

UNIVERSITE SAAD DAHLAB DE BLIDA

Faculté de Technologie

Département d'Electronique

THESE DE DOCTORAT

En électronique

DEVELOPPEMENT DE METHODES HYBRIDES EN VUE DE L'OPTIMISATION D'UN PROCESSUS, APPLICATION A LA MACHINE ASYNCHRONE

Par

DIF Naas

Devant le jury composé de

M. MAAMOUN	Maître de Conférences (A), USD de Blida	Président
R. Bradai	Maitre de Conférences (A), USD de Blida	Examineur
M. Rezaoui	Professeur, UZA de Djelfa	Examineur
B. MHAMDI	Maître de Conférences (A), UZA de Djelfa	Examineur
A. Rabhi	Maître de Conférences (A), UAY de Tissemsilt	Examineur
M. BOUNEKHLA	Professeur, USD de Blida	Directeur de thèse
E. BOUDISSA	Maitre de Conférences (A), USD de Blida	Invitée

Blida, 2022

RESUME

L'objectif de ce travail est l'amélioration et l'hybridation de l'algorithme de luciole standard afin d'utiliser à l'identification paramétrique de la machine asynchrone en considérant le modèle à une cage et le modèle à deux cages. Cet algorithme souffre à la convergence rapide ou prématurée vers un optimum local dû au manque de diversité comme toutes les métaheuristiques. Pour y remédier, nous proposons six améliorations. La première proposition est une amélioration de l'algorithme standard par l'orientation de mouvement vers le centroïde global de l'essaim dans le cas où le mouvement est vers la luciole plus brante est un échec. Les cinq autres propositions sont des hybridations de l'algorithme de luciole avec l'optimiseur de loups gris et l'algorithme sinus cosinus respectivement. Les résultats obtenus en utilisant les algorithmes proposés sont meilleurs que l'algorithme standard.

L'approche suggérée pour l'identification permet de déterminer simultanément les paramètres électriques et mécaniques de la machine à l'aide de la méthode du modèle de référence, en utilisant uniquement le courant de démarrage et la tension simple correspondante comme signaux d'entrées-sorties.

Les paramètres de la machine sont déterminés par minimisation de l'erreur quadratique entre le courant mesuré et celui calculé à partir du modèle adopté de la machine. La validation des résultats obtenus à l'aide de données expérimentales relevées sur deux machines différentes, est confirmée par la bonne superposition des courants mesurés et ceux calculés par le modèle adopté de la machine.

ABSTRACT

The aim of this work is the improvement and hybridization of the standard firefly algorithm to use for parametric identification of the asynchronous machine considering the single cage model and the two cage model.

The standard firefly algorithm suffers from premature convergence due to the lack of diversity like all Metaheuristics. In order to remedy this, we propose six firefly algorithm improvements. The first proposal is an improvement of the standard algorithm by orienting the movement toward the global centroid of the swarm in the case where the movement towards the brighter firefly is a failure. The other five proposals are hybridizations of the firefly algorithm with the gray wolf optimizer and the sine cosine algorithm respectively. The results obtained with the suggested algorithms are superior to the standard algorithm.

The suggested approach for identification allows the simultaneous determination of the electrical and mechanical parameters of the machine using the reference model method, using only the starting current and the corresponding single voltage as input-output signals.

The machine parameters are determined by minimizing the square error between the measured current and the one calculated from the adopted machine model. The validation of the results obtained with experimental data from two different machines is confirmed by the good superposition of the measured currents and those calculated by the adopted model of the machine.

ملخص

الهدف من هذا العمل هو تحسين وتهجين خوارزمية اليراع القياسية لاستخدامها في حساب قيم عناصر الآلة اللامتزامنة باستعمال نموذجين الأول نموذج القفص الفردي والثاني نموذج القفصين.

تعاني خوارزمية اليراع القياسية من تقارب المبكر بسبب نقص التنوع في السرب. من أجل علاج هذا، نقترح ستة تحسينات في خوارزمية اليراع القياسية. الاقتراح الأول هو تحسين الخوارزمية القياسية من خلال توجيه الحركة نحو مركز ثقل السرب في حالة فشل الحركة نحو اليراع الأكثر إشراقاً. المقترحات الخمسة الأخرى هي تهجين خوارزمية اليراع مع محسن الذئب الرمادي وخوارزمية جيب التمام الجيب على التوالي. النتائج التي تم الحصول عليها مع الخوارزميات المقترحة متفوقة على الخوارزمية القياسية.

يسمح النهج المقترح لتحديد الهوية بتحديد البارامترات الكهربائية والميكانيكية للآلة في نفس الوقت باستخدام طريقة النموذج المرجعي، باستخدام تيار البداية والجهد الفردي المقابل فقط كإشارات مدخلات وخرج.

يتم تحديد معلم الآلة عن طريق تقليل الخطأ المربع بين التيار المقاس والتيار المحسوب من نموذج الآلة المعتمد. يتم تأكيد التحقق من صحة النتائج التي تم الحصول عليها من خلال البيانات التجريبية من جهازين مختلفين من خلال التراكب الجيد للتيارات المقاسة وتلك المحسوبة بواسطة النموذج المعتمد للآلة.

REMERCIEMENTS

Les travaux de cette thèse ont été effectués au sein du Laboratoire de Systèmes Electriques et Télécommunications, du département d'Electronique, Université Saad Dahlab de Blida.

Cette thèse a été menée à terme sous la direction scientifique éclairée du Professeur **M'hamed BOUNEKHLA**. Qu'il trouve dans ces modestes lignes l'expression de ma profonde gratitude et de mon immense reconnaissance. Il a su joindre à sa grande expérience et à ses précieux conseils, une disponibilité et une gentillesse constante.

Je tiens à témoigner toute ma gratitude à Madame **Elghalia BOUDISSA** pour sa disponibilité et son aide.

Je rends hommage à Monsieur **M. MAAMOUN**, Maître de conférences (A) au département d'Electronique, de l'USDB, pour l'honneur qu'il me fait en présidant le jury d'examen de cette thèse.

Je remercie les membres de ce même jury pour leurs contributions. Je cite, par ordre alphabétique, Monsieur **A. RABHI**, Maître de conférences (A) à l'UAY de Tissemsilt, Monsieur **B. MHAMDI**, Maître de conférences (A) à l'UZA de Djelfa, Monsieur **M.RAZAOUI**, professeur à l'UZA de Djelfa, et Monsieur **R. BRADAI**, Maître de conférences (A) à l'USDB.

TABLE DES MATIERES

THESE DE DOCTORAT	0
RESUME	1
ABSTRACT	2
ملخص	3
REMERCIEMENTS	4
TABLE DES MATIERES	5
LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX	8
INTRODUCTION GENERALE	11
CHAPITRE 1 MODELISATION ET IDENTIFICATION DE LA MAS	15
1.1. Introduction	15
1.2. Principe de l'identification	15
1.3. Méthodes d'identification	16
1.3.1. Méthodes de base (Méthodes graphiques)	16
1.3.2. Méthode non récursive	17
1.3.3. Méthode récursive	17
1.3.4. Méthode du modèle	17
1.4. Modélisation de la Machine Asynchrone, Modèle à cage	18
1.4.1. Equations électriques	19
1.4.2. Equations magnétiques	20
1.4.3. Equation mécanique	21
1.4.4. Modèle de la machine dans le modèle de Park	21
1.4.5. Modèle à cinq paramètres électriques	23
1.4.6. Modèle à quatre paramètres électriques	24
1.5. Modélisation de la Machine Asynchrone, Modèle à double cage	24
1.5.1. Équations électriques	25
1.5.2. Équations des flux	26
1.5.3. Équation du couple	26
1.5.4. Modèle de Park	27
1.5.5. Modèle à sept paramètres	28
1.6. Identification paramétriques de la machine asynchrone	29
1.6.1. Identification basée sur les normes IEEE et IEC	29

1.6.2. Identification basée sur les catalogues/fabricants/plaques signalétiques	30
1.6.3. Identification basée sur les données mesurées	31
1.6.4. Identification basée sur les techniques d'optimisation	31
1.6.5. Méthode du modèle de référence	33
1.7. Conclusion	34
CHAPITRE 2 TECHNIQUES D'OPTIMISATION	35
2.1. Introduction	35
2.2. Classification des problèmes d'optimisation	35
2.3. Technique d'optimisation	35
2.4. Méthodes déterministes	36
2.4.1. Méthodes déterministes à base de gradient	36
2.4.2. Méthodes déterministes sans calcul de dérivées	38
2.5. Méthodes stochastiques	41
2.5.1. Méthodes à base de solution unique	42
2.5.2. Les métaheuristiques à population de solutions	44
2.6. Hybridation des algorithmes d'optimisation	49
2.6.1. Motivation de l'hybridation	50
2.6.2. Classification des stratégies d'hybridation	51
2.7. Conclusion	54
CHAPITRE 3 HYBRIDATION ET AMELIORATION DE L'ALGORITHME DE LUCIOLE	55
3.1. Introduction	55
3.2. Implémentation de l'algorithme de luciole	55
3.3. Problèmes de Test (Fonction de référence)	56
3.4. Amélioration de l'algorithme de luciole	58
3.4.1. Algorithme de luciole amélioré, proposition 01	59
3.4.2. Résultats d'optimisation avec les fonction de référence	60
3.5. Hybridation de l'algorithme de luciole avec l'optimiseur de loups gris	60
3.5.1. Optimiseur de loups gris	60
3.5.2. Algorithme hybride proposé, proposition 02	63
3.5.3. Résultats d'optimisation avec les fonctions de référence	64
3.6. Hybridation de l'algorithme de luciole avec l'algorithme sinus cosinus	64
3.6.1. Algorithme d'optimisation sinus cosinus	64
3.6.2. Algorithme hybride proposé, proposition 03	65

3.6.3.	Algorithme hybride proposé, proposition 04	66
3.6.4.	Résultats d'optimisation avec les fonctions de référence	67
3.7.	Hybridation de l'algorithme de luciole hybride avec l'OLG et l'ASC:	68
3.7.1.	Algorithme hybride proposé, proposition 05	68
3.7.2.	Algorithme hybride proposé, proposition 06	69
3.7.3.	Résultats d'optimisation avec les fonctions de référence	70
3.8.	Discussions des résultats	71
3.9.	Conclusion	78
CHAPITRE 4 APPLICATION A L'IDENTIFICATION PARAMETRIQUE DE LA MAS		80
4.1.	Introduction	80
4.2.	Identification à l'aide de modèle de référence.....	80
4.3.	Identification paramétrique de la machine asynchrone	81
4.3.1.	Identification l'aide des données simulées	81
4.3.2.	Identification l'aide des données expérimentales	85
4.3.3.	Résultats et Discussions.....	100
4.4.	Conclusion	107
CONCLUSION GENERALE		108
APPENDICE A		110
APPENDICE B		112
APPENDICE C		114
REFERENCES		116

LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX

FIGURE 1.1: STATOR DE LA MACHINE ASYNCHRONE	19
FIGURE 1.2 : REPRESENTATION DE LA MACHINE ASYNCHRONE TRIPHASEE ET BIPHASEE EQUIVALENTE.	22
FIGURE 1.3 : REPRESENTATION SCHEMATIQUE D'UNE MACHINE ASYNCHRONE	25
FIGURE 1.4: SCHEMA DE PRINCIPE DE LA METHODE DU MODELE DE REFERENCE	34
FIGURE 2.1 ILLUSTRATION DE LA METHODE DE RELAXATION CYCLIQUE	38
FIGURE 2.2 ORGANIGRAMME DE LA METHODE DE RELAXATION CYCLIQUE	39
FIGURE 2.3: ILLUSTRATION DE LA METHODE HOOKE ET JEEVES	40
FIGURE 2.4: ALGORITHME DE LA METHODE HOOKE ET JEEVES.....	40
FIGURE 2.5 ILLUSTRATION DE LA METHODE DE ROSENBRACK	41
FIGURE 2.6 ORGANIGRAMME DE LA METHODE DE ROSENBRACK	41
FIGURE 2.7 PRÉSENTATION DE PSO.....	46
FIGURE 2.8 ORGANIGRAMME DE L'ALGORITHME D'ESSAIS PARTICULAIRES	46
FIGURE 2.9 ORGANIGRAMME DE L'ALGORITHME DE LA COLONIE DE FOURMIS	47
FIGURE 2.10 ORGANIGRAMME DE L'ALGORITHME DE LUCIOLE.....	49
FIGURE 2.11 HYBRIDATION PARALLELE (A GAUCHE), SEQUENTIELLE (A DROITE).....	52
FIGURE 2.12 HYBRIDATION BAS NIVEAU DE COOPERATION.....	52
FIGURE 2.13 HYBRIDATION HAUT NIVEAU DE COOPERATION.....	53
FIGURE 2.14 CLASSIFICATION DES STRATEGIES D'HYBRIDATION	54
FIGURE 3.1 PRESENTATION DE L'ALGORITHME DE LUCIOLE AMELIORE	60
FIGURE 3.2 PRESENTATION DE L'OPTIMISEUR DE LOUPS GRIS (OLG)	63
FIGURE 3.3 ALGORITHME HYBRIDE PROPOSEE ALOLG	64
FIGURE 3.4 ALGORITHME HYBRIDE PROPOSEE ALASC1.....	66
FIGURE 3.5 ALGORITHME HYBRIDE PROPOSEE ALASC2.....	67
FIGURE 3.6 ALGORITHME HYBRIDE PROPOSEE ALOLGASC1	69
FIGURE 3.7 ALGORITHME HYBRIDE PROPOSEE ALOLGASC2	70
FIGURE 3.8 COURBES DE CONVERGENCE OBTENUES AVEC LES FONCTIONS F_{01} , F_{04} , F_{09} , F_{10} , F_{11} , F_{19}	76
FIGURE 3.9 COURBES DE CONVERGENCE SUR LES DIMENSIONS 150, 300 ET 500 OBTENUES AVEC F_{01} , F_{07} ET F_{11}	78
FIGURE 4.1 (A) TENSION D'ALIMENTATION (B) COURANT STATORIQUE CALCULE CORRESPONDANT	82
FIGURE 4.2 EVOLUTION DES PARAMETRES DE LA MACHINE CONNUE	85
FIGURE 4.3 MOTEUR M1 (A) TENSION MESURE (B) COURANT MESURE	85
FIGURE 4.4 MOTEUR M2 (A) TENSION MESURE (B) COURANT MESURE	86
FIGURE 4.5 SUPERPOSITION DU COURANT MESURE - - - - ET CALCULE ----- A L'AIDE DES PARAMETRES ESTIMES PAR AL ET ALV1 DU MOTEUR M1	88
FIGURE 4.6 ERREUR ENTRE LE COURANT MESURE ET CELUI CALCULE AVEC LES PARAMETRES ESTIMES PAR AL ET ALV1 DU MOTEUR M1.....	88
FIGURE 4.7 SUPERPOSITION DU COURANT MESURE - - - - ET CALCULE ----- A L'AIDE DES PARAMETRES ESTIMES PAR ALV2 ET ALV3 DU MOTEUR M1	89

FIGURE 4.8 ERREUR ENTRE LE COURANT MESURE ET CELUI CALCULE AVEC LES PARAMETRES ESTIMES PAR ALV2 ET ALV3 DU MOTEUR M1	89
FIGURE 4.9 SUPERPOSITION DU COURANT MESURE - - - - ET CALCULE ----- A L'AIDE DES PARAMETRES ESTIMES PAR ALA ET ALOLG DU MOTEUR M1	90
FIGURE 4.10 ERREUR ENTRE LE COURANT MESURE ET CELUI CALCULE AVEC LES PARAMETRES ESTIMES PAR ALA ET ALOLG DU MOTEUR M1	90
FIGURE 4.11 SUPERPOSITION DU COURANT MESURE - - - - ET CALCULE ----- A L'AIDE DES PARAMETRES ESTIMES PAR ALASC1 ET ALASC2 DU MOTEUR M1	91
FIGURE 4.12 ERREUR ENTRE LE COURANT MESURE ET CELUI CALCULE AVEC LES PARAMETRES ESTIMES PAR ALASC1 ET ALASC2 DU MOTEUR M1	91
FIGURE 4.13 SUPERPOSITION DU COURANT MESURE - - - - ET CALCULE ----- A L'AIDE DES PARAMETRES ESTIMES PAR ALOLGASC1 ET ALOLGASC2 DU MOTEUR M1.....	92
FIGURE 4.14 ERREUR ENTRE LE COURANT MESURE ET CELUI CALCULE AVEC LES PARAMETRES ESTIMES PAR ALOLGASC1 ET ALOLGASC2 DU MOTEUR M1	92
FIGURE 4.15 EVOLUTION DES PARAMETRES DE LA MACHINE M1 PAR ALA, ALASC1,2	94
FIGURE 4.16 EVOLUTION DES PARAMETRES DE LA MACHINE M1 PAR AL, ALA.	95
FIGURE 4.17 SUPERPOSITION DU COURANT MESURE - - - - ET CALCULE ----- A L'AIDE DES PARAMETRES ESTIMES PAR ALA ET ALOLG DU MOTEUR M1	96
FIGURE 4.18 ERREUR ENTRE LE COURANT MESURE ET CELUI CALCULE AVEC LES PARAMETRES ESTIMES PAR ALA ET ALOLG DU MOTEUR M1	96
FIGURE 4.19 SUPERPOSITION DU COURANT MESURE - - - - ET CALCULE ----- A L'AIDE DES PARAMETRES ESTIMES PAR ALASC1 ET ALASC2 DU MOTEUR M1	97
FIGURE 4.20 ERREUR ENTRE LE COURANT MESURE ET CELUI CALCULE AVEC LES PARAMETRES ESTIMES PAR ALASC1 ET ALASC2 DU MOTEUR M1	97
FIGURE 4.21 EVOLUTION DES PARAMETRES ESTIME AVEC LES ALGORITHMES ALA, ALOLG, ALASC1 ET ALASC2	99
FIGURE 4.22 SUPERPOSITION DU COURANT MESURE - - - - ET CALCULE ----- A L'AIDE DES PARAMETRES ESTIMES PAR AL ET ALV1 DU MOTEUR M2.....	101
FIGURE 4.23 ERREUR ENTRE LE COURANT MESURE ET CELUI CALCULE AVEC LES PARAMETRES ESTIMES PAR AL ET ALV1 DU MOTEUR M2.....	101
FIGURE 4.24 SUPERPOSITION DU COURANT MESURE - - - - ET CALCULE ----- A L'AIDE DES PARAMETRES ESTIMES PAR ALV2 ET ALV3 DU MOTEUR M2	102
FIGURE 4.25 ERREUR ENTRE LE COURANT MESURE ET CELUI CALCULE AVEC LES PARAMETRES ESTIMES PAR ALV2 ET ALV3 DU MOTEUR M2	102
FIGURE 4.26 SUPERPOSITION DU COURANT MESURE - - - - ET CALCULE ----- A L'AIDE DES PARAMETRES ESTIMES PAR ALA ET ALOLG DU MOTEUR M2	103
FIGURE 4.27 ERREUR ENTRE LE COURANT MESURE ET CELUI CALCULE AVEC LES PARAMETRES ESTIMES PAR ALA ET ALOLG DU MOTEUR M2	103
FIGURE 4.28 SUPERPOSITION DU COURANT MESURE - - - - ET CALCULE ----- A L'AIDE DES PARAMETRES ESTIMES PAR ALASC1 ET ALASC2 DU MOTEUR M2	104
FIGURE 4.29 ERREUR ENTRE LE COURANT MESURE ET CELUI CALCULE AVEC LES PARAMETRES ESTIMES PAR ALASC1 ET ALASC2 DU MOTEUR M2	104
FIGURE 4.30 SUPERPOSITION DU COURANT MESURE - - - - ET CALCULE ----- A L'AIDE DES PARAMETRES ESTIMES PAR ALOLGASC1 ET ALOLGASC2 DU MOTEUR M2.....	105
FIGURE 4.31 ERREUR ENTRE LE COURANT MESURE ET CELUI CALCULE AVEC LES PARAMETRES ESTIMES PAR ALOLGASC1 ET ALOLGASC2 DU MOTEUR M2	105

