = L_{ddm} \frac{dI_{dm}}{dt} + L_{qdm} \frac{dI_{qm}}{dt} \quad (\text{C.11})$$

$$\text{avec } L_{qdm} = (L - L_m) \cos\mu \sin\mu = \frac{L - L_m}{2} \sin 2\mu \quad (\text{C.12})$$

En posant que :

$$\frac{L - L_m}{2} = L_2 \quad \text{et } L_{2s} = L_2 \sin 2\mu \quad \text{nous aurons } L_{ddm} = L_{2s} \quad (\text{C.13})$$

$$L_{ddm} = L (\sin\mu)^2 + L_m (\cos\mu)^2 = L + (L_m - L) (\cos\mu)^2 = L + (L_m - L) \left(\frac{1 + \cos 2\mu}{2} \right) \quad (\text{C.13})$$

$$L_{qdm} = \frac{L + L_m}{2} - \frac{L - L_m}{2} \cos 2\mu \quad (\text{C.14})$$

$$\text{En posant que } \frac{L + L_m}{2} = L_o \quad \text{et } L_{2c} = L_2 \cos 2\mu \quad \text{nous aurons } L_{qdm} = L_o - L_{2c} \quad (\text{C.15})$$

APPENDICE D

CALCUL DE $\frac{d\Phi_{qm}}{dt}$

Sachant que $\Phi_{qm}=L_m I_{qm}$ et $\Phi_m=L_m I_m$ nous pouvons écrire que :

$$\frac{d\Phi_{qm}}{dt} = \frac{d}{dt} \left(\Phi_m \frac{I_{qm}}{I_m} \right) = \frac{d\Phi_m}{dt} \frac{I_{qm}}{I_m} + \Phi_m \frac{d}{dt} \left(\frac{I_{qm}}{I_m} \right) \quad (D.1)$$

L'expression précédente peut s'écrire sous la forme :

$$\frac{d\Phi_{qm}}{dt} = \frac{d\Phi_m}{dI_m} \frac{dI_m}{dt} \frac{I_{qm}}{I_m} + \Phi_m \frac{\frac{dI_{qm}}{dt} I_m - dI_m \frac{I_{qm}}{dt}}{I_m^2} \quad (D.2)$$

Sachant que $I_m = \sqrt{I_{dm}^2 + I_{qm}^2}$ donc $\frac{dI_m}{dt} = \frac{I_{dm} \frac{dI_{dm}}{dt} + I_{qm} \frac{dI_{qm}}{dt}}{I_m}$ (D.3)

Posons :
$$\begin{cases} L = \frac{d\phi_m}{dI_m} \\ L_m = \frac{\phi_m}{I_m} \end{cases} \quad (D.4)$$

L et L_m représentent respectivement l'inductance dynamique et l'inductance statique (magnétisante).

$$\frac{d\Phi_{qm}}{dt} = L \left(\frac{I_{dm}}{I_m} \frac{dI_{dm}}{dt} + \frac{I_{qm}}{I_m} \frac{dI_{qm}}{dt} \right) \frac{I_{qm}}{I_m} + L_m \left(\frac{dI_{qm}}{dt} - \frac{I_{qm}}{I_m} \frac{dI_m}{dt} \right) \quad (D.5)$$

$$\frac{d\Phi_{qm}}{dt} = L \left(\frac{I_{dm}}{I_m} \frac{dI_{dm}}{dt} + \frac{I_{qm}}{I_m} \frac{dI_{qm}}{dt} \right) \frac{I_{qm}}{I_m} + L_m \frac{dI_{qm}}{dt} - L_m \frac{I_{qm}}{I_m} \left(\frac{I_{dm}}{I_m} \frac{dI_{dm}}{dt} + \frac{I_{qm}}{I_m} \frac{dI_{qm}}{dt} \right) \quad (D.6)$$

Puisque μ désigne toujours l'angle entre I_m et l'axe d , nous obtenons :

$$\frac{d\Phi_{qm}}{dt} = L \left(\cos \mu \frac{dI_{dm}}{dt} + \sin \mu \frac{dI_{qm}}{dt} \right) \sin \mu + L_m \frac{dI_{qm}}{dt} - L_m \sin \mu \left(\cos \mu \frac{dI_{dm}}{dt} + \sin \mu \frac{dI_{qm}}{dt} \right) \quad (D.7)$$

$$\frac{d\Phi_{qm}}{dt} = (L-L_m)\cos\mu\sin\mu\frac{dI_{dm}}{dt} + (L-L_m)(\sin\mu)^2\frac{dI_{qm}}{dt} + L_m\frac{dI_{qm}}{dt} \quad (\text{D.8})$$

$$\frac{d\Phi_{qm}}{dt} = (L-L_m)\cos\mu\sin\mu\frac{dI_{dm}}{dt} + (L(\sin\mu)^2 + L_m(\cos\mu)^2)\frac{dI_{qm}}{dt} \quad (\text{D.9})$$