TABLEAU 3.1 PARAMETRAGES DES ALGORITHMES	56
TABLEAU 3.2 CARACTERISTIQUES DES FONCTIONS-TESTS UTILISEES POUR LA COMPARAISON	57
TABLEAU 3.3 RESULTATS D'OPTIMISATION	61
TABLEAU 3.4 RESULTATS D'OPTIMISATION	65
TABLEAU 3.5 RESULTATS D'OPTIMISATION	68
TABLEAU 3.6 RESULTATS D'OPTIMISATION	71
TABLEAU 3.7 RESULTATS POUR LES FONCTIONS UNI-MODALES (F01 A F07).....	72
TABLEAU 3.8 RESULTATS POUR LES FONCTIONS UNI-MODALES (F01 A F07).....	72
TABLEAU 3.9 RESULTATS POUR LES FONCTIONS MULTIMODALES (F08 A F23)	72
TABLEAU 3.10 RESULTATS POUR LES FONCTIONS MULTIMODALES (F08 A F23).....	73
TABLEAU 3.11 RESULTATS POUR UNE DIMENSION = 150	74
TABLEAU 3.12 RESULTATS POUR UNE DIMENSION D= 300	75
TABLEAU 3.13 RESULTATS POUR UNE DIMENSION D= 500	75
TABLEAU 4.1 IDENTIFICATION L'AIDE DES DONNEES SIMULEES.....	83
TABLEAU 4.2 IDENTIFICATION L'AIDE DES DONNEES SIMULEES.....	83
TABLEAU 4.3 IDENTIFICATION DES PARAMETRES DE MOTEUR M1	86
TABLEAU 4.4 IDENTIFICATION DES PARAMETRES DE MOTEUR M1	86
TABLEAU 4.5 PARAMETRES DU MOTEUR M1 EN CONSIDERANT LA DEUXIEME CAGE	95
TABLEAU 4.6 IDENTIFICATION DES PARAMETRES DE MOTEUR M2	99
TABLEAU 4.7 IDENTIFICATION DES PARAMETRES DE MOTEUR M2	100
TABLEAU 4.8 PARAMETRES DU MOTEUR M2 EN CONSIDERANT LA DEUXIEME CAGE	100
TABLEAU 4.9 LE POURCENTAGE D'ERREUR DES DEUX MODELES POUR LE MOTEUR M1.....	107
TABLEAU 4.10 COMPARAISON ENTRE LE MODELE A CAGE ET A DEUX CAGES.....	107

INTRODUCTION GENERALE

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 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 les développements technologiques, tant dans le domaine de l'électronique de puissance que celui de la microélectronique (DSP, FPGA), ont rendu possible l'usage de commandes performantes faisant de la machine asynchrone un concurrent potentiel dans les domaines de la vitesse variable. La conception d'une commande performante requière une bonne modélisation et une bonne identification des systèmes.

Identifier un système consiste à proposer une structure entre son entrée et sa sortie, ainsi de déterminer à partir du couple (entrée/sortie) les valeurs des paramètres de son modèle. Ainsi, différentes méthodes d'identification peuvent être utilisées. Les différentes méthodes développées peuvent être classées en trois catégories : les méthodes graphiques, les méthodes non récursives (méthodes des moindres carrés) et méthodes récursives (méthodes des moindres carrés récursifs). D'une manière générale, le principe des méthodes d'identification consiste à minimiser l'écart observé entre la sortie prédite par le modèle et celle mesurée en définissant un critère. Ce principe est connu par la méthode du modèle, qui est utilisée en pratique car le problème d'identification est formulé sous forme d'un problème d'optimisation dont les paramètres à identifier sont les variables de décision. La méthode du modèle reste l'une des méthodes d'identification les plus efficaces.

Plusieurs méthodes d'optimisation sont proposées dans la littérature [1]–[3], et peuvent être classées comme suit :

- Les méthodes déterministes.
- Les méthodes stochastiques.

Les méthodes déterministes utilisent les informations topologiques (continuité, dérivabilité) sur les valeurs aux points courants de la fonction à optimiser. Généralement, elles convergent vers l'optimum local le plus proche du point initial. Par contre, les méthodes stochastiques ne nécessitent aucune connaissance a priori sur la fonction à optimiser.

Les méthodes stochastiques qui ont eu le plus grand succès sont basées sur des métaheuristiques, qui sont des méthodes algorithmiques capables de guider, d'orienter et d'explorer le processus de recherche dans un espace de solution, souvent très grand à des régions riches en solutions optimales et de converger vers une solution satisfaisante en un temps raisonnable. Le fait de rendre cette méthode abstraite et plus générique conduit à une vaste utilisation pour des champs d'applications différentes. Cependant, le choix des paramètres d'une métaheuristique est très critique; il affecte, d'une manière considérable, ses performances et son comportement.

L'hybridation des métaheuristiques [4],[5] est la voie la plus prometteuse pour l'amélioration de la qualité des solutions. Elle a permis de bénéficier des points forts de chacune de ces méthodes. Elle consiste à exploiter les avantages respectifs de deux ou plusieurs méthodes en combinant leurs algorithmes suivant une approche synergétique. Elle a permis d'avoir un compromis entre l'exploration et l'exploitation de l'espace de recherche des solutions.

L'algorithme de luciole est l'un des algorithmes métaheuristiques d'intelligence en essaim récent, introduit en 2008 par Xin She Yang [6], il est basé sur le comportement des lucioles tropicales. L'algorithme standard est simple, flexible et facile à comprendre et à mettre en œuvre. Il a été appliqué à l'optimisation continue [7], multimodal [6], à l'optimisation sous contrainte [7], [8]. Cet algorithme est largement utilisé pour résoudre de nombreux problèmes d'ingénieries [9]–[12], mais elle a tendance d'être coincée dans un optimum local, ce qui guide l'essaim vers la convergence prématurée et cesse la convergence vers l'optimum global. Aussi, les mouvements des lucioles dépendent uniquement de la performance actuelle sans mémoriser les meilleures solutions et performances précédentes. Cela peut conduire à la perte de meilleures solutions qui a la plus grande attractivité, et de dégrader la performance de l'algorithme. C'est pourquoi la modification de l'algorithme standard de la luciole pour améliorer ses performances a été l'un des sujets de recherche les plus intéressants . Parmi

les améliorations, les chercheurs proposent de changer la formule de mise à jour [13]–[15] ou de conduire le déplacement vers des directions admissibles afin d'améliorer le taux de convergence[16].

Dans cette thèse, le travail présenté est consacré à l'amélioration de l'algorithme de luciole standard et au développement des algorithmes de luciole hybride avec d'autres métaheuristiques. Pour cela, Nous proposons deux propositions :

- La première proposition consiste à orienter le mouvement des lucioles vers le centroïde global de l'essaim d'une manière prédéfinie afin d'améliorer les performances de l'algorithme standard s'il n'y a pas de luciole plus brillante que celle-ci, ce qui augmente l'exploration si les lucioles sont loin entre eux, et renforce l'exploitation si les lucioles sont proches entre eux. C'est-à-dire augmenter l'exploration au début de recherche et renforcer l'exploitation à la fin de la recherche.

- La deuxième proposition consiste à hybrider l'algorithme de luciole avec l'optimiseur de loup gris ou / et avec l'algorithme de sinus cosinus.

Comme d'autres métaheuristiques [17]–[19], l'algorithme de luciole est appliqué avec succès à l'identification paramétrique de la machine asynchrone. Plusieurs approches d'identification [1], [20] 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 la technique de minimisation adoptée.

Nous adoptons une approche d'identification utilisant uniquement le courant de démarrage et la tension simple correspondante permettant d'estimer simultanément les paramètres électriques et mécaniques de la machine asynchrone. Ainsi en utilisant la méthode de modèle de référence, les paramètres sont déterminés par minimisation de l'erreur quadratique entre les courants mesurés et calculés à partir du modèle adopté de la machine. La minimisation de l'erreur quadratique est assurée par les algorithmes développés.

La vérification de l'approche d'identification est assurée par les données simulées. À l'aide de données expérimentales, la validation des résultats est confirmée par la bonne superposition des courants mesurés et ceux calculés par le modèle adopté de la machine. Une comparaison des différents algorithmes de luciole est établie en terme de vitesse de convergence et temps de calcul.

Pour atteindre notre objectif, nous avons le plan suivant :

dans le premier chapitre, nous commençons par un bref rappel sur les techniques d'identification des systèmes, puis nous rappelons le modèle de la machine en considérant l'existence d'une seule cage puis en considérant la deuxième cage pour affiner le modèle de plus. Nous présentons également l'identification paramétrique de la machine asynchrone à l'aide la méthode du modèle de référence.

Le deuxième chapitre consiste à présenter les différentes techniques d'optimisation qui peuvent être réparties en deux catégories :

- Les méthodes déterministes sans et avec le calcul de dérivées, sont présentées.

- Les méthodes stochastiques qui reposent sur les règles de transition probabiliste, et qui peuvent être réparties en algorithmes à solution unique et algorithmes à population de solutions.

À la fin de ce chapitre, nous discutons quelque truque de perfectionnement des métaheuristiques par l'hybridation avec d'autres métaheuristiques. Nous citons aussi une classification de l'hybridation entre les métaheuristiques.

Dans le troisième chapitre, les algorithmes de lucioles sont le point essentiel de ce travail. Nous présentons les différentes étapes d'implémentation des algorithmes de luciole standard et certaines variantes existants dans la littérature. Pour améliorer les performances des algorithmes de luciole standard, nous proposons en premier temps une orientation de la phase aléatoire vers le centroïde de l'essaim, puis nous hybridons l'algorithme standard une fois par l'optimiseur de loups gris et autres fois par l'algorithme d'optimisation sinus cosinus. Une étude expérimentale et des comparaisons avec l'algorithme de luciole standard, via différentes fonctions de référence classiques, sont également présentées.

Dans le quatrième chapitre, nous vérifions nos algorithmes de luciole développés en identification des paramètres électriques et mécaniques à l'aide de données simulées. La validation est réalisée à l'aide de données mesurées effectuées sur deux moteurs différents.

La conclusion générale clôture cette thèse et expose quelques perspectives qui ouvrent des voies pour des développements et des recherches ultérieures.

CHAPITRE 1

MODELISATION ET TECHNIQUE D'IDENTIFICATION DE LA MACHINE ASYNCHRONE

1.1. Introduction

L'identification paramétrique est basée sur l'extraction du modèle mathématique à partir des observations. Ce modèle doit permettre de calculer la sortie du procédé à n'importe quel instant de temps si les conditions initiales du système sont connues. Pour cela, Il est tout de même important d'avoir des connaissances basiques du système pour choisir un type de modèle adopté. On rappelle dans ce premier chapitre, le problème d'identification et ses différentes méthodes, puis on discute la modélisation de la machine asynchrone à cage et à double cage dédiées à l'identification paramétrique. Puis, Nous avons cité aussi certaines techniques d'identification de la machine asynchrone.

1.2. Principe de l'identification

L'identification est l'opération de détermination des caractéristiques dynamiques d'un procédé (système) dont la connaissance est nécessaire pour la conception et la mise en œuvre d'un système performant de régulation. La notion de modèle mathématique, c'est-à-dire d'un ensemble d'équations liant ses entrées et ses sorties, est un concept fondamental.

En général, il existe trois classes de modèles, chaque type de modèle étant destiné à une application particulière. La première consiste à utiliser les principes phénoménologiques (lois de la physique, de la biologie, etc.) gouvernant le système. On parle alors de modèles de connaissance ou encore de modèle "boîte blanche". Une alternative à cette approche est fondée sur une procédure expérimentale consistant à mesurer, à analyser et à corrélérer les entrées et les sorties du système. Il s'agit alors de modèles comportementaux (ou modèles "boîte noire"). La troisième est une combinaison des deux précédentes : quelquefois appelée boîte grise, elle utilise, dans le modèle boîte noire, les connaissances phénoménologiques disponibles afin de mieux adapter les paramètres du modèle mathématique aux données expérimentales.

Dans les deux premiers cas, certains paramètres peuvent être inconnus (ou mal connus). La détermination d'un modèle à partir des données entrée-sortie, c'est-à-dire l'identification du modèle, constitue un problème inverse, par opposition au problème direct qui consiste à déterminer la (ou les) sortie(s), connaissant la (ou les) entrée(s) et les paramètres du modèle (ce que l'on fait en simulation).

Dans la pratique, l'objectif général de l'identification est la détermination de modèles de conduite afin de simuler, d'analyser ou de commander un système. Pour cela, la détermination des modèles de connaissance est une tâche qui intéresse plus les physiciens (ou les biologistes, etc.) que les automaticiens. Ainsi, nous sommes amenés à mettre en œuvre une méthodologie d'identification directe de ces modèles dynamiques (de commande) qui sont sous deux types :

- Les modèles non paramétriques (réponse fréquentielle, réponse à un échelon, etc.),
- Les modèles paramétriques (fonction de transfert, équation différentielle ou aux différences, etc.).

Expérimentalement, l'identification comporte quatre étapes :

1. acquisition des entrées/sorties sous protocole d'expérimentation ;
2. choix d'une structure de modèle ;
3. estimation des paramètres du modèle ;
4. Validation du modèle identifié.

Le travail présenté dans cette thèse porte essentiellement sur la phase d'estimation des paramètres d'un modèle préalablement établi (Modèle de la machine asynchrone).

1.3. Méthodes d'identification

Les différentes méthodes d'identification peuvent être scindées en quatre classes :

1.3.1. Méthodes de base (Méthodes graphiques)

Ces méthodes sont basées sur la réponse indicelle du système. Elles sont graphiques et ne permettent pas d'avoir des résultats précis mais peuvent conduire à un modèle suffisant. La réponse d'un système à un échelon ou une

impulsion sur la variable d'entrée (commande) fournit un renseignement intéressant sur sa dynamique.

Les méthodes graphiques sont mal adaptées à l'identification du modèle précis ou d'ordre élevé (≥ 2). Parmi les méthodes graphiques on peut citer la méthode de Streje, méthode de Broïda.

1.3.2. Méthode non récursive

Ces méthodes traitent l'ensemble des données (entrées/sorties) obtenues sur un horizon "période" de temps. On distingue le principe des moindres carrés selon Gauss, qui est de choisir les paramètres d'un modèle de telle sorte que la somme des carrés de mesures expérimentalement multipliées par un coefficient caractérisant le degré de précision, soit minimale. Cette approche peut s'appliquer à n'importe quel type de modèle, mais une solution analytique ne peut être obtenue que si le modèle est linéaire en fonction des paramètres. La formule standard des moindres carrés consiste à supposer que les valeurs calculées (prédites) $\hat{y}(t)$ sont données par le modèle.

1.3.3. Méthode récursive

Ces méthodes traitent les pairs de données (entrées-sorties) à chaque instant et produisant une estimation paramétrique, données ou de la lecture des données.

L'identification récursive permet de calculer en ligne, à chaque période d'échantillonnage, un nouveau estimé du vecteur de paramètres du système $\hat{\theta}(k)$.

L'algorithme sera de la forme suivante :

$$\hat{\theta}(k) = \hat{\theta}(k-1) + C(k) \quad (1.1)$$

Le nouveau vecteur estimé est par conséquent l'estimé calculée à la période précédente auquel on ajoute une correction basée sur la nouvelle information disponible $y(k)$ et $u(k)$.

L'identification récursive permet de suivre les variations des paramètres du procédé particulièrement si on donne un poids fort, dans le calcul, aux mesures les plus récentes.

1.3.4. Méthode du modèle

La méthode du modèle suit strictement, dans son déroulement, le principe de base de la modélisation.

On cherche, par une méthode itérative, à minimiser l'écart entre la variable mesurée et celle estimée par le modèle. Sachant que cet écart est une fonction positive des paramètres du modèle. Le problème se ramène donc à un problème de minimisation d'une fonction, généralement réalisée par des techniques d'optimisation.

On constate que les méthodes qui procèdent directement à une minimisation d'un écart entre les mesures et les valeurs prédites par le modèle présentent des performances plus intéressantes, bien qu'elles soient plus lourdes de mise en œuvre.

1.4. Modélisation de la Machine Asynchrone, Modèle à cage

Plusieurs travaux de recherche sont proposés pour la modélisation de la machine asynchrone triphasée[21]–[23], Nous distinguons :

- Les méthodes basées sur le circuit équivalent de la machine.
- Les méthodes basées sur le modèle dynamique de la machine, qui permettent de présenter le modèle de la machine sous forme des systèmes d'équation différentielle, ce qui réduit la complexité du modèle et permet d'étudier les modes permanents et transitoires.

La machine asynchrone triphasée étant un système dynamique non linéaire. Il est nécessaire de disposer d'un modèle représentant fidèlement son comportement au niveau de ses modes électrique, électromagnétique et mécanique. Pour cela, nous adoptons les hypothèses suivantes :

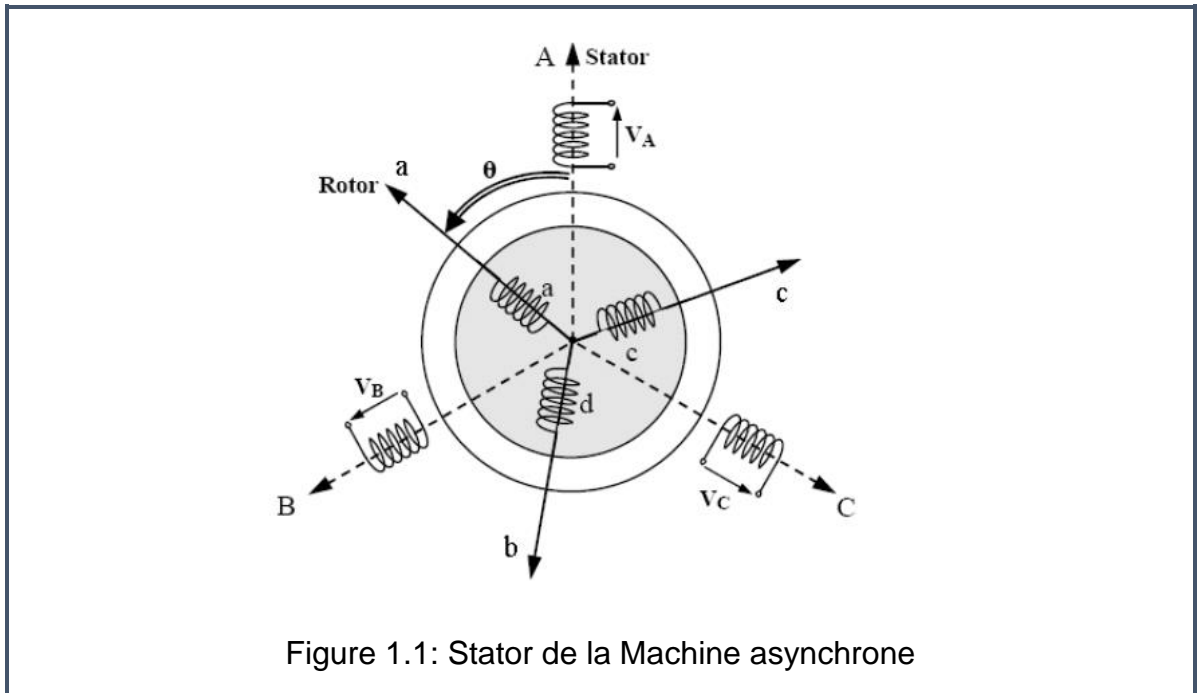
- Les enroulements statoriques sont identiques par rapport à l'axe de symétrie.
- L'inductance et l'effet de peau et de l'échauffement sur les caractéristiques ne sont pas pris en compte.
- L'induction électromagnétique est à répartition spatiale sinusoïdale.
- Le circuit magnétique est non saturé et à perméabilité constante.
- L'entre fer est supposé constant.

Compte tenu des hypothèses précédentes, on peut citer les conséquences suivantes :

- L'additivité des flux.
- La constance des inductances propres.

- La loi de variation sinusoïdale des inductances mutuelles entre les enroulements statoriques et rotoriques en fonction de l'angle électrique de leurs axes magnétiques.

La machine asynchrone triphasée est représentée schématiquement par la figure 1.1. Le stator et le rotor sont représentés respectivement par trois enroulements identiques et déphasés de 120° les uns par rapport aux autres. L'angle θ caractérise la position angulaire du rotor par rapport au stator.



1.4.1. Equations électriques

En appliquant la loi de Faraday à chaque enroulement de la machine représentée par la **figure 1.1**, on peut écrire :

$$[V_s] = [R_s][I_s] + \frac{d[\phi_s]}{dt} \quad (1.1)$$

$$[V_r] = [R_r][I_r] + \frac{d[\phi_r]}{dt} \quad (1.2)$$

Où l'indice s et l'indice r désignent respectivement les grandeurs relatives au stator et au rotor.

Avec

$$[V_s] = \begin{bmatrix} V_{sa} \\ V_{sb} \\ V_{sc} \end{bmatrix}, [I_s] = \begin{bmatrix} I_{sa} \\ I_{sb} \\ I_{sc} \end{bmatrix}, [\phi_s] = \begin{bmatrix} \phi_{sa} \\ \phi_{sb} \\ \phi_{sc} \end{bmatrix} \quad (1.3)$$

$$[V_r] = \begin{bmatrix} V_{ra} \\ V_{rb} \\ V_{rc} \end{bmatrix}, [I_r] = \begin{bmatrix} I_{ra} \\ I_{rb} \\ I_{rc} \end{bmatrix}, [\phi_r] = \begin{bmatrix} \phi_{ra} \\ \phi_{rb} \\ \phi_{rc} \end{bmatrix} \quad (1.4)$$

Les matrices des résistances statoriques et rotoriques de la machine asynchrone sont données par :

$$[R_s] = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix}, [R_r] = \begin{bmatrix} R_r & 0 & 0 \\ 0 & R_r & 0 \\ 0 & 0 & R_r \end{bmatrix} \quad (1.5)$$

1.4.2. Equations magnétiques

Sous les hypothèses précédentes, les relations linéaires entre le flux et les courants sont exprimées sous forme matricielles comme suit

$$\begin{bmatrix} [\phi_s] \\ [\phi_r] \end{bmatrix} = \begin{bmatrix} [L_s] & [M_{sr}] \\ [M_{rs}] & [L_r] \end{bmatrix} \begin{bmatrix} [I_s] \\ [I_r] \end{bmatrix} \quad (1.6)$$

Avec

$$[L_s] = \begin{bmatrix} L_s & m_s & m_s \\ m_s & L_s & m_s \\ m_s & m_s & L_s \end{bmatrix}, [L_r] = \begin{bmatrix} L_r & m_r & m_r \\ m_r & L_r & m_r \\ m_r & m_r & L_r \end{bmatrix} \quad (1.7)$$

$$[M_{rs}] = [M_{sr}]^t = m_{rs} \begin{bmatrix} \cos(\theta) & \cos(\theta + \frac{2\pi}{3}) & \cos(\theta - \frac{2\pi}{3}) \\ \cos(\theta - \frac{2\pi}{3}) & \cos(\theta) & \cos(\theta + \frac{2\pi}{3}) \\ \cos(\theta + \frac{2\pi}{3}) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta) \end{bmatrix} \quad (1.8)$$

- L_s et L_r : Inductances propres statorique et rotorique.
- m_s et m_r : Inductances mutuelles entre deux phases statoriques ou rotoriques.
- m_{sr} : Inductance mutuelle maximale entre une phase statorique et rotorique.
- θ : écart angulaire entre stator et rotor.

On obtient finalement le modèle asynchrone triphasé :

$$\begin{cases} [V_{sabc}] = [R_s][I_{sabc}] + \left(\frac{d}{dt}\right)\{[L_s][I_{sabc}] + [M_{sr}][I_{rabc}]\} \\ [V_{rabc}] = [R_r][I_{rabc}] + \left(\frac{d}{dt}\right)\{[M_{rs}][I_{sabc}] + [L_r][I_{rabc}]\} \end{cases} \quad (1.9)$$

1.4.3. Equation mécanique

La relation fondamentale de la dynamique permet d'écrire :

$$C_{em} - C_r = J \frac{d\Omega}{dt} + f\Omega \quad (1.10)$$

Où

C_{em} : représente le couple électromagnétique.

C_r : représente le couple résistant.

J : Le moment d'inertie de l'ensemble des parties tournantes.

f : Le coefficient de frottement.

La machine étant à structure lisse, L'expression du couple électromagnétique [34] [39] [41] s'écrit :

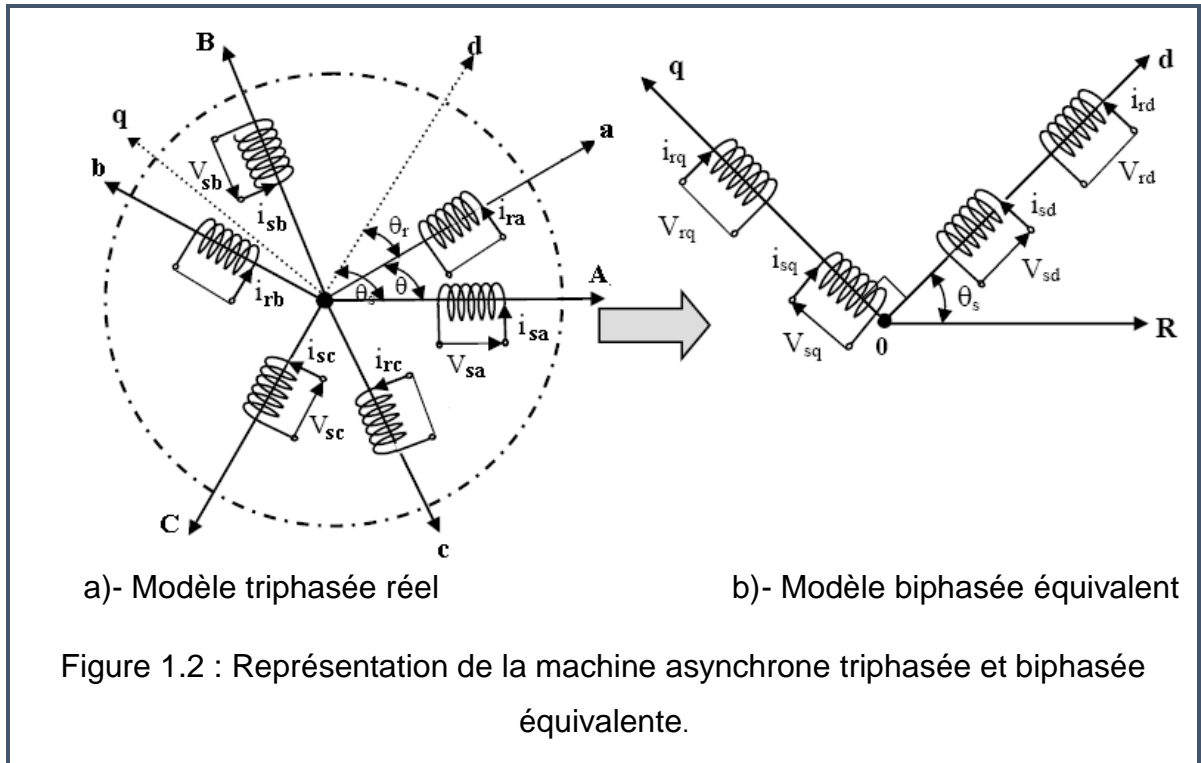
$$C_{em} = [I_s]^T \frac{\partial [M_{sr}(\theta)]}{\partial \theta} [I_r] \quad (1.11)$$

1.4.4. Modèle de la machine dans le modèle de Park

La transformation de Park [4] [37-38] [40], consiste à remplacer les trois phases statoriques et rotoriques par un système diphasé d'axe d et q. 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'équation (1.12).

La figure 1.2 représente le schéma d'une machine asynchrone triphasée et sa machine biphasée équivalente issue de la transformation de Park.

$$P(\theta) = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta - \frac{4\pi}{3}) \\ -\sin(\theta) & -\sin(\theta - \frac{2\pi}{3}) & -\sin(\theta - \frac{4\pi}{3}) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \quad (1.12)$$



Avec :

θ_s : L'angle entre l'axe direct et l'axe statorique.

θ_r : L'angle entre l'axe direct et l'axe rotorique

θ : L'angle entre l'axe de référence statorique et rotorique.

Dans un repère dq tournant à la vitesse ω_a par rapport au stator, le système d'équations (1.9) s'écrit après application de la transformation de Park aux différences grandeurs comme suit :

$$\left\{ \begin{array}{l} v_{ds} = R_s I_{ds} + \frac{d}{dt} \varphi_{ds} - \omega_a \varphi_{qs} \\ v_{qs} = R_s I_{qs} + \frac{d}{dt} \varphi_{qs} + \omega_a \varphi_{ds} \\ v_{dr} = R_r I_{dr} + \frac{d}{dt} \varphi_{dr} - (\omega_a - \omega_m) \varphi_{qr} \\ v_{qr} = R_r I_{qr} + \frac{d}{dt} \varphi_{qr} + (\omega_a - \omega_m) \varphi_{dr} \end{array} \right. \quad (1.13)$$

ω_a et ω_m désignent respectivement la vitesse du repère dq et la vitesse du rotor par rapport au stator. Pour un repère lié au stator $\omega_a = 0$.

Les expressions des flux s'écrivent dans le modèle de Park :

$$\begin{cases} \varphi_{ds} = L_s I_{ds} + L_m I_{dr} \\ \varphi_{qs} = L_s I_{qs} + L_m I_{qr} \\ \varphi_{dr} = L_r I_{dr} + L_m I_{ds} \\ \varphi_{qr} = L_r I_{qr} + L_m I_{qs} \end{cases} \quad (1.14)$$

Les différentes inductances désignant :

- 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 rotor.

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

$$C_{em} = PL_m (I_{qs} I_{dr} - I_{ds} I_{qr}) \quad (1.15)$$

P désigne le nombre de pair de pôles de la machine.

1.4.5. Modèle à cinq paramètres électriques

Pour un référentiel lié au stator ($\omega_a = 0$), la machine sera totalement décrite par le système suivant :

$$\begin{cases} V_{ds} = R_s I_{ds} + L_s \frac{dI_{ds}}{dt} + L_m \frac{dI_{dr}}{dt} \\ V_{qs} = R_s I_{qs} + L_s \frac{dI_{qs}}{dt} + L_m \frac{dI_{qr}}{dt} \\ V_{dr} = R_r I_{dr} + L_r \frac{dI_{dr}}{dt} + L_m \frac{dI_{ds}}{dt} + p\Omega L_r I_{qr} + p\Omega L_m I_{qs} \\ V_{qr} = R_r I_{qr} + L_r \frac{dI_{qr}}{dt} + L_m \frac{dI_{qs}}{dt} + p\Omega L_r I_{dr} - p\Omega L_m I_{ds} \\ J \frac{d\Omega}{dt} = pL_m (I_{qs} I_{dr} - I_{ds} I_{qr}) - f_r \Omega \end{cases} \quad (1.16)$$

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

$$\begin{cases} \frac{d}{dt} I_{ds} = -\frac{L_r R_s}{A} I_{ds} + \frac{L_m^2}{A} p\Omega I_{qs} + \frac{L_m R_r}{A} I_{dr} + \frac{L_m L_r}{A} p\Omega I_{qr} + \frac{L_r V_{ds}}{A} \\ \frac{d}{dt} I_{qs} = \frac{L_m^2}{A} p\Omega I_{ds} - \frac{L_r R_s}{A} I_{qs} + \frac{L_m L_r}{A} p\Omega I_{dr} + \frac{L_m R_r}{A} I_{qr} + \frac{L_r V_{qs}}{A} \\ \frac{d}{dt} I_{dr} = \frac{L_m R_s}{A} I_{ds} - \frac{L_m L_s}{A} p\Omega I_{qs} - \frac{L_s R_r}{A} I_{dr} - \frac{L_s L_r}{A} p\Omega I_{qr} - \frac{L_m V_{ds}}{A} \\ \frac{d}{dt} I_{qr} = \frac{L_m L_s}{A} p\Omega I_{ds} + \frac{L_m R_s}{A} I_{qs} + \frac{L_s L_r}{A} p\Omega I_{dr} - \frac{L_s R_r}{A} I_{qr} - \frac{L_m V_{qs}}{A} \\ \frac{d}{dt} \Omega = \frac{pL_m}{J} (I_{qs} I_{dr} - I_{ds} I_{qr}) - \frac{B}{J} \Omega \end{cases} \quad (1.17)$$

Ou $A = L_s \cdot L_r - L_m^2$

L'équation **(1.17)**, montre que le fonctionnement de la machine dépend de cinq paramètres électriques $[R_s, R, L_r, L_s, L_m]$ et de deux paramètres mécaniques $[J, f_r]$. Pour réduire le nombre de paramètres électrique de modèle de cinq à quatre nous faisons un changement de variable adéquate. Pour cela P.L. ALGER [22] propose l'égalité des inductances de fuite au rotor et au stator.

1.4.6. Modèle à quatre paramètres électriques

Pratiquement en mode normale de fonctionnent les tensions d'alimentation appliquer au rotor V_{dr} et V_{qr} , ce qui a pour conséquence de réduire le nombre de paramètres à quatre lorsqu'on effectue le changement de variables suivant :

$$I_{dr} = \frac{L_m}{L_r} I'_{dr}, I_{qr} = \frac{L_m}{L_r} I'_{qr} \quad (1.18)$$

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 **(1.17)** deviennent :

$$\left\{ \begin{array}{l} \frac{d}{dt} I_{ds} = -\frac{1}{\sigma T_s} I_{ds} + \frac{1-\sigma}{\sigma} p\Omega I_{qs} + \frac{1-\sigma}{\sigma T_r} I'_{dr} + \frac{1-\sigma}{\sigma} p\Omega I'_{qr} + \frac{V_{ds}}{\sigma L_s} \\ \frac{d}{dt} I_{qs} = -\frac{1-\sigma}{\sigma} p\Omega I_{ds} + \frac{1}{\sigma T_s} I_{qs} - \frac{1-\sigma}{\sigma} p\Omega I'_{dr} + \frac{1-\sigma}{\sigma T_r} I'_{qr} + \frac{V_{qs}}{\sigma L_s} \\ \frac{d}{dt} I'_{dr} = -\frac{1}{\sigma T_s} I_{ds} - \frac{1}{\sigma} p\Omega I_{qs} - \frac{1}{\sigma T_r} I'_{dr} - \frac{1}{\sigma} p\Omega I'_{qr} - \frac{V_{ds}}{\sigma L_s} \\ \frac{d}{dt} I'_{qr} = \frac{1}{\sigma} p\Omega I_{ds} + \frac{1}{\sigma T_s} I_{qs} + \frac{1}{\sigma} p\Omega I'_{dr} - \frac{1}{\sigma T_r} I'_{qr} - \frac{V_{qs}}{\sigma L_s} \\ \frac{d}{dt} \Omega = \frac{p}{J} (1-\sigma) L_s (I'_{qr} I_{ds} - I_{qr} I'_{dr}) - \frac{B}{J} \Omega \end{array} \right. \quad (1.19)$$

L'équation **(1.19)** montre que le fonctionnement de la machine dépend de quatre paramètres électriques $[\sigma, T_r, L_s, T_s]$ et de deux paramètres mécaniques $[J, f_r]$.

1.5. Modélisation de la Machine Asynchrone, Modèle à double cage

Au démarrage, le moteur asynchrone développe un couple relativement faible par rapport au fort courant appelé, ce qui peut provoquer un échauffement nuisible de la machine. Des solutions ont été apportées pour améliorer le démarrage de la MAS, et parmi elles, on peut citer l'utilisation des moteurs à

encoches profondes ou à double cage qui sont basées sur le principe de refoulement de courant.