L'expression précédente peut se mettre sous la forme suivante :

$$\frac{d\Phi_{qm}}{dt} = L_{dqm}\frac{dI_{dm}}{dt} + L_{qqm}\frac{dI_{qm}}{dt} \quad (\text{D.10})$$

$$\text{Avec : } L_{dqm} = (L-L_m)\cos\mu\sin\mu = \frac{L-L_m}{2}\sin 2\mu \quad (\text{D.11})$$

$$\text{En posant que: } \frac{L-L_m}{2} = L_2 \quad \text{et } L_{2s} = L_2\sin 2\mu \quad \text{nous avons } L_{dqm} = L_{2s} \quad (\text{D.12})$$

$$L_{qqm} = L(\sin\mu)^2 + L_m(\cos\mu)^2 = L + (L_m - L)(\cos\mu)^2 = L + (L_m - L)\left(\frac{1 + \cos 2\mu}{2}\right) \quad (\text{D.13})$$

$$L_{qqm} = \frac{L+L_m}{2} - \frac{L-L_m}{2}\cos 2\mu \quad (\text{D.14})$$

$$\text{En posant que : } \frac{L+L_m}{2} = L_o \quad \text{et } L_{2c} = L_2\cos 2\mu, \quad \text{donc } L_{qqm} = L_o - L_{2c} \quad (\text{D.15})$$

Finalement $\frac{d\Phi_{dm}}{dt}$ et $\frac{d\Phi_{qm}}{dt}$ s'écrivent :

$$\begin{cases} \frac{d\Phi_{dm}}{dt} = (L_o + L_{2c})\frac{dI_{dm}}{dt} + L_{2s}\frac{dI_{qm}}{dt} \\ \frac{d\Phi_{qm}}{dt} = L_{2s}\frac{dI_{dm}}{dt} + (L_o + L_{2c})\frac{dI_{qm}}{dt} \end{cases} \quad (\text{D.16})$$

REFERENCES

1. G. Berthiau et P. Siarry, *Etat de l'art des méthodes d'optimisation globale*, RAIRO-Operations Research, 35 (2001) 329-365.
2. M. Minoux « Programmation mathématique théorie et algorithmes » collection technique et scientifique des télécommunications Ed. Dunod, Paris 1983.
3. C.H. Papadimitriou, K. Steiglitz, *Combinatorial optimization – algorithms and complexity*. Prentice Hall, 1982.
4. C.C. Ribeiro, N. Maculan (Eds.), Applications of combinatorial optimization. *Annals of Operations Research* 50, 1994.
5. M.R. Garey, D.S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness*, W.H. Freeman and Company, New York, 1979.
6. G. Laporte, I.H. Osman, *Metaheuristics in combinatorial optimization*, *Annals of Operations Research* 63, J.C. Baltzer Science Publishers, Basel, Switzerland, 1996.
7. I.H. OSMAN, J.P. KELLY (Eds.), *Meta-heuristics: theory and applications*, Kluwers Academic Publishers, Boston, 1996.
8. 93a] C.R. Reeves (Ed.) *Modern heuristic techniques for combinatorial problems*, Blackwell Scientific Publications, Oxford, 1993.
9. E.H.L. Aarts, J.K. Lenstra (Eds.), *Local search in combinatorial optimization*, John Wiley & Sons, 1997.
10. Eric D. Taillard and all “Adaptive memory programming a unified view of metaheuristics”. *European journal of operational research* 135 (2001) 1-16.
11. C.H. Papadimitriou, K. Steiglitz, *Combinatorial optimization – algorithms and complexity*. Prentice Hall, 1982.
12. F. Glover et M. Laguna, *Tabu Search* (Kluwer Academic Publishers, 1997).
13. P. Hansen, The steepest ascent mildest descent heuristic for combinatorial programming. Congress on Numerical Methods in Combinatorial Optimization, Capri, Italie, 1986.