Dans le machine a double cage le rotor est construit suivant le principe du rotor à cage simple, mais avec deux cages électriquement indépendantes. Une cage externe à la périphérie du rotor est composée de matériaux résistifs (laiton, bronze) et possède une faible dispersion magnétique. Une cage interne en cuivre possède une résistivité plus faible et une dispersion magnétique importante. La cage externe, surtout active au démarrage, permet d'obtenir un couple plus important dans cette phase de fonctionnement, tandis qu'à régime nominal la cage interne permet de retrouver les caractéristiques d'un rotor à simple cage.

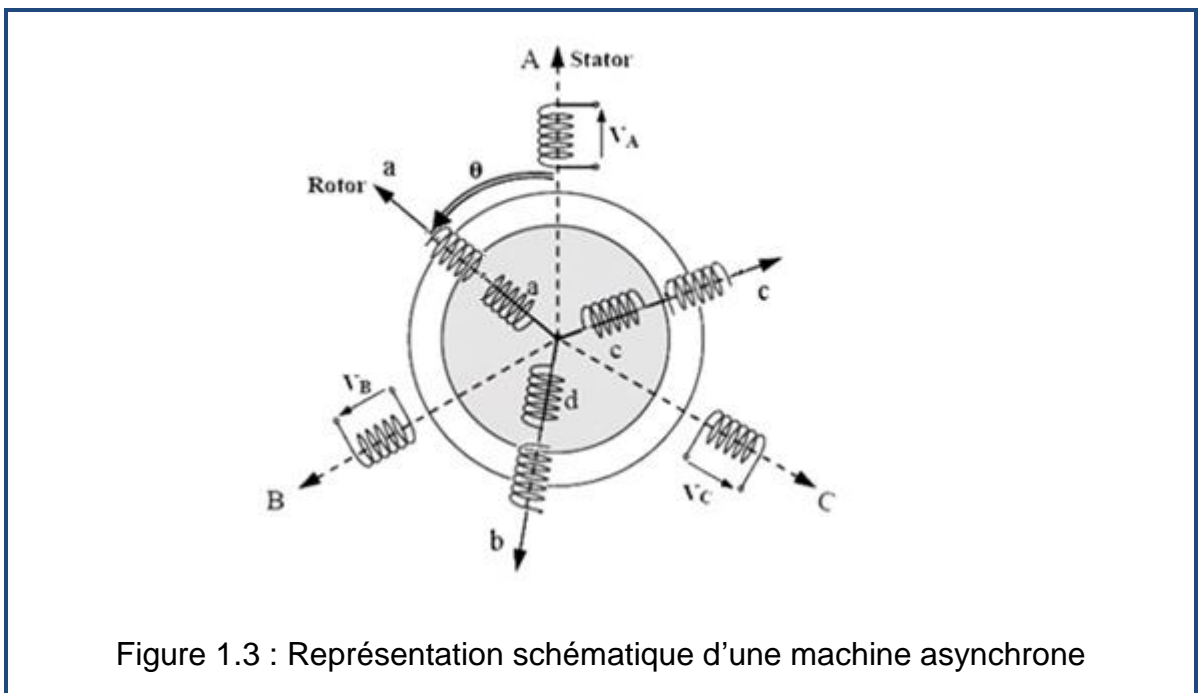


Figure 1.3 : Représentation schématique d'une machine asynchrone

1.5.1.Équations électriques

Les équations de tension des trois phases statorique et des six phases de rotorique (rotor interne et rotor externe) sont données par les expressions suivantes:

$$\begin{cases} V_{sabc} = R_s I_s + \frac{d}{dt} \phi_{sabc} \\ V_{r1abc} = R_{r1} I_{r1} + \frac{d}{dt} \phi_{r1abc} \\ V_{r2abc} = R_{r2} I_{r2} + \frac{d}{dt} \phi_{r2abc} \end{cases} \quad (1.20)$$

Où les indices s, r₁ et r₂ désignent respectivement les grandeurs relatives au stator et aux deux rotors.

$$[V_{r1}] = \begin{bmatrix} V_{r1a} \\ V_{r1b} \\ V_{r1c} \end{bmatrix}, [I_{r1}] = \begin{bmatrix} I_{r1a} \\ I_{r1b} \\ I_{r1c} \end{bmatrix}, [\phi_{r1}] = \begin{bmatrix} \phi_{r1a} \\ \phi_{r1b} \\ \phi_{r1c} \end{bmatrix} \quad (1.21)$$

$$[V_{r2}] = \begin{bmatrix} V_{r2a} \\ V_{r2b} \\ V_{r2c} \end{bmatrix}, [I_{r2}] = \begin{bmatrix} I_{r2a} \\ I_{r2b} \\ I_{r2c} \end{bmatrix}, [\phi_{r2}] = \begin{bmatrix} \phi_{r2a} \\ \phi_{r2b} \\ \phi_{r2c} \end{bmatrix} \quad (1.22)$$

1.5.2. Équations des flux

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

$$\begin{cases} \phi_{sabc} = L_s I_s + M_{sr2} I_{r2} + M_{sr1} I_{r1} \\ \phi_{r1abc} = M_{sr1} I_s + L_{r1} I_{r1} + M_{r1r2} I_{r2} \\ \phi_{r2abc} = M_{sr2} I_s + M_{r2r1} I_{r1} + L_{r2} I_{r2} \end{cases} \quad (1.23)$$

Les neuf sous matrices d'inductances s'écrivent :

$$L_s = \begin{pmatrix} l_s & m_s & m_s \\ m_s & l_s & m_s \\ m_s & m_s & l_s \end{pmatrix}, L_{r1} = \begin{pmatrix} l_{r1} & m_{r1} & m_{r1} \\ m_{r1} & l_{r1} & m_{r1} \\ m_{r1} & m_{r1} & l_{r1} \end{pmatrix}, L_{r2} = \begin{pmatrix} l_{r2} & m_{r2} & m_{r2} \\ m_{r2} & l_{r2} & m_{r2} \\ m_{r2} & m_{r2} & l_{r2} \end{pmatrix} \quad (1.24)$$

$$M_{sr1} = m_{sr1} \begin{pmatrix} \cos(\theta) & \cos(\theta + 120) & \cos(\theta - 120) \\ \cos(\theta - 120) & \cos(\theta) & \cos(\theta + 120) \\ \cos(\theta + 120) & \cos(\theta - 120) & \cos(\theta) \end{pmatrix} \quad (1.25)$$

$$M_{sr2} = \frac{m_{sr2}}{m_{sr1}} M_{sr1}, M_{sr1} = M^t_{r1s} \text{ et } M_{sr2} = M^t_{r2s}, M_{r1r2} = M^t_{r1r2} \quad (1.26)$$

$$M_{r1r2} = m_{r1r2} \begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 1 & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & 1 \end{pmatrix} \quad (1.27)$$

1.5.3. Équation du couple

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

$$C_{em} = [I_s]^t \frac{\partial [M_{sr1}(\theta)]}{\partial \theta} [I_{r1}] + [I_s]^t \frac{\partial [M_{sr2}(\theta)]}{\partial \theta} [I_{r2}] \quad (1.28)$$

1.5.4. Modèle de Park

Après l'application de la transformation de Park, nous avons simplifiée la complication des équations des tensions et des flux, ces transformations permettent de représenter notre machine par un système d'équations à coefficients constants.

Les équations complètes qui décrivent notre machine sont données comme suit :

$$\left\{ \begin{array}{l} V_{ds} = R_s I_{ds} + L_s \frac{d}{dt} I_{ds} + L_s (1 - \sigma_1) \frac{dI'_{dr_1}}{dt} + L_s (1 - \sigma_2) \frac{dI'_{dr_2}}{dt} \\ V_{qs} = R_s I_{qs} + L_s \frac{d}{dt} I_{qs} + L_s (1 - \sigma_1) \frac{dI'_{qr_1}}{dt} + L_s (1 - \sigma_2) \frac{dI'_{qr_2}}{dt} \\ V_{dr_1} = \omega I_{qs} + \frac{1}{T_{r_1}} I'_{dr_1} + \omega I'_{qr_1} + \omega \frac{ML_{m_2}}{L_{m_1} L_{r_2}} I_{qr_2} + \frac{dI_{ds}}{dt} + \frac{dI'_{r_1}}{dt} + \frac{ML_{m_2}}{L_{m_1} L_{r_2}} \frac{dI_{dr_2}}{dt} \\ V_{qr_1} = -\omega I_{ds} - \omega I_{dr_1} + \frac{1}{T_{r_1}} I'_{dr_1} - \omega \frac{ML_{m_2}}{L_{m_1} L_{r_2}} I'_{dr_2} + \frac{d}{dt} I_{qs} + \frac{dI'_{qr_1}}{dt} + \frac{ML_{m_2}}{L_{m_1} L_{r_2}} \frac{dI'_{qr_2}}{dt} \\ V_{r_2d} = L_{m_2} \frac{dI_{sd}}{dt} + M_{r_1 r_2} \frac{dI_{r_1d}}{dt} + R_{r_2} I_{r_2d} + L_{r_2} \frac{dI_{r_2d}}{dt} + P\Omega L_{m_2} I_{sq} + P\Omega M_{r_2 r_1} I_{r_1q} + P\Omega L_{r_2} I_{r_2q} \\ V_{r_2q} = L_{m_2} \frac{dI_{sq}}{dt} + M_{r_2 r_1} \frac{dI_{r_1q}}{dt} + R_{r_2} I_{r_2q} + L_{r_2} \frac{dI_{r_2q}}{dt} - P\Omega L_{m_2} I_{sd} - P\Omega M_{r_2 r_1} I_{r_1d} + P\Omega L_{r_2} I_{r_2d} \end{array} \right. \quad (1.29)$$

Les différentes inductances désignent :

L_s : Inductance cyclique propre d'une phase statorique.

L_{r_1} : Inductance cyclique propre d'une phase rotorique de rotor interne.

L_{r_2} : Inductance cyclique propre d'une phase rotorique de rotor externe.

L_{m_1} : Inductance cyclique mutuelle entre stator et rotor interne.

L_{m_2} : Inductance cyclique mutuelle entre stator et rotor externe.

$M_{r_1 r_2}$: Inductance cyclique mutuelle entre rotor interne et rotor externe.

$M_{r_1 r_2}$: Inductance cyclique mutuelle entre rotor externe et rotor interne.

L'expression du couple électromagnétique peut être obtenue par la formule suivante :

$$C_{em} = P^* \Omega \left\{ L_{m_1} (I_{sq} I_{r_1d} - I_{sd} I_{r_1q}) + L_{m_2} (I_{sq} I_{r_2d} - I_{sd} I_{r_2q}) \right\} \quad (1.30)$$

P désigne le nombre de pair de pôles de la machine.

Le système des équations (1.29), montre que le fonctionnement de la machine à double cage dépend de neuf paramètres électrique et de deux paramètres mécaniques $[R_s, R_{r1}, R_{r2}, L_s, L_{r1}, L_{r2}, L_{m1}, L_{m2}, M_{r1r2}]$, $[J, f_r]$.

Les paramètres qui dépend des rotors ne sont pas accessible directement à la mesures. Donc il faut aborder un changement de variable convenable pour faire identifier simultanément les paramètres électriques et pour déduire un modèle à sept paramètres électriques mesurables. Nous avons admet l'absence des inductances des fuites aux rotors.

1.5.5. Modèle à sept paramètres

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:

$$I_{dr1,2} = \frac{L_{m1,2}}{L_{r1,2}} I'_{dr1,2} \text{ et } I_{qr1,2} = \frac{L_{m1,2}}{L_{r1,2}} I'_{qr1,2} \quad (1.31)$$

En introduisant les coefficients des dispersions $\sigma_{1,2} = 1 - \frac{L_{m1,2}^2}{L_s L_{r1,2}}$, les constantes de temps rotorique $T_{r1,2} = \frac{L_{r1,2}}{R_{r1,2}}$ et statorique $T_s = \frac{L_s}{R_s}$.

Le système des équations (1.29) deviennent:

$$\left\{ \begin{array}{l} V_{ds} = R_s I_{ds} + L_s \frac{d}{dt} I_{ds} + L_s (1 - \sigma_1) \frac{dI'_{dr1}}{dt} + L_s (1 - \sigma_2) \frac{dI'_{dr2}}{dt} \\ V_{qs} = R_s I_{qs} + L_s \frac{d}{dt} I_{qs} + L_s (1 - \sigma_1) \frac{dI'_{qr1}}{dt} + L_s (1 - \sigma_2) \frac{dI'_{qr2}}{dt} \\ V_{dr1} = \omega I_{qs} + \frac{1}{T_{r1}} I'_{dr1} + \omega I'_{qr1} + \omega \frac{ML_{m2}}{L_{m1}L_{r2}} I_{qr2} + \frac{dI_{ds}}{dt} + \frac{dI'_{r1}}{dt} + \frac{ML_{m2}}{L_{m1}L_{r2}} \frac{dI_{dr2}}{dt} \\ V_{qr1} = -\omega I_{ds} - \omega I_{dr1} + \frac{1}{T_{r1}} I'_{dr1} - \omega \frac{ML_{m2}}{L_{m1}L_{r2}} I'_{dr2} + \frac{d}{dt} I_{qs} + \frac{dI'_{qr1}}{dt} + \frac{ML_{m2}}{L_{m1}L_{r2}} \frac{dI'_{qr2}}{dt} \\ V_{r2d} = L_{m2} \frac{dI_{sd}}{dt} + M_{r1r2} \frac{dI_{r1d}}{dt} + R_{r2} I_{r2d} + L_{r2} \frac{dI_{r2d}}{dt} + P\Omega L_{m2} I_{sq} + P\Omega M_{r2r1} I_q + P\Omega L_{r2} I_{r2q} \\ V_{r2q} = L_{m2} \frac{dI_{sq}}{dt} + M_{r2r1} \frac{dI_{r1q}}{dt} + R_{r2} I_{r2q} + L_{r2} \frac{dI_{r2q}}{dt} - P\Omega L_{m2} I_{sq} - P\Omega M_{r2r1} I_{r1d} + P\Omega L_{r2} I_{r2d} \end{array} \right. \quad (1.32)$$

L'équation de couple est donnée :

$$C_{em} = P * L_s [(1 - \sigma_1) * (I_{qs} \cdot I'_{dr1} - I_{ds} \cdot I'_{qr1}) + (1 - \sigma_2) * (I_{qs} \cdot I'_{dr2} - I_{ds} \cdot I'_{qr2})] \quad (1.33)$$

On pose :

$$P_1 = \frac{ML_{m1}}{L_{m2}L_{r1}}, P_2 = \frac{ML_{m2}}{L_{m1}L_{r2}} \quad (1.34)$$

On peut écrire P_2 en fonction de P_1 :

$$P_2 = P_1 \cdot \frac{1 - \sigma_2}{1 - \sigma_1} \quad (1.35)$$

Le système des équations (1.32) montrent que le fonctionnement de la machine dépend de sept paramètres électriques $[\sigma_1, \sigma_2, T_{r1}, T_{r2}, L_s, T_s, P_1]$ et de deux paramètres mécaniques $[J, f_r]$.

1.6. Identification paramétriques de la machine asynchrone

L'objectif de l'identification de la machine asynchrone est de déterminer les valeurs spécifiques de ces paramètres. Plusieurs techniques d'identification proposées sont basées sur l'exploitation du circuit équivalent ou sur le modèle dynamique de la machine [1], [19], [20], [24]–[29]. Ces techniques peuvent être divisées en trois groupes principaux :

- Le premier groupe représente l'usage d'IEEE et les normes IEC.
- Le deuxième groupe représente l'usage de données de catalogue/fabricant/plaque.
- Le troisième groupe représente l'usage de données mesurées pour l'estimation de paramètre de machine.

1.6.1. Identification basée sur les normes IEEE et IEC

L'IEEE recommande dans la norme 112 [30] différentes méthodes d'estimation et de calcul des paramètres de la machine asynchrone. Certaines des méthodes recommandées utilisent les données des essais sans charge ou à rotor verrouillé avec des fréquences nominales/non cotées (méthodes 1 et 2), tandis que d'autres utilisent les données des essais de glissement sans charge ou à charge complète avec une tension d'alimentation réduite (méthodes 3 et 4).

La méthode 1 nécessite un test à un maximum de 25% de la fréquence nominale, tandis que la méthode 2 nécessite des tests pour trois fréquences différentes. D'autre part, la méthode 3 suppose qu'un test doit être effectué avec ou sans charge couplée, tandis que la méthode 4 implique d'effectuer des essais de rotor verrouillé à fréquence nominale. Les méthodes 1, 2 et 4 exigent la mise

en œuvre de tests à jour évalués, c'est-à-dire une mesure précise du courant, tandis que la méthode 3 exige une mesure précise du glissement.

Outre les normes IEEE, il existe également une norme IEC [31]. La norme 60034-28 de l'IEC établit des procédures pour obtenir des valeurs pour les éléments de diagrammes de circuits équivalents à une phase à partir d'essais et définit les éléments standards de ces diagrammes.

Cette norme donne les procédures pour la détermination des paramètres de circuit équivalents à partir d'un essai de courbe de charge comme une alternative à la rotation inverse et les essais de rotor verrouillé. Par conséquent, on peut conclure que les normes de l'IEEE et de l'IEC pour la détermination des paramètres de machine asynchrone ne sont pas faciles à mettre en œuvre. Ils peuvent être erronés, ce qui est particulièrement mentionné dans [32].

1.6.2. Identification basée sur les catalogues/fabricants/plaques signalétiques

La plaque signalétique de machine asynchrone fournit des données très importantes. Cependant, différents fabricants de machines donnent des données différentes. Sur certaines machines, la plaque signalétique ne peut donner que des informations de base sur la machine, telles que sa tension nominale, la puissance et la vitesse, tandis que sur d'autres, la plaque signalétique peut également donner quelques informations liées aux données de couple.

Toutefois, la documentation technique de la machine (données du catalogue ou données du fabricant) pour les nouvelles machines asynchrones donne beaucoup d'informations. Pour cette raison, les méthodes de ce groupe publiées dans la littérature sont basées sur l'utilisation de différentes variables [32]–[35].

Par conséquent, on peut dire que les méthodes basées sur le catalogue / fabricant / données de plaque signalétique sont très appropriés pour les nouvelles machines asynchrones qui ont des données complètes du fabricant. Cependant, ils ne sont pas utiles pour les vieilles machines car ils n'ont pas de données de catalogue ou leurs données de catalogue ne contiennent pas d'informations détaillées. En outre, pendant le fonctionnement à long terme, les machines asynchrones modifient ses caractéristiques (matériau magnétique, isolement, excentricité, etc.) et ces méthodes ne peuvent pas être considérées comme une méthode standard pour l'estimation des paramètres.

1.6.3. Identification basée sur les données mesurées

Les méthodes basées sur des données mesurées nécessitent certaines mesures sur la machine observée. Pour cette raison, ces méthodes sont très fiables et reflètent l'état réel de la machine.

Ces méthodes sont très efficaces pour les anciennes machines asynchrones qui n'ont pas de données complètes du catalogue. Toutefois, pour leur mise en œuvre, un capteur précis doit être utilisé. Il a été noté plus tôt que la mesure du couple de la machine n'est pas nécessaire pour une représentation adéquate de la machine [32].

Toutefois, pour présenter une valeur de charge différente, une mesure de vitesse est nécessaire (sauf si nous appliquons des méthodes basées sur l'estimation des paramètres en temps réel, comme [36]). Par conséquent, des méthodes basées sur des mesures de la vitesse de phase, de la puissance d'entrée de la machine, du couple de la machine et du courant sont présentées dans [37].

En utilisant toutes ces données, les auteurs ont présenté un modèle de machine complet en tenant compte des changements de paramètres avec vitesse. D'autre part, les méthodes d'estimation des paramètres basées sur les mesures du courant de phase et des facteurs de puissance à différentes valeurs de vitesse sont décrites dans [38], [39]. En outre, les méthodes basées sur la mesure du couple appartiennent également à ce groupe [40].

1.6.4. Identification basée sur les techniques d'optimisation

L'estimation des paramètres de la machine asynchrone, qui utilisent le catalogue, la plaque signalétique du fabricant ou les données mesurées, nécessite de résoudre un grand nombre d'équations complexes. Les équations prédéfinies peuvent être résolues analytiquement ou en utilisant des techniques itératives ou d'optimisation. Les méthodes analytiques, telles que les méthodes décrites dans [41], [42] ainsi que celles basées sur des expériences en circuit ouvert et en circuit court, sont très simples à mettre en œuvre.

Toutefois, les méthodes analytiques nécessitent l'introduction d'hypothèses appropriées ou l'utilisation d'une formulation approximative. Par conséquent, ces

méthodes permettent d'obtenir des paramètres très rapidement avec des valeurs de faible précision. Les techniques itératives les plus couramment utilisées pour l'estimation des paramètres de la machine asynchrone sont basées sur l'utilisation de l'algorithme Newton-Raphson bien connu ou de l'algorithme Levenberg-Marquardt [37], [43].

Pour la mise en œuvre de ces algorithmes, certaines hypothèses ou données connues supplémentaires sont nécessaires. En outre, pour sa mise en œuvre, il est nécessaire de définir avec précision l'étape d'itération, les valeurs de départ et les critères d'itération appropriés.

D'autre part, de nombreux articles traitant de l'estimation des paramètres sont basés sur l'utilisation de différentes techniques d'optimisation (généralement méta heuristiques) [14-35,43]. L'utilisation de techniques d'optimisation nécessite la résolution de l'équation pour satisfaire à des critères prédéfinis ou à une fonction objectif.

Du point de vue des fonctions objectifs pour l'estimation des paramètres de machine asynchrone, le problème de trouver des paramètres de machine inconnus est réduit au problème de minimisation de l'écart entre les données mesurées, catalogue ou fabricant et les valeurs estimées d'une certaine variable. L'écart mentionné est connu sous le nom de fonction objectif.

Cette méthode est fondée sur l'existence d'un modèle mathématique dépendant de plusieurs paramètres qui décrit le comportement d'un processus physique dont on cherche à identifier et à évaluer ces paramètres.

Donc, quelque soient les méthodes d'identification, la stratégie que l'on adopte afin de parvenir à la détermination des paramètres est la suivante :

- ✓ Les données expérimentales suffisantes pour construire un modèle.
- ✓ Choix du modèle paramétrique.
- ✓ Recherche du vecteur optimal.
- ✓ Vérification de la validité du modèle.

Ainsi à partir d'un jeu de relevés expérimentaux mesurés sur des processus réels, la première chose nécessaire est la détermination d'un critère qui puisse exprimer ces données expérimentales afin d'aboutir à une fonction appelée de

coût, qui est l'indice de la capacité du modèle mathématique d'exprimer les données expérimentales.

De cette manière, l'objectif de l'identification revient à la recherche d'un vecteur de paramètres tel que le modèle puisse décrire le plus réellement possible le comportement des systèmes pour un type de fonctionnement [7, 8-9].

1.6.5. Méthode du modèle de référence

La méthode du modèle de référence permet d'une part de déterminer les paramètres de modèle lorsque les entrées-sorties sont quelconques et de l'autre, elle tolère des structures variées en ce qui concerne le modèle mathématique, lui-même. C'est pourquoi, cette méthode est extrêmement utilisée pour déterminer les paramètres des machines électriques dont le modèle est sous forme d'équations d'état ou de système d'équations différentielle.

La méthode du modèle de référence est basée sur la comparaison du comportement de l'objet (système à identifier) et celui du modèle. Tant qu'il n'y a pas d'identité de comportement, l'objectif de la méthode est de modifier le modèle de telle sorte que l'on tende vers une meilleure conformité.

Un critère est alors défini, dépendant implicitement des paramètres du modèle, et dont la minimisation assure l'atteinte de cette conformité optimale. Cet objectif de minimisation du critère conduit alors à l'utilisation d'algorithmes classiques qui font appel à la programmation non linéaire.

La méthode du modèle s'applique quelle que soit la structure du modèle (équations différentielles, fonctions de transfert, équations d'état, etc.). La figure 1.4 montre le schéma de principe de cette méthode.

Donc, cette méthode repose essentiellement sur quatre étapes à suivre :

- Choix d'un modèle qui décrit bien le régime transitoire de la machine asynchrone triphasée ;
- Choix convenable des signaux d'excitation réponse (signaux d'entrée sortie) ;
- Choix du critère d'évaluation entre le modèle et le processus ;
- Choix de la méthode de minimisation.

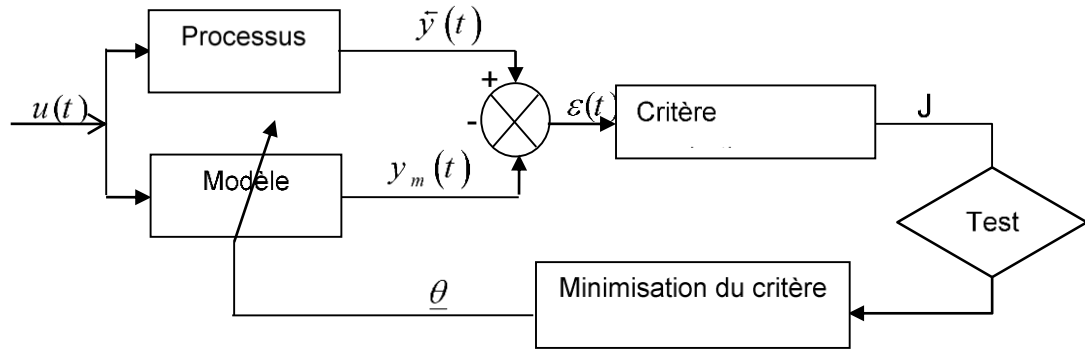


Figure 1.4: Schéma de principe de la méthode du modèle de référence

La structure du modèle dans notre cas est connue, il s'agit des équations de Park, qui décrivent la dynamique de la machine asynchrone triphasée. C'est donc, un modèle du type paramétrique dont les paramètres sont liés aux équations différentielles du système, décrites dans les sections 1.4.6 et 1.5.5. On doit appliquer la même excitation sur le modèle que celle appliquée sur le processus (système réels). L'estimation du vecteur paramètres est ensuite effectuée par minimisation du critère d'évaluation retenu. Ce dernier représente généralement l'erreur quadratique entre les grandeurs mesurées sur le système et les grandeurs calculées avec la même d'excitation.

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 générale, un rôle important sur la convergence et il est nécessaire de faire appel à des procédures particulières afin d'éviter les optimums locaux.

1.7. Conclusion

Ce chapitre a été consacré à l'identification et la modélisation de la machine asynchrone. Nous avons présenté les modèles la machine asynchrone (Modèle à une seule cage et modèle à double cage) dans la référence de PARK. Puis, nous avons présenté certains techniques d'identification de la machine asynchrone basant essentiellement sur l'existence du modèle qui représente fidèlement notre machine.

CHAPITRE 2

TECHNIQUES D'OPTIMISATION

2.1. Introduction

Dans le modèle de référence, un problème d'identification est devenu un problème d'optimisation. Dans ce chapitre, nous présentons une classification problèmes d'optimisation, puis nous exposons certaines méthodes d'optimisation déterministes et stochastiques. Enfin nous montrons une classification synergétique de l'hybridation de ces méthodes dans le but d'améliorer leurs convergences vers la solution optimale en un temps raisonnable.

2.2. Classification des problèmes d'optimisation

Les techniques d'optimisation sont classées suivant plusieurs critères tels que la nature de problème étudier, et à quelle catégorie est appartenu. En effet, les algorithmes développés sont conçus pour résoudre un type de problème donné et sont peu efficaces pour un type différent. Donc, la classification des problèmes d'optimisation change d'un auteur à l'autre. On distingue :

- Les problèmes d'optimisation continue versus les problèmes d'optimisation discrète.
- Les problèmes d'optimisation avec et sans contrainte.
- Les problèmes d'optimisation mono-objectif ou multi-objectif.
- Les problèmes d'optimisation déterministe ou stochastique.

2.3. Technique d'optimisation

Les méthodes utilisées pour la résolution des problèmes d'optimisation peuvent être classées en deux grandes catégories :

- Les méthodes exactes ou déterministes : garantissent la mise en fin de la recherche, mais très coûteuses en temps de calcul et parfois même inapplicables.
- Les méthodes approchées ou stochastiques : gagnent en efficacité, tout en perdant en complétude. Elles ne garantissent pas l'optimalité des solutions,

mais elles s'exécutent dans un temps raisonnable. elles sont une alternatives, intéressante lorsque l'optimalité n'est pas une nécessité

2.4. Méthodes déterministes

Ces méthodes utilisent la recherche par les directions les plus favorables, souvent donnée par le calcul du gradient de la fonction objectif par rapport aux paramètres, ou par essais successifs de l'espace de recherche. Ces méthodes peuvent être subdivisées en deux grandes sous classes :

- Méthodes déterministes à base de gradient (nécessitent le calcul de dérivées).
- Méthodes déterministes sans calcul de dérivées (méthodes d'ordre 0).

2.4.1.Méthodes déterministes à base de gradient

Elles sont basées sur l'évaluation de la dérivée à chaque itération. La direction de descente est donnée par le gradient [44].

2.4.1.1.Méthode du gradient

La méthode de gradient est une méthode itérative qui nécessite le calcul du gradient de la fonction à minimiser pour déplacer la solution vers une solution voisine. Le gradient donne seulement la direction de recherche. La procédure itérative est :

$$P_{k+1} = P_k + \lambda_k \cdot d_k \quad (2.1)$$

Dans un problème de minimisation, d_k est choisi dans la direction opposée à celle de gradient :

$$d_k = -grad(P_k) \quad (2.2)$$

Dans ces conditions le choix du pas du gradient λ_k est très important, il conditionne la bonne détermination de la direction de recherche. Les exemples les plus significatifs des méthodes mathématiques sont la méthode de Cauchy ou méthode de la plus grande pente [11] et la méthode du gradient conjugué [50] qui sont d'ordre un. D'autres techniques de gradient construisent une estimation du Hessien c'est-à-dire des dérivées secondes comme les méthodes DFP [13] et BFGS [14] qui sont dites quasi-Newton [50] [15] [11]. Parmi ces méthodes, la méthode du gradient conjugué, la méthode quasi-Newton, la méthode PQS sont présentées brièvement.

2.4.1.2.Méthode de gradient conjugué

La méthode de gradient conjugué [22] [50] [66] [3] est une variante améliorée de la méthode de la plus grande pente, qui consiste à suivre la direction opposée au gradient. Cette méthode à l'inconvénient de créer des directions de recherche orthogonales, ce qui ralentit la convergence de l'algorithme. La méthode de Fletcher et Reeves [50] résout ce problème en déterminant la nouvelle direction de recherche à partir du gradient aux pas courant et précédent.

2.4.1.3.Méthode quasi-Newton

Les méthodes quasi-Newton consistent à imiter la méthode de Newton où a l'optimisation d'une fonction est obtenue à partir de minimisations successives de son approximation au second ordre. Elles ne calculent pas le Hessien mais elles utilisent une approximation définie positive du Hessien qui peut être obtenue soit par l'expression proposé par Davidon-Fletcher-Powell (DFP) [14], soit par celle proposée par Broyden-Fletcher-Goldfard-Shanno (BFGS) [14].

2.4.1.4.Méthode PQS

La méthode de programmation quadratique séquentielle (PQS), développée par Schittkowski en 1983 [67], est une méthode de programmation non-linéaire. Elle a été reconnue comme étant une des méthodes les plus efficaces pour résoudre des problèmes d'optimisation avec contraintes de taille petite et moyenne. Comme son nom le suggère, la méthode PQS trouve la solution optimale par une séquence de problèmes de programmation quadratique. A chaque itération, une approximation quadratique de la fonction objectif et des approximations linéaires des contraintes sont utilisées. Le Hessien est construit par la méthode BFGS. Une analyse canonique fournit le minimum de la fonction Lagrangienne et un déplacement est fait vers ce point.

Ces méthodes convergent rapidement et précisément vers l'optimum si celui-ci est proche du point initial. Si ces méthodes sont intéressantes en raison de leur grande rapidité de convergence, elles ont plusieurs inconvénients :

1. Les valeurs de la fonction objectif et éventuellement de ses dérivées doivent être accessibles.
2. Lorsque le gradient de la fonction n'est pas calculable directement, sa détermination par la méthode des différences finies est toujours délicate à cause de problèmes liés au choix du pas de variation pouvant conduire à

des problèmes de convergence [50].

3. Ces méthodes nécessitent la résolution de systèmes matriciels pouvant être mal conditionnés [14].
4. La convergence est exclusivement locale. L'optimum trouvé dépend du point initial. La sensibilité par rapport aux conditions initiales est importante.

2.4.2. Méthodes déterministes sans calcul de dérivées

Elles sont beaucoup plus robustes que les méthodes à base de gradient. Elles explorent l'espace par essais successifs en recherchant les directions les plus favorables. Les plus répandus sont :

- Relaxation cyclique.
- Hooke et Jeeves.
- Rosenbrock.

2.4.2.1. Relaxation cyclique

Cette technique procède par minimisation unidirectionnelle successive le long des directions constituées par le vecteur paramètre en partant chaque fois du dernier point obtenu. Lorsque chaque variable a été examinée une fois, on peut recommencer un ou plusieurs cycles jusqu'à ce que le test d'arrêt de la procédure soit satisfait [45], [46]. Cette technique est illustrée par la figure 2.1 pour une itération et dans le cas de deux paramètres.

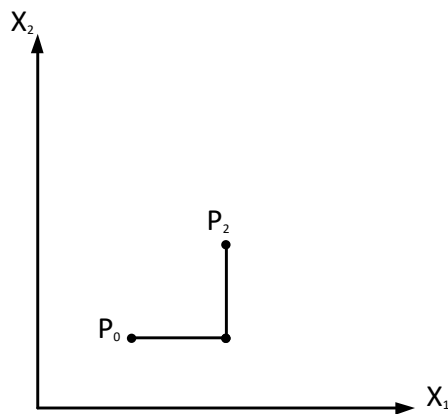


Figure 2.1 Illustration de la méthode de relaxation cyclique

L'organigramme de la méthode est présenté par la figure 2.2

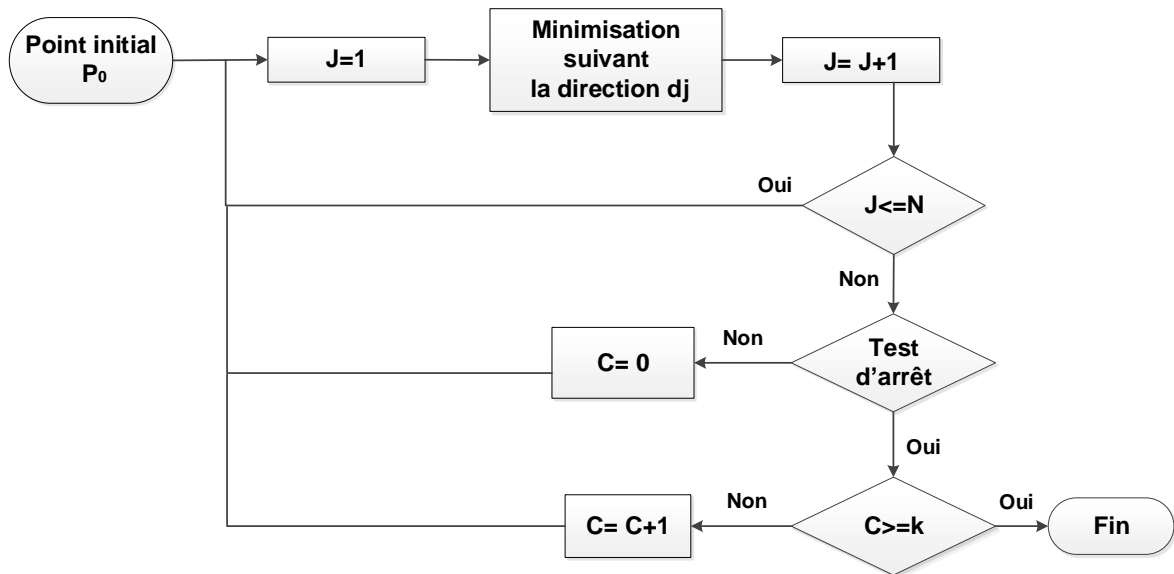


Figure 2.2 Organigramme de la méthode de relaxation cyclique

2.4.2.2. Hooke et Jeeves

La méthode de Hooke et Jeeves (1960) [47] appartient à la famille de méthodes directes, car elle ne nécessite pas le calcul de dérivées de la fonction objectif. Chaque itération de cette méthode se décompose en deux étapes :

- Une étape d'exploration qui consiste à se déplacer suivant les directions de base, de manière à améliorer la valeur de la fonction objectif.
- Une étape de progression qui consiste à faire une projection suivant les deux derniers vecteurs ayant amélioré la fonction objectif.

La figure 2.3 illustre la méthode de Hooke et Jeeves dans le cas de deux paramètres. La procédure répétitive de cet algorithme est :

1. Soit P_1 le point initial.
2. La minimisation selon les directions constituant les paramètres donne le point P_2 .
3. La minimisation suivant la direction P_1P_2 donne le point P_3 .
4. En démarrant du point P_3 , une nouvelle recherche selon les axes donne le point P_4 .
5. Une recherche selon la direction P_2P_4 donne le point suivant.