14. F. Glover, Heuristics for integer programming using surrogate constraints. *Decision Sciences* 8: 156-166, 1977.
15. L.J. Fogel, A.J. Owens, M.J. Walsh, Artificial intelligence through simulated evolution, Wiley, New York, 1966.
16. H.P. Schwefel, Numerical optimization of computer models, John Wiley & Sons, 1981.
17. T. BÄCK, *Evolutionary algorithms in theory and practice*. Oxford University Press, New York, 1995.
18. J.H. Holland, Adaptation in natural and artificial systems. The University of Michigan Press, Ann Arbor, 1975
19. D.E. Goldberg, Genetic algorithms in search, optimization, and machine learning, Addison-Wesley, New York, 1989.
20. M. Dorigo, G. Di Caro, The ant colony optimization meta-heuristic. Dans D. Corne, M. Dorigo, F. Glover (Eds.), *New Ideas in Optimization*, McGraw-Hill, 1999.
21. T. Bäck, F. Hoffmeister, H-P. Schwefel, A Survey of evolutionary strategies. *Proc. of 4th Intl. Conference on Genetic Algorithms (ICGA'91)*, R. Belew, L. Booker (Eds.), Morgan Kaufmann, p. 2-9, 1991.
22. J. Kennedy et R.C. Eberhart, Particle Swarm Optimization, IEEE International Conference on Neural Networks, Perth, Australia, 1995.
23. M. Clerc et J. Kennedy, The Particle Swarm. Explosion, stability, and convergence in a multidimensional complex space, *IEEE Transactions on Evolutionary Computation*, 6 (2002) 58-73.
24. J. J. Liang and P. N. Suganthan « Dynamic Multi-Swarm Particle Swarm Optimizer” -7803-8916-6/05/ IEEE
25. J.J. Liang, A.K. Qin, P.N.Suganthan and S. Baskar “Particle Swarm (Optimization Algorithms with Novel Learning Strategies” 2004 IEEE International Conference on Systems, Man and Cybernetics
26. P. N. Sugant han “Particle Swarm Optimiser with Neighbourhood Operator” 0-7803-5536-9/99/ IEEE
27. S. Z. Zhao, P. N. Suganthan, “Dynamic Multi-Swarm Particle Swarm Optimizer with Local Search for Large Scale Global Optimization” 978-1-4244-1823-7/08/ IEEE
28. James Kennedy and Russell C. Eberhart “A discrete binary version of the particle swarm algorithm” 0-78034053-1/97/ IEEE

29. S.Baskar P.N.Suganthan "A Novel Concurrent Particle Swarm Optimization" 0-7803-851 5-2/04/ IEEE
30. James Kennedy and Rui Mendes "Neighborhood Topologies in Fully Informed and Best-of-Neighborhood Particle Swarms" IEEE transactions on systems, man, and cybernetics—part c: applications and reviews, vol. 36, no. 4, july 2006
31. J. J. Liang P. N. Suganthan "Dynamic Multi-Swarm Particle Swarm Optimizer with Local Search" 0-7803-9363-5/05/ IEEE.
32. Maurice Clerc and James Kennedy "The Particle Swarm—Explosion, Stability, and Convergence in a Multidimensional Complex Space" IEEE transactions on evolutionary computation, vol. 6, no. 1, February 2002
33. Stefan Janson and Martin Middendorf, "A Hierarchical Particle Swarm Optimizer and Its Adaptive Variant" IEEE transactions on systems, man, and cybernetics—part b: cybernetics, vol. 35, no. 6, December 2005
34. Shubham Agrawal, Yogesh Dashora "Interactive Particle Swarm: A Pareto-Adaptive Metaheuristic to Multiobjective Optimization", IEEE transactions on systems, man, and cybernetics—part a: systems and humans, vol. 38, no. 2, march 2008
35. Frans van den Bergh and Andries P. Engelbrecht, "A Cooperative Approach to Particle Swarm Optimization" IEEE transactions on evolutionary computation, vol. 8, no. 3, june 2004
36. Asanga Ratnaweera, Saman K. Halgamuge "Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients" IEEE transactions on evolutionary computation, vol. 8, no. 3, june 2004
37. Rui Mendes, "The Fully Informed Particle Swarm: Simpler, Maybe Better" IEEE transactions on evolutionary computation, vol. 8, no. 3, june 2004
38. Arvind S. Mohais Christopher Ward Christian Posthoff "Randomized Directed Neighborhoods with Edge in Particle Swarm Optimization" 0-7803-85 15-2/04/ IEEE
39. Shinn-Ying Ho "OPSO: Orthogonal Particle Swarm Optimization and Its Application to Task Assignment Problems" IEEE transactions on systems, man, and cybernetics part a: systems and humans, vol. 38, no. 2, march 2008
40. Daniel Parrott and Xiaodong Li, "Locating and Tracking Multiple Dynamic Optima by a Particle Swarm Model Using Speciation" IEEE transactions on evolutionary computation, vol. 10, no. 4, august 2006
41. Tony Huang, Student "A Hybrid Boundary Condition for Robust Particle Swarm Optimization" IEEE antennas and wireless propagation letters, vol.4,2005