Cette procédure itérative ne s'arrête jusqu'à ce que le test sur l'erreur quadratique soit satisfait. La figure 2.4 montre l'organigramme de l'algorithme

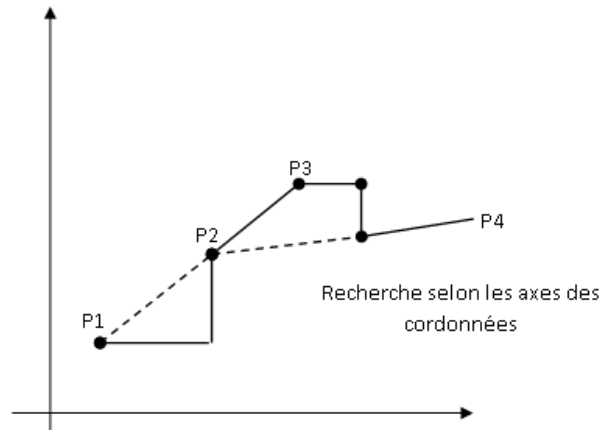


Figure 2.3: Illustration de la méthode Hooke et Jeeves

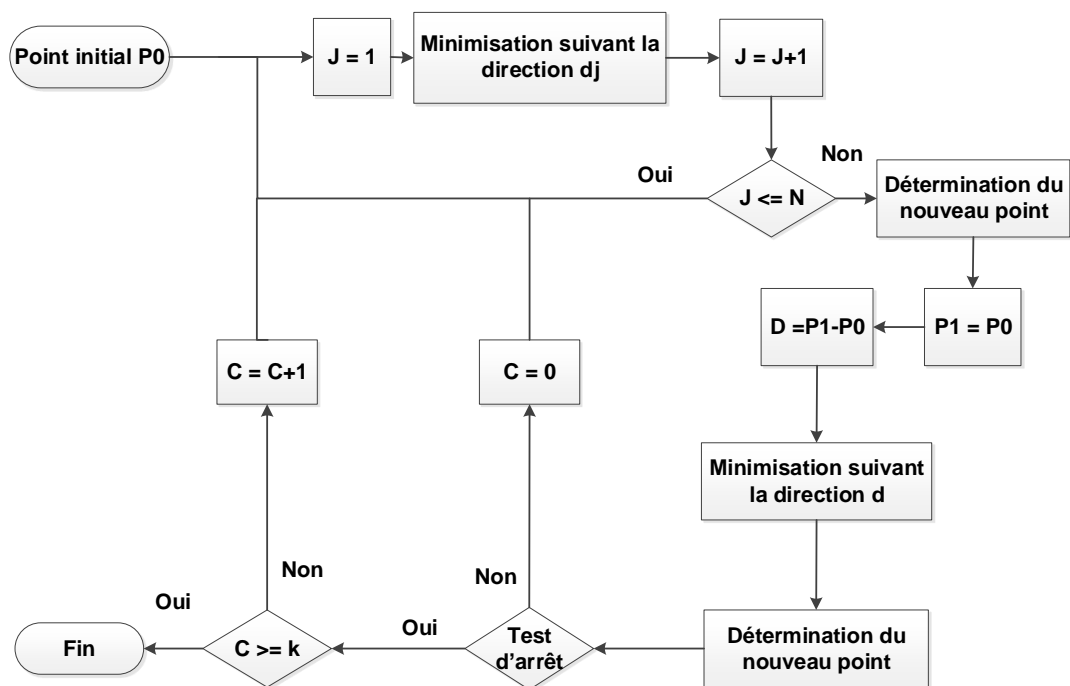


Figure 2.4: Algorithme de la méthode Hooke et Jeeves

2.4.2.3. Rosenbrock

La méthode de Rosenbrock (dite aussi « rotation de coordonnées ») permet de rendre l'exploration indépendante des directions initiales. C'est une méthode itérative qui se décompose en deux étapes répétitives [48].

- Une étape d'exploration qui réalise des améliorations successives via des directions privilégiées.
- Une étape globale qui permet de construire une nouvelle base dans la direction de deux sorties consécutives.

La vitesse de convergence de la méthode dépend du choix de la base de départ. De plus, elle nécessite la reconstruction d'une nouvelle base à chaque itération ce qui rend la méthode coûteuse lorsque la dimension du vecteur des variables de décision est grande. La méthode de Rosenbrock est illustrée par la figure 2.5 dans le cas bidimensionnel. L'organigramme de l'algorithme est donné par la figure 2.6.

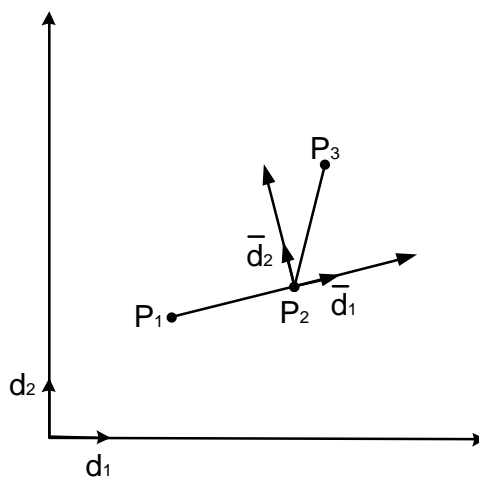


Figure 2.5 Illustration de la méthode de Rosenbrock

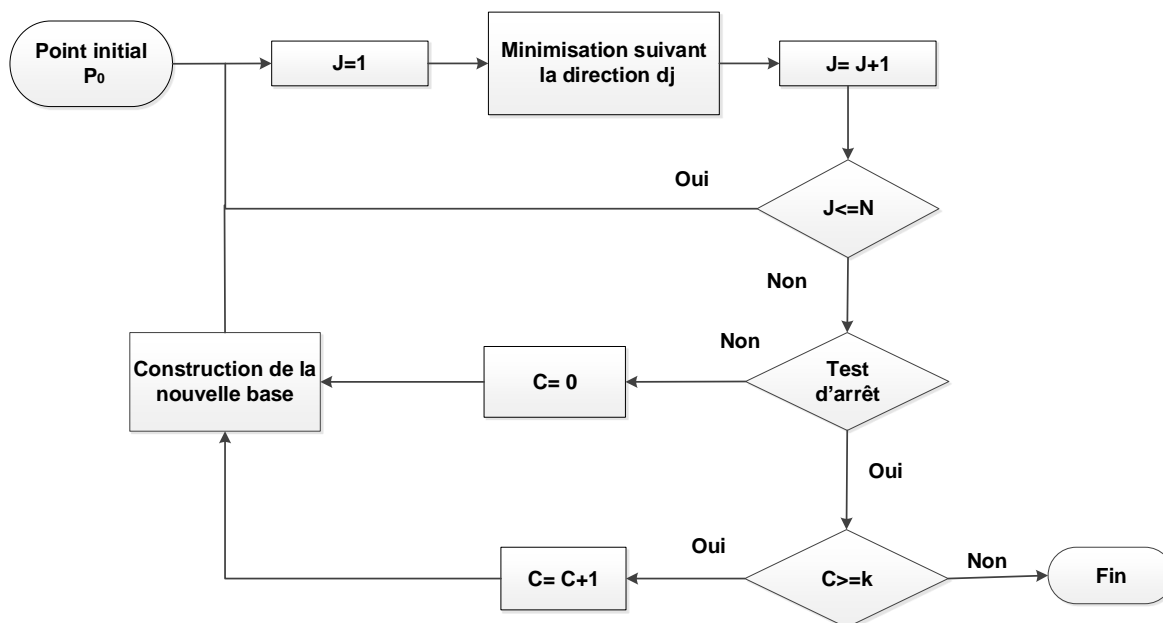


Figure 2.6 Organigramme de la méthode de Rosenbrock

2.5. Méthodes stochastiques

Les méthodes d'optimisation stochastiques s'appuient sur des mécanismes de transition probabilistes et aléatoires. Ces méthodes ont une grande capacité à

trouver l'optimum global du problème. Contrairement à la plupart des méthodes déterministes. Elles ne nécessitent aucune connaissance a priori sur la fonction objectif pour atteindre la solution optimale.

Par ailleurs, elles constituent une alternative pour traiter les problèmes d'optimisation de grande taille [49]–[51]. Cependant, elles nécessitent un volume de calcul important pour atteindre la solution du problème.

On peut subdiviser les méthodes stochastiques en deux familles :

- Les heuristiques : peut être définie comme étant une procédure exploitant au mieux la structure du problème considéré dans le but de trouver une solution de qualité raisonnable dans des délais courts que possible.
- Les métaheuristiques : est un processus de génération itératif qui guide une heuristique sous-jacente en combinant intelligemment différents concepts pour exploiter et explorer l'espace de solutions afin d'atteindre la solution optimale.

La biologie, la physique et l'éthologie est l'origine d'inspiration de plusieurs métaheuristiques. Ces métaheuristiques peuvent être réparties en deux classes :

Les méthodes à base de solution unique (voisinage) :

- Recherche gloutonne aléatoire adaptative.
- Recherche à voisinage variable
- Recherche locale itérée

Les méthodes évolutionnaires :

- Essaims particulaires.
- Colonies de fourmis.
- Algorithme de luciole.

2.5.1. Méthodes à base de solution unique

Ces méthodes utilisent la notion de voisinage pour améliorer une solution itérativement. La recherche commence par une solution initiale x_0 , en construisant pas à pas une nouvelle solution voisine afin d'obtenir une solution optimale pour que la fonction objectif soit minimale [52]. Parmi ces méthodes, nous évoquons la recherche gloutonne aléatoire adaptative, la recherche à voisinage variable et la recherche locale itérée.

2.5.1.1. La recherche gloutonne aléatoire adaptative

La procédure de recherche gloutonne aléatoire adaptative (GRASP : Greedy Randomized Adaptive Search Procedure), proposée par Feo et Resende dans [53], [54], est une métaheuristique à départs multiples, dépourvue de mémoire, fondée sur les algorithmes gloutons randomisés et les techniques de recherche de voisinage. Chaque itération de l'algorithme GRASP se compose de deux étapes dites de construction et de recherche locale.

L'étape de la construction est similaire à l'heuristique semi-gloutonne (semi-greedy heuristic) proposée indépendamment par Hart et Shaoguan [55]. Elle génère une solution réalisable par l'application d'une procédure d'initialisation gloutonne. Dans la deuxième étape, cette solution est utilisée comme solution initiale de la procédure de recherche locale. Après un nombre donné d'itérations, l'algorithme GRASP se termine et la meilleure solution trouvée est conservée. Une étude bibliographique de la méthode GRASP est fournie dans [56][57].

2.5.1.2. La recherche à voisinage variable

La recherche à voisinage variable (VNS : Variable neighborhood search) est une métaheuristique proposée par Hansen et Mladenovic en 1997 [58], [59]. Elle est basée sur le principe de changement systématique de voisinage durant la recherche.

À l'étape d'initialisation, un ensemble de structures de voisinage doit être définie. Ses voisinages peuvent être choisis arbitrairement, mais souvent une séquence $N_1, N_2, \dots, N_{n_{\max}}$ est rangée par ordre de tailles croissantes des voisinages qu'elle utilise. La procédure de VNS se compose de trois étapes : perturbation (shaking), recherche locale et déplacement. Étant donnée une solution initiale s , l'étape de perturbation génère au hasard une solution s' dans le n ème voisinage de la solution courante s .

Une procédure de recherche locale est alors appliquée, avec s' comme solution initiale, afin d'obtenir un optimum local s'' . Si s'' est meilleure que s , alors s'' remplace s et on génère une nouvelle solution dans le voisinage N_1 . Dans le cas contraire, la solution courante reste s et on change de voisinage en générant une solution s' dans le $n+1$ ème voisinage. Plus généralement, on change de voisinage à chaque fois que l'un d'entre eux n'est pas parvenu, après application de la procédure de recherche locale, à améliorer la solution courante s . Par

contre, dès qu'un voisinage permet d'améliorer s , alors on recommence le processus avec le premier voisinage.

Cet algorithme est efficace si les structures de voisinage sont complémentaires en ce sens qu'un minimum local pour un voisinage n'en n'est pas nécessairement un pour un autre.

2.5.1.3. La recherche locale itérée.

La recherche locale itérée (ILS : Iterated local search) [60] est une métaheuristique basée sur une idée simple : au lieu de l'application répétée d'une procédure de recherche locale à partir de solutions générées aléatoirement, il génère la solution de départ pour la prochaine itération en appliquant une perturbation sur l'optimum local trouvé à l'itération courante. Ceci est fait dans l'espoir que le mécanisme de perturbation fournit une solution située dans le bassin d'attraction d'un meilleur optimum local.

Le mécanisme de perturbation est un élément clé de ILS : d'une part, une perturbation trop faible peut ne pas être suffisante pour s'échapper du bassin d'attraction de l'optimum local actuel ; d'autre part, une perturbation trop forte correspond à une recherche locale multi-départs à partir de combinaisons initiales générées aléatoirement.

Un critère d'acceptation définit les conditions que le nouvel optimum local p^* doit satisfaire afin de remplacer l'optimum local courant s^* . Le critère d'acceptation, combiné avec le mécanisme de perturbation, permet de contrôler le compromis entre l'intensification et la diversification. Par exemple, un critère d'acceptation qui favorise l'intensification doit accepter seulement les solutions qui sont des améliorations à la valeur généralement optimale.

En revanche, le critère qui favorise la diversification accepte toutes les solutions perturbées, quelle que soit leur qualité. De nombreux critères d'acceptation qui concilient les deux objectifs peuvent être appliqués [4].

2.5.2. Les métaheuristiques à population de solutions

Les méthodes d'optimisation à population de solutions améliorent, au fur et à mesure les itérations d'une population de solutions. On distingue dans cette catégorie, les algorithmes évolutionnaires comme les algorithmes génétiques, et les algorithmes d'intelligence en essaim. L'intérêt de ces méthodes est d'utiliser la population comme facteur de diversité.

Parmi ces méthodes, nous distinguons essentiellement l'algorithme d'essaims particuliers, les colonies de fourmis, l'algorithme de luciole, l'optimiseur de loup gris et l'algorithme d'optimisation sinus cosinus.

2.5.2.1. Algorithme d'essaims particuliers

Les algorithmes d'optimisation par essaim de particules (PSO) ont été introduits en 1995 par Kennedy et Eberhart [61]. Ces algorithmes sont inspirés des essaims de particules (ou des bancs de poissons ou des nuées d'oiseaux) et de leurs comportements collectifs de mouvements. En effet, tout comme ces animaux se déplacent en groupe pour trouver de la nourriture ou éviter les prédateurs.

L'essaim particulier est constitué d'une population des agents. Chaque agent correspond à une solution proposée du problème traité. Elle est caractérisée par :

- Sa position courante x_i dans l'essaim.
- Sa vitesse V_i .

À chaque itération, l'agent va communiquer à l'agent voisin sa propre position courante, et sa propre expérience qui est dans ce cas la mémoire de la meilleure position qu'elle a visitée. Pour construire le mouvement dans l'essaim chaque agent va actualiser sa nouvelle vitesse et sa nouvelle position. La vitesse et la position sont mise à jour en tenant compte :

- La propension à suivre son propre chemin.
- Sa tendance à revenir vers sa meilleure position atteinte.
- Sa tendance à aller vers son meilleur voisin suivant la formule cité ci-dessous :

$$v_i(t+1) = C_1 * v_i(t) + C_2 * rand * (x_{pbest} - x_t) + C_3 * rand * (x_{vbest} - x_t) \quad (2.3)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2.4)$$

- x_{pbest} est sa meilleure solution visitée par l'agent.
- x_{vbest} est la meilleure solution visitée par sa propre voisine.

Le coefficient d'inertie C_1 joue un rôle important dans la convergence. Une faible valeur de C_1 nous guide à une exploitation locale tandis qu'une grande valeur nous permet une exploration de l'espace de recherche. C_1 représente

l'attraction de la particule vers sa meilleure position, c'est la mémoire propre à la particule, tandis que $C2$ représente l'attraction vers la meilleure de ses voisines engendrant la mémoire partagée.

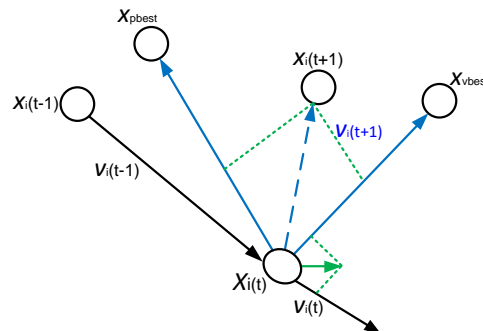


Figure 2.7 Présentation de PSO

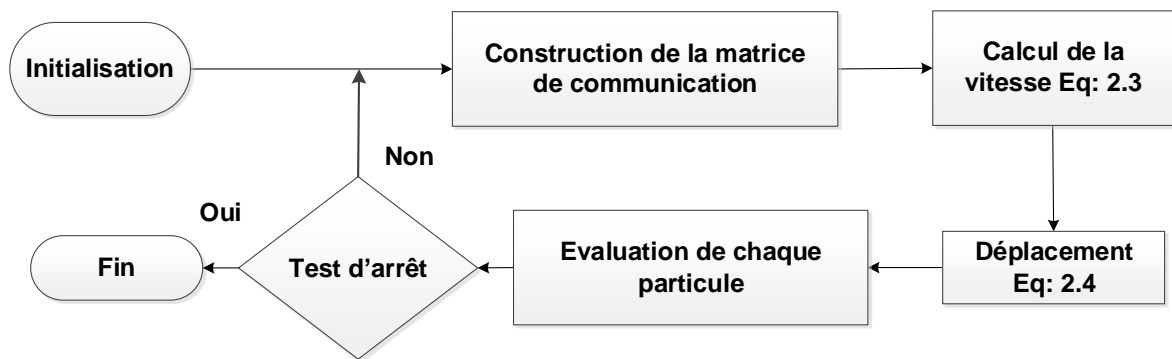


Figure 2.8 Organigramme de l'Algorithme d'essaims particulaires

2.5.2.2. La colonie de fourmis

La colonie de fourmis est un algorithme d'optimisation inspiré de la nature qui remonte à l'étude du comportement de fourmis à la recherche de nourriture au départ de leur nid [62]–[65]. En se déplaçant du nid à la source de nourriture et vice-versa. 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 marquer de chemin, quand les fourmis choisissent leur chemin, elles ont tendance à choisir la piste qui porte la plus forte concentration de phéromones qui correspond au chemin le plus court. 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.

Dans une itération de l'algorithme de colonies de fourmis, f agents (fourmis) construisent chacun une solution d'après des décisions basées sur des critères de recherche et sur des traces de phéromone. Les traces sont mises à jour en examinant les solutions obtenues. Elles sont renforcées pour les décisions ayant donné de meilleures solutions et diminuées pour les autres. Ce mécanisme permet d'améliorer progressivement les solutions au cours des itérations jusqu'à la réalisation d'un critère d'arrêt.

- Mise à jour des traces de phéromone dans le réseau ;
- Génération de « f » nouvelles solutions par les fourmis, en exploitant les traces de phéromone ;
- Application d'une probabilité donnée pour une recherche locale de ses solutions.

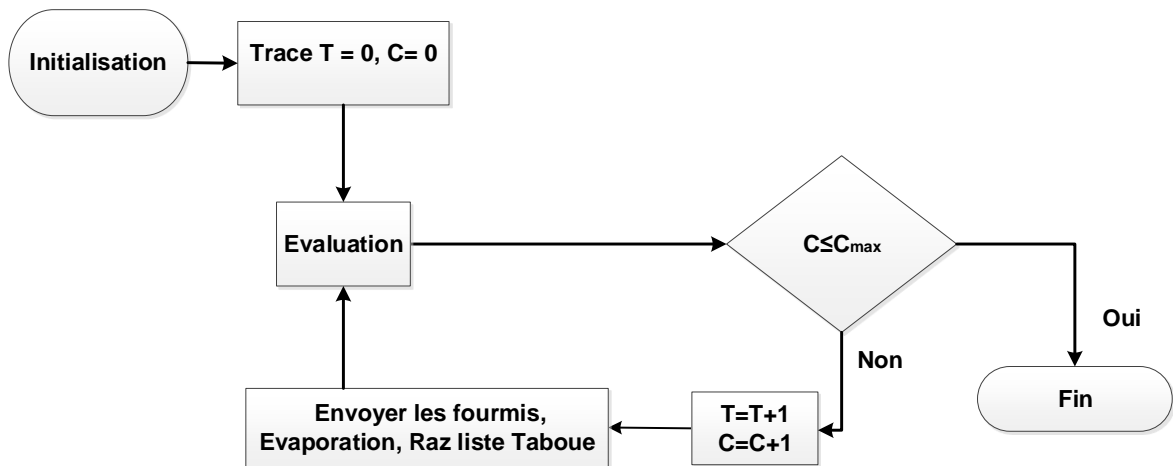


Figure 2.9 Organigramme de l'Algorithme de La colonie de fourmis

2.5.2.3. Algorithme de luciole

L'algorithme de luciole (en Anglais Firefly algorithm) est une métaheuristique introduit par Xin-She Yang [66]. Elle est inspirée du comportement d'émission et d'absorption de la lumière des lucioles. La méthode est basée sur le principe d'attraction mutuelle entre les lucioles et simule le comportement d'un essaim de lucioles dans la nature. Ce qui lui donne de nombreuses similitudes avec autres métaheuristiques basées sur l'intelligence collective du groupe [67].

Pour simplifier la description de l'algorithme de luciole, nous devons idéaliser certaines des caractéristiques clignotantes des lucioles. Pour cela nous prendrons en compte trois règles idéalisées suivantes [68], [69]:

1. Toutes les lucioles sont unisexes, ce qui fait l'attraction entre celles-ci n'est pas en fonction de leur sexe.
2. L'attraction est proportionnelle à leurs luminosités, donc pour deux lucioles, la moins lumineuse se déplacera vers la plus lumineuse. Si aucune luciole n'est lumineuse qu'une luciole particulière, cette dernière se déplacera aléatoirement.
3. La luminosité des lucioles est déterminée en fonction d'une fonction objectif (à optimiser).

L'évolution des lucioles dont le but est d'orienter la recherche vers la meilleure localisation prometteuse se base sur :

- La variation de l'intensité de la luminosité.
- La formulation de l'attractivité mutuelle entre les lucioles.

Pour un problème de minimisation, la luminosité « I » de chaque luciole est inversement proportionnelle à la fonction objectif.

L'attractivité « β » est relative à la position des autres lucioles, la variation de l'attractivité entre deux lucioles adjacentes est donnée par l'équation suivante :

$$\beta = \beta_0 e^{-\gamma r^2} \quad (2.5)$$

Où « r » est la distance entre deux lucioles, β_0 est l'attractivité initiale à « $r=0$ », et « γ » est le coefficient d'absorption.

La distance « r » peut être déterminée à l'aide de la distance euclidienne :

$$r_{ij} = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (2.6)$$

Avec :

- $x_{i,k}$ est le k ème composante de la position de la luciole « i » ;
- $x_{j,k}$ est le k ème composante de la position de la luciole « j » ;

Le mouvement d'une luciole « i » est attiré par un autre plus attractant (plus lumineux) luciole « j » est déterminé par :

$$x_i^{t+1} = x_i^t + \beta(x_j^t - x_i^t) + \alpha_t(rand - 0.5) \quad (2.7)$$

Où le second terme est dû à l'attraction et le troisième terme est dû la randomisation, tandis que le coefficient « α_t » est le coefficient de randomisation.

Après le déplacement, la nouvelle position de luciole a calculé sa nouvelle intensité. Pendant le déplacement des lucioles la meilleure luciole est mise à jour leur position et leur intensité lumineuse de manière itérative.

Sur la base des trois règles de simplification et l'équation de déplacement les étapes de base de l'algorithme de luciole (**AL**) peuvent être résumées par l'organigramme suivant :

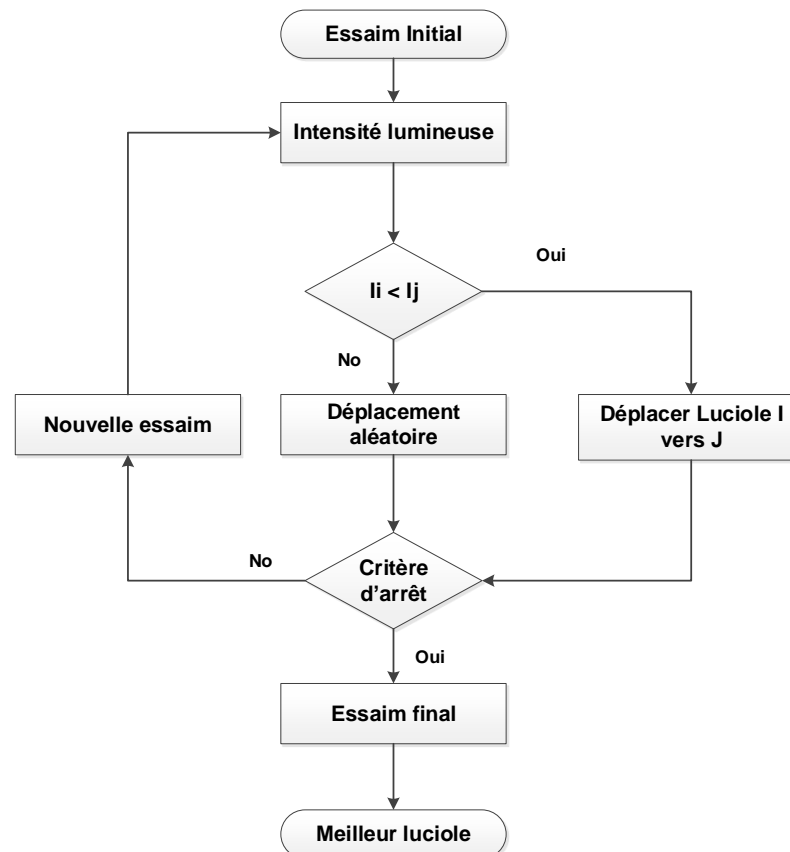


Figure 2.10 Organigramme de l'Algorithme de luciole

2.6. Hybridation des algorithmes d'optimisation

L'hybridation consiste à exploiter les avantages respectifs de deux ou plusieurs méthodes en combinant leurs algorithmes suivant une approche synergétique. Une méthode hybride peut être mauvaise ou bonne selon le choix et les rôles de ses composants. Pour pouvoir construire une méthode hybride efficace, il faut combiner les avantages et surmonter sur les limitations de chacune des méthodes qui le composent.

Par exemple, une méthode est très performante lorsqu'il s'agit d'explorer l'espace de recherche, mais il s'avère ensuite incapable d'exploiter efficacement la zone vers laquelle la population des solutions converge. Il est alors plus

intéressant d'utiliser dans ce stade une autre méthode permettant la bonne exploitation de la zone.

Les premières approches hybrides proposées ont combiné des métaheuristiques à base de populations (algorithmes génétiques) avec des métaheuristiques à solution unique (recuit simulé ou recherche tabou) [72]–[74].

Plusieurs approches hybrides combinant les métaheuristiques et les méthodes exactes ont été proposées dans la littérature [33], [75]. Cette approche peut devenir une alternative très intéressante car les deux méthodes ont des particularités bien différentes qui peuvent être associées pour produire de meilleurs résultats.

2.6.1. Motivation de l'hybridation

Malgré le succès des métaheuristiques pour résoudre les problèmes complexes, plusieurs problèmes ont été rencontrés, à savoir :

1. La présence de bruits lors de l'évaluation des solutions.
2. La distinction entre une solution optimale locale et une autre globale est parfois difficile à percevoir.
3. Le problème de la convergence prématurée.
4. Le problème d'ajustement des paramètres.

L'hybridation des méthodes bio-inspirées a connu une grande popularité et a permis de bénéficier des points forts de chacune de ces méthodes et de surmonter leurs limites.

L'hybridation a permis d'avoir un compromis entre l'exploration et l'exploitation de l'espace de recherche des solutions. Pour avoir une bonne exploitation, un algorithme est utilisé pour localiser les meilleures régions de l'espace de recherche, un autre est utilisé pour converger vers l'optimum global. L'hybridation est utilisée aussi pour optimiser les paramètres généraux.

On peut conclure que l'hybridation des métaheuristiques est la voie la plus prometteuse pour l'amélioration de la qualité des solutions dans beaucoup d'applications réelles. Ainsi, le choix d'une approche hybride devient aujourd'hui déterminant pour obtenir de meilleures performances lors de la résolution des problèmes complexes.

2.6.2. Classification des stratégies d'hybridation

On peut classer les différentes hybridations selon la taxinomie proposée par Talbi [80]. Cette classification permet de comparer les métaheuristiques hybrides de façon qualitative. La taxonomie comporte deux aspects : une classification hiérarchique permet d'abord d'identifier la structure de l'hybridation, ensuite, une classification générale spécifie les détails des algorithmes impliqués dans l'hybridation.

2.6.2.1. Classification hiérarchique

La classification hiérarchique des méthodes hybrides est subdivisée en deux classes :

- **méthodes intégratives ou bas niveau** : lorsqu'une métaheuristique remplace un opérateur d'une autre métaheuristique qui l'englobe, a pour but d'utiliser les propriétés de diversification de la métaheuristique de base et d'intensifier la recherche avec l'hybridation embarquée.
- **Méthodes coopératives ou haut niveau** : chaque métaheuristique garde sa structure entière et fonctionne de manière indépendante. Elle consiste à combiner une métaheuristique avec d'autres méthodes sans que leurs fonctionnements internes ne soient en relation. Le but est d'exécuter une séquence de méthodes.

Chacune de deux classes d'hybridation précédentes se subdivise en deux autres classes :

- **méthodes à relais** : lorsque les métaheuristiques sont exécutées de façon séquentielle, l'une utilisant le résultat de la précédente comme entrée.
- **Méthodes Co-évolutionnaire** : se fait lorsque des agents coopèrent en parallèle pour explorer l'espace de solutions.

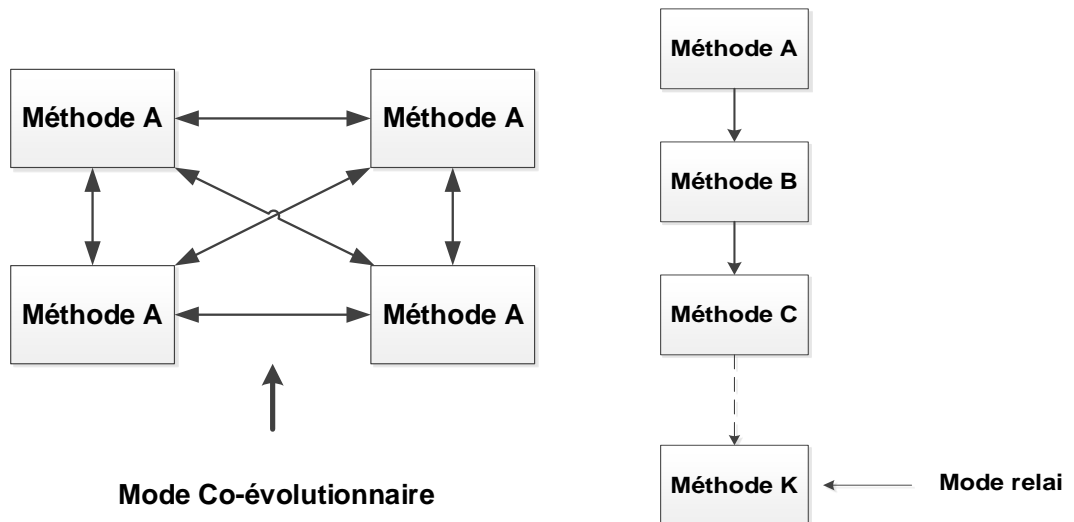


Figure 2.11 Hybridation Parallèle (à gauche), Séquentielle (à droite)

La combinaison de toutes les classes nommées précédemment donne quatre classes différentes :

- **hybridation relais de bas niveau** : représente les algorithmes dans lesquels une métaheuristique est incorporée dans une autre métaheuristique à solution unique.
- **Hybridation Co-évolutionnaire de bas niveau** : consiste à incorporer un algorithme de recherche axée sur l'exploitation à une métaheuristique à population axée sur l'exploration exécutée en parallèle. L'avantage de ce type d'hybridation est de compenser la puissance d'exploitation d'une recherche locale et celle d'exploration d'une recherche globale ; ces deux types de métaheuristicques ont des forces et faiblesses complémentaires : l'exploitation raffine une solution tandis que l'exploration fouille plus largement l'ensemble des solutions.

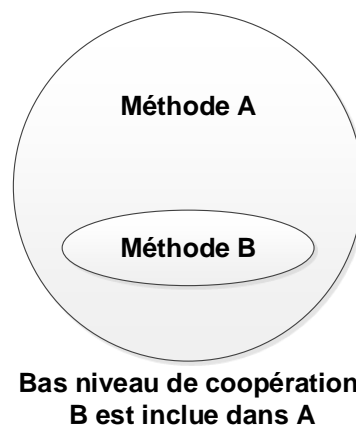


Figure 2.12 Hybridation bas niveau de coopération

- **Hybridation relais de haut niveau** : se fait lorsque des métaheuristiques complètes sont exécutées séquentiellement. Par exemple, on peut améliorer la solution d'un algorithme à population en utilisant comme solution de départ d'un algorithme de recherche locale.
- **Hybridation co-évolutionnaire de haut niveau** : implique un ensemble de métaheuristiques complètes qui travaillent en parallèle et coopèrent pour trouver la solution optimale d'un problème.

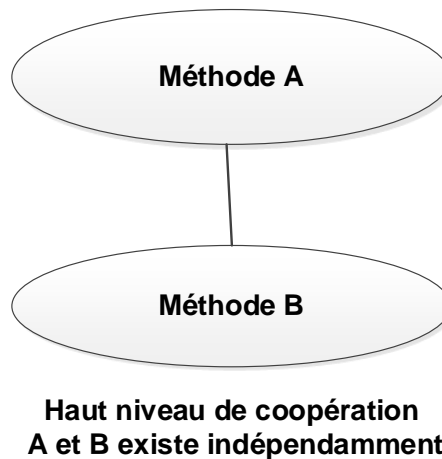


Figure 2.13 Hybridation haut niveau de coopération

2.6.2.2. Classification générale

La classification générale de la taxonomie de Talbi [80] comporte trois dichotomies :

- Les approches hybrides homogènes versus hétérogènes ;
- Globales versus partielles ;
- Spécialistes versus générales.

Une hybridation est dite homogène lorsque les métaheuristiques combinées sont identiques. À l'inverse, une hybridation hétérogène combine des métaheuristiques différentes.

Une hybridation globale fait en sorte que toutes les métaheuristiques explorent l'ensemble de l'espace de solutions. D'un autre côté, l'hybridation partielle décompose un problème en sous problèmes ayant leur espace de solutions, et chaque sous problème est donné par un algorithme.

L'hybridation générale est celle où tous les algorithmes résolvent le même problème d'optimisation. L'hybridation spécialiste est celle où chaque algorithme résout un problème d'optimisation différent.

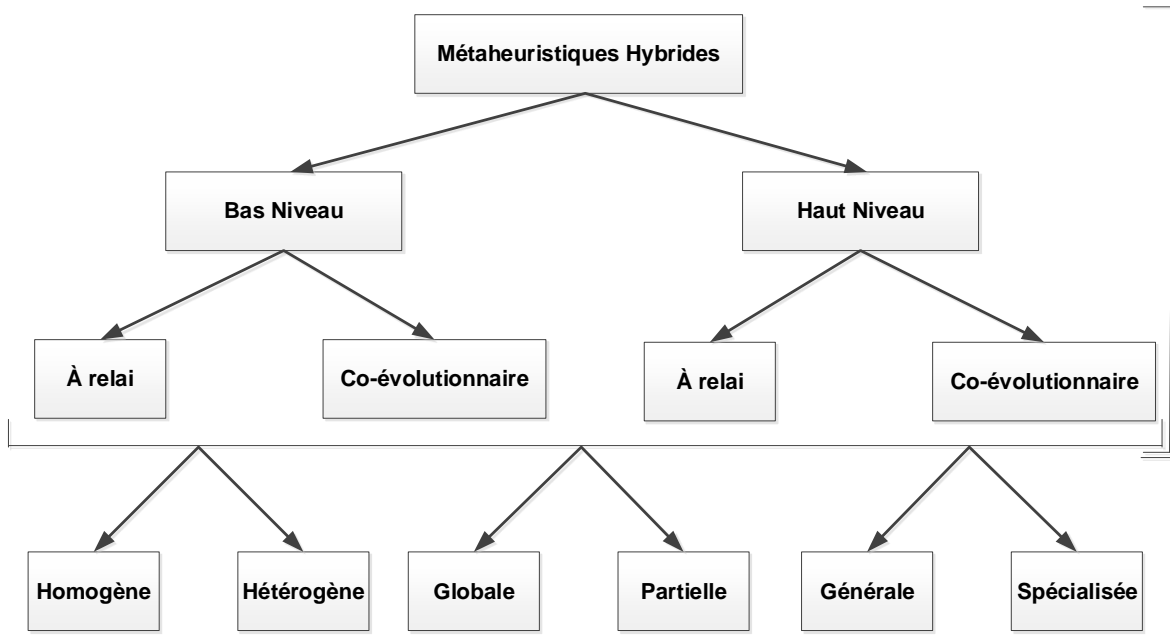


Figure 2.14 Classification des stratégies d'hybridation

2.7. Conclusion

Ce chapitre, nous a permis de passer en revue les différentes techniques d'optimisation qui peuvent être réparties en deux catégories : les méthodes déterministes sans et avec le calcul de dérivée, les méthodes stochastiques qui reposent sur des règles de transition probabiliste. Ce chapitre a permis également de présenter une classification hiérarchique et une classification générale des méthodes hybrides afin d'améliorer les avantages des méthodes qui le compose.