42. S. L. Ho, Shiyong Yang, Guangzheng “A Particle Swarm Optimization-Based Method for Multiobjective Design Optimizations” IEEE transactions on magnetics, vol. 41, no. 5, may 2005
43. Stefano Selleri “Some Insight Over New Variations of the Particle Swarm Optimization Method” IEEE antennas and wireless propagation letters, vol.5, 2006
44. U. Baumgartner, Ch. Magele, and W. Renhart “Pareto Optimality and Particle Swarm Optimization” IEEE transactions on magnetics, vol. 40, no. 2, march 2004
45. Suomin Cui “Application of a Parallel Particle Swarm Optimization Scheme to the Design of Electromagnetic Absorbers” IEEE transactions on antennas and propagation, vol. 53, no. 11, november 2005
46. Azuma Ide And Keiichiro Yasuda “A Basic Study of Adaptive Particle Swarm Optimization” Electrical Engineering in Japan, Vol. 151, No. 3, 2005
47. Said Mikki and Ahmed Kishk “Improved particle swarm optimization technique using hard boundary conditions” Microwave and optical technology letters / vol. 46, no. 5, september 5 2005
48. Ying-Nan Zhang ,Qing-Ni Hu and Hong-Fei Teng “Active target particle swarm Optimization” Concurrency and computation: practice and experience Concurrency Computat.: Pract. Exper. 2007; 20:29–40
49. Byung-Il Koh, Alan D. George, Raphael “Parallel asynchronous particle swarm optimization” International journal for numerical methods in engineering 2006; 67:578–595
50. Seth t. Call, Dmitry yu. Zubarev, “Global Minimum Structure Searches via Particle Swarm Optimization” 2007 Wiley Periodicals, Inc. J Comput Chem 28: 1177–1186, 2007
51. Teruyoshi Yamaguchi, Nobuhiro Iwasaki, “Adaptive Particle Swarm Optimization Using Information about Global Best” Electrical Engineering in Japan, Vol. 159, No. 4, 2007
52. J. F. Schutte, and al. “Parallel global optimization with the particle swarm algorithm” International journal for numerical methods in engineering int. j. numer. meth. engng 2004; 61:2296–2315
53. Takashi Okamoto And Eitaro Aiyoshi “Global Optimization with the PSO Coupling-Type Discrete Gradient Chaos Model” Electrical Engineering in Japan, Vol. 165, No.4, 2008
54. Ying-Nan Zhang¹ and Hong-Fei Teng “Detecting particle swarm optimization” Concurrency and computation: practice and experience (2008).

55. M. Clerc * et P. Siarry « Une nouvelle métaheuristique pour l'optimisation difficile : la méthode des essais particuliers » J3eA - Vol. 3 - 7 (2004).
56. J. Kennedy, R. Eberhart et Y. Shi, *Swarm Intelligence* (Morgan Kaufmann Academic Press, 2001).
57. [41]J.Chatelain, "Machines électriques", Traité d'électricité, Volume X, Presses Polytechniques Romandes, Lausanne, 1983
58. Caron J.P., Hautier J.P., "Modélisation et commande de la machine asynchrone", Ed. Technip, 1995.
59. J.Lesenne, F.Notelet, G.Segulier, "Introduction à l'électrotechnique approfondie", Technique et Documentation, Paris 1981.
60. P.L.Alger, "Induction machines", Gordon and Breach Science Publishers, second edition 1970.
61. James V Beck and Keith A Woodbury "Inverse problems and parameter estimation: integration of measurements and analysis" Meas. Sci. Technol. **9** (1998) 839–847. Printed in the UK
62. H. GARNIER, M. GILSON, T. BASTOGNE « Identification de modèles paramétriques à temps continu. Méthodes, outil logiciel et avantages » Journées Identification et Modélisation Expérimentale JIME'2006 – Novembre – Poitiers
63. S.Salon, D.Burow, M.Debortoli, G.Slavik, " Effects of slot closure and magnetic saturation on induction machine behavior" IEEE, Vol.30 N°5. September 1994.
64. P.Vas, K.E.Hallenius and J.E.Brown, " Cross-Saturation in smooth air gap electrical machines ", IEEE Trans .Energy Convers, Vol.116, N°1,pp.103-112, 1986
65. Mekhtoub. S, "Machine asynchrone en régime saturé", Thèse de Doctorat d'état és-Sciences de l'ENP d'Alger.2001.
66. M.Abdel-Salam, S.Abou-Shadi, Y.Sayed, " Analys of induction motors fed from constant current source taking core-loss and saturation into consideration" Electric Machines and Power Sytems, pp 581-599, 1999.
67. J.E.Brown, K.P.Kovacs, P.Vas, "A method of including the effects of main flux saturation in the generalized equations of A.C.machines", IEEE Trans.on power Apparatus and Systems, vol Pas 102, n°1, January pp 96-103, 1983
68. J.O.Ojo. and T.Y.Lipo, "An improved model of saturated induction machines", IEEE Trans. On Industry Applications, Vol.26 N°2. march-april 1990.
69. M. Bounekhla "Contribution à l'identification paramétrique de la machine asynchrone" Thèse de Doctorat d'Etat, Ecole nationale polytechnique d'Alger, 2004