CHAPITRE 3

HYBRIDATION ET AMELIORATION DE L'ALGORITHME DE LUCIOLE

3.1. Introduction

L'algorithme de luciole est un algorithme simple, flexible et facile à comprendre et à mettre en œuvre. Il a été appliqué avec succès pour l'optimisation continue [60], l'optimisation multimodale [51], ou l'optimisation sous contrainte [61]. Il est largement utilisé pour résoudre de nombreux problèmes d'ingénierie [62]–[65], cependant l'algorithme standard de luciole a tendance à être piégé dans plusieurs optimums locaux et une convergence prématurée lors de la résolution d'applications complexes d'ingénierie dans le monde réel. Dans ce chapitre, nous présentons trois variantes de l'algorithme de Luciole, trouvées dans la littérature, puis nous proposons dans un premier temps d'améliorer la version standard en orientant le mouvement des lucioles vers le centroïde global courant de l'essaim dans le cas où le mouvement vers la luciole le plus brillant est en échec. Dans un deuxième temps, on va hybrider ledit algorithme avec deux autres méta heuristiques : l'optimiseur de loup gris et l'algorithme d'optimisation sinus cosinus. Nous allons vérifier les algorithmes proposés via plusieurs tests en utilisant des fonctions dédiées aux tests des algorithmes d'optimisation.

3.2. Implémentation de l'algorithme de luciole

L'implémentation de l'algorithme de luciole passe par les étapes suivantes :

1. Un réglage faisable du vecteur contrôle (tableau 3.1) qui contient tous les paramètres prédéfinis de l'algorithme de luciole.
2. Génération aléatoirement d'une population initiale bien diversifiée utilisant le codage réel car il est précis et moins encombrant que le codage binaire. Cette population doit être représentée un ensemble de solutions possibles dans un domaine de recherche bien déterminée.
3. Calculer l'intensité lumineuse de chaque luciole.
4. Comparer l'intensité de chaque deux luciole adjacente.
5. Déplacer la luciole moins lumineuse vers la plus lumineuse.
6. Répéter l'étape 3 jusqu'à ce que la condition d'arrêt soit satisfaite.

Tableau 3.1 Paramétrages des algorithmes

Paramètre	Valeur
Nombre maximal d'itérations	2500
Taille de la population	50
Coefficient d'absorption	2
Coefficient d'attraction	2

3.3. Problèmes de Test (Fonction de référence)

Il existe de nombreux algorithmes d'optimisation dans des domaines très variés. Une question fondamentale qui se pose ici est : « comment pouvons-nous comparer différents algorithmes d'optimisation de manière à pouvoir choisir le plus performant pour un problème d'optimisation donné ? » Demeurent toutefois des questions relatives à l'utilité d'une métaheuristique particulière pour résoudre un large éventail de problèmes.

Des protocoles de tests génériques pour la comparaison d'algorithmes existent et nous fournissent des éléments de réponse. En effet, plusieurs fonctions analytiques de tests, regroupées sous l'appellation de « benchmarks », ont été mises en place pour évaluer les performances et les capacités de convergence des algorithmes d'optimisation.

Les fonctions de tests sont définies dans le cadre de la conférence 2005 **IEEE Congress on Evolutionary Computation (CEC'05)**. L'optimum global de ces fonctions est connu. Ces fonctions de tests sont classées en trois groupes :

- les fonctions unimodales à haute dimension (**F1 à F7**) : elles ne possèdent qu'un seul minimum global. Elles sont conçues pour tester la capacité d'exploitation des métaheuristicques.
- Les fonctions multimodales à haute dimension (**F8 à F13**) : elles sont des fonctions hautement multimodales, elles ont plusieurs optima locaux. Elles sont conçues pour tester la capacité d'exploration et l'évitement de la convergence prématurée.
- Les fonctions multimodales à faible dimension (**F14 à F23**) : elles sont similaires à la catégorie précédente, mais avec des dimensions basses, donc un nombre inférieur d'optima locaux.

Tableau 3.2 Caractéristiques des fonctions-tests utilisées pour la comparaison

Fonction-test	Domaine	Dimension	f_{min}
Fonctions uni modales (7)			
<i>F1 Sphere Function</i>	[-100, 100]	10	0
<i>F2 Schwefel's 2.22 Function</i>	[-10, 10]	10	0
<i>F3 Schwefel's 1.2 Function</i>	[-100, 100]	10	0
<i>F4 Schwefel's 2.21 Function</i>	[-100, 100]	10	0
<i>F5 Generalized Rsenbrock's Function</i>	[-30, 30]	10	0
<i>F6 Step Function</i>	[-100, 100]	10	0
<i>F7 Quartic Function with Noise</i>	[-1.28, 1.28]	10	0
Fonctions multimodales à haute dimension (7)			
<i>F8 Schwefel's 2.26 Function</i>	[-500, 500]	10	-2094.915
<i>F9 Generalized Rastrigin's Function</i>	[-5.12, 5.12]	10	0
<i>F10 Ackley Function</i>	[-32, 32]	10	0
<i>F11 Generalized Griewank Function</i>	[-600, 600]	10	0
<i>F12 Generalized Penalized Function 1</i>	[-50, 50]	10	0
<i>F13 Generalized Penalized Function 2</i>	[-50, 50]	10	0
<i>F14</i>	[-65, 65]	2	1
<i>F15</i>	[-5, 5]	4	0.0003
<i>F16</i>	[-5, 5]	2	-1.0316
<i>F17</i>	[-5, 5]	2	0.398
<i>F18</i>	[-2, 2]	2	3
<i>F19</i>	[1, 3]	3	-3.86
<i>F20</i>	[0, 1]	6	-3.32
<i>F21</i>	[0, 10]	4	-10.1532
<i>F22</i>	[0, 10]	4	-10.4028
<i>F23</i>	[0, 10]	4	-10.5363

Les résultats des simulations présentés dans la section résultats correspondent à la valeur moyenne μ obtenue sur nombre fixe d'exécutions $N_{exc}=25$, l'écart-type σ entre les différentes exécutions et la meilleure valeur trouvée. Le résultat de chaque algorithme proposé est comparé avec l'algorithme de luciole originale. Les expériences sont effectuées avec les 23 fonctions de tests avec une dimension égale à **30**.

Les algorithmes ont été initialisés de la même manière de sorte que la comparaison soit la plus pertinente possible. La première génération est distribuée uniformément au sein des intervalles de définition de chaque fonction de test. Pour plus de clarté, les solutions obtenues, qui sont de meilleure qualité, ont été mises en gras.

3.4. Amélioration de l'algorithme de luciole

Le développement d'un algorithme de luciole passe par les étapes suivantes:

1. la génération d'un essaim (population) initial de taille « N » qui représente les locations initiales des lucioles.
2. Le calculer de l'intensité lumineuse de chaque luciole.
3. La définition de coefficient d'absorption lumineuse, et le coefficient d'attraction.
4. Une comparaison de l'intensité de chaque deux lucioles adjacentes.
5. Un déplacement de la luciole moins lumineuse vers la plus lumineuse.

Plusieurs améliorations de l'algorithme de luciole standard sont soumises pour remédier au problème de tomber sur des optima locaux et pour augmenter la vitesse de convergence lors de la résolution de problèmes complexes d'ingénierie réels. Tahereh et al. [14] ont trouvé que l'optimum global affecte le mouvement des lucioles tout en comparant la luminosité de deux lucioles. La formule de mise à jour modifiée est définie par :

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\mathcal{M}_{i,j}^2} (x_j^t - x_i^t) + \beta_0 e^{-\mathcal{M}_{i,best}^2} (x_{best}^t - x_i^t) + \alpha_i (rand - 0.5) \quad (3.1)$$

Cette amélioration a l'avantage de ne pas tomber sur un optimum local.

Mehrnoosh [15] a réduit l'effet de la randomisation progressivement par l'ajout d'une dimension sociale à chaque luciole. La formule de mise à jour modifiée est définie par :

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\mathcal{M}_{i,j}^2} (x_j^t - x_i^t) + \alpha_i (rand - 0.5) + \rho (rand - 0.5) (x_i^t - g_{best}^t) \quad (3.2)$$

Où ρ est un paramètre similaire à α .

Divya Gupta et Medha Gupta [13] ont proposé une nouvelle formule de mise à jour modifiée :

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\mathcal{M}_{i,j}^2} (x_j^t - x_i^t) + \beta_0 e^{-\mathcal{M}_{i,best}^2} (x_{best}^t - x_i^t) + \alpha_i (rand - 0.5) + \rho (rand - 0.5) (x_i^t - g_{best}^t) \quad (3.3)$$

Cette formule combine toutes les meilleures caractéristiques des variantes proposées de l'algorithme de luciole. Elle réduit le caractère aléatoire de l'algorithme pour obtenir rapidement la convergence et influence le mouvement des lucioles vers l'optimum global pour réduire la probabilité de l'algorithme à se coincer dans un optimum local.

3.4.1. Algorithme de luciole amélioré, proposition 01

L'algorithme de luciole standard peut parfois cesser de converger vers l'optimum global [13]–[16]. Il est impossible d'obtenir un algorithme de luciole général qui résout de manière efficace tous les problèmes d'optimisation du monde réel, ce qui motive les chercheurs à développer d'autres améliorations de l'algorithme donnant de meilleures solutions approximatives.

L'évolution de recherche vers la solution optimale dans l'algorithme de luciole par la formule de mise à jour se base sur l'existence des meilleures lucioles que la luciole courante. Cette formule se construit de deux composantes le premier est dû à l'attractivité de la luciole la plus attrayante et le deuxième est la composante aléatoire, ce dernier est utile pour la détermination de la nouvelle location dans le domaine de recherche, elle ayant un impact décisif lors de l'exploration de l'espace. L'absence des meilleures lucioles voisines s'oblige la luciole de déplacer aléatoirement, ce mouvement ralentit la vitesse de convergence. Cela dégrade les performances de l'algorithme par conséquent l'algorithme tombe dans le problème de convergence prématurée et ce qui oriente l'algorithme d'être piégé dans des optimums locaux. Pour pallier cet inconvénient et éviter la stagnation de convergence, nous proposons de faire déplacer les lucioles vers la centroïde globale de l'essaim de l'itération actuelle dans le cas où le mouvement vers le plus attrayant est en échec. La nouvelle structure de déplacement est garantie en utilisant l'équation suivante:

$$x_i^{t+1} = x_i^t + rand * (x_g^t - x_i^t) \quad (3.4)$$

$$x_g^t = \frac{\sum \alpha_i * essaim(i,:)}{\sum \alpha_i} \quad (3.5)$$

Le coefficient de pondération a été obtenu par une distribution parabolique statique. L'algorithme de luciole amélioré peut augmenter l'exploration au début de la recherche lorsque le centroïde de l'essaim ne soit pas proche de l'optimum global, et augmenter l'exploitation dans la zone du centroïde à la fin de la recherche lorsque le centroïde de l'essaim soit proche de l'optimum global.

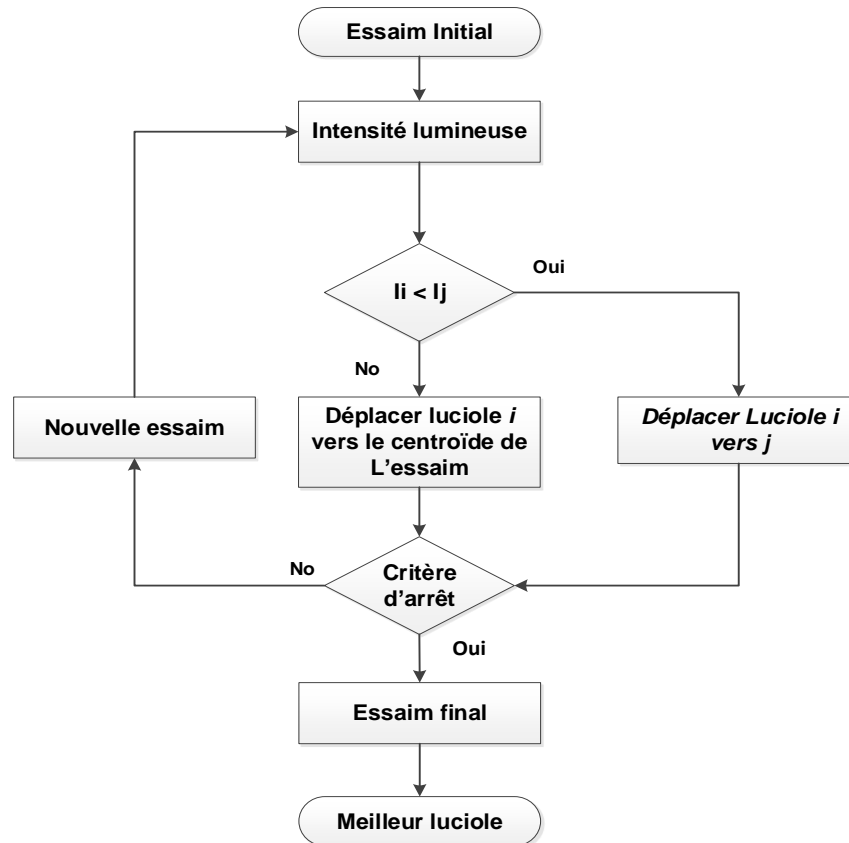


Figure 3.1 Présentation de l'Algorithme de luciole amélioré

3.4.2. Résultats d'optimisation avec les fonction de référence

L'algorithme hybride est testé sous les 23 fonctions de référence citée auparavant pour étudier ses performances par rapport à l'algorithme de luciole, les résultats sont présentés dans le tableau 3.2.

3.5. Hybridation de l'algorithme de luciole avec l'optimiseur de loups gris

Pour remédier aux faiblesses d'exploitation de l'algorithme de luciole standard et pour augmenter sa capacité explorative de recherche, on vise à l'hybrider avec l'optimiseur de loups gris. Avant de passer à l'algorithme proposé, nous commençons d'abord par introduisant l'optimiseur de loups gris (OLG).

3.5.1. Optimiseur de loups gris

L'optimiseur de Loup gris s'inspire de la hiérarchie sociale et de la méthode de chasse intelligente des loups gris. La hiérarchie sociale des loups gris est structurée comme suit [70]:

- le loup alpha (α) (le leader) est la meilleure solution.
- Les loups (β), (δ) sont le deuxième et le troisième meilleures solutions

Tableau 3.3 Résultats d'optimisation

Fonction	AL		ALA	
	μ	σ	μ	σ
F1	1.4E+04	3.8E+03	1.9E+03	7.7E+01
F2	7.4E+02	1.5E+03	1.1E+01	4.5E+00
F3	1.5E+04	4.5E+03	4.7E+03	2.2E+03
F4	6.4E+01	7.7E00	2.3E+01	6.1E00
F5	3.1E+07	1.5E+07	3.2E+05	2.6E+05
F6	1.4E+04	3.7E+03	1.5E+03	4.5E02
F7	5.1E00	2.6E00	2.6E-01	2.1E-01
F8	-1.3 E+03	2.6 E+02	-9.4 E+03	6.6 E+03
F9	1.2E+02	9.2E+01	6.0E+01	1.3E+01
F10	1.9E+01	5.0E-01	1.2E+01	1.8E+01
F11	1.3E+02	3.4E+01	5.4E00	00E00
F12	5.7E+07	3.4E+07	4.6E+04	1.2E+05
F13	1.1E+08	6.9E+07	2.2E+05	4.3E+05
F14	1.1E+02	1.2E+02	3.2E00	3.4E00
F15	1.2E-01	9.3 E-02	1.9 E-02	1.8 E-02
F16	2.2E-01	5.8E-01	-1.0E00	3.0E-02
F17	1.1 E00	7.7E-01	5.4 E-01	3.1E-01
F18	3.6 E+01	4.2 E+01	1.3 E+01	1.2 E+01
F19	-3.3 E+00	0.3 E-01	-3.6 E+00	1.9 E+01
F20	-1.5 E+00	5.5 E-01	-2.2 E+00	4.8 E-01
F21	-6.4 E-01	3.2 E-01	-5.3 E+00	2.4 E+00
F22	-7.9 E-01	3.1 E-01	-4.3 E+00	1.9 E+00
F23	-9.1 E-01	3.0 E-01	-3.9 E+00	2.3 E+00

- Les loups Omega (ω) sont les autres solutions.

La chasse au loup gris comprend les trois parties principales suivantes :

1. suivre, chasser et approcher la proie.
2. Poursuivre, encercler et harceler la proie jusqu'à ce qu'elle arrête de bouger.
3. Attaquer la proie.

L'optimiseur du loup gris modélise le comportement des loups, et assimile la recherche du minimum d'une fonction coût à la stratégie de chasse. La recherche des parois (la solution prometteuse) et la chasse est guidée par α , β et δ , et les loups ω . Le suivi de la proie correspond à l'exploration ou recherche globale, l'attaque correspond à l'exploitation ou recherche locale.

L'optimiseur du loup gris est formulé comme suit [70]

Le comportement d'encerclement :

$$\vec{d} = |\vec{c} \cdot \vec{x}_p - \vec{x}(t)| \quad (3.6)$$

$$\vec{x}(t+1) = \vec{x}_p - \vec{a} \cdot \vec{d} \quad (3.7)$$

Où « t » indique l'itération courante, « X_p » est le vecteur de position de la proie, et « X » indique le vecteur de position d'un loup gris. Les vecteurs « A » et « C » sont calculés par :

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (3.8)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (3.9)$$

\vec{r}_1 , \vec{r}_2 sont des vecteurs aléatoires dans [0, 1].

\vec{a} Est une fonction qui décroît régulièrement de 2 à 0 entre $t = 1$ et $t = T$.

Ainsi, un loup gris peut mettre à jour sa position dans l'espace autour de la proie en utilisant des équations aléatoires. Les trois premières meilleures solutions sont sauvegardées et l'autre agent est obligé de mettre à jour ses positions en fonction de la position des meilleurs agents de recherche, comme indiquent les équations suivantes :

$$\begin{cases} \vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \\ \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \\ \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \end{cases} \quad (3.10)$$

$$\begin{cases} \vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha) \\ \vec{X}_2 = \vec{X}_\beta - \vec{A}_1 \cdot (\vec{D}_\beta) \\ \vec{X}_3 = \vec{X}_\delta - \vec{A}_1 \cdot (\vec{D}_\delta) \end{cases} \quad (3.11)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (3.12)$$

Cette procédure se répète jusqu'à l'obtention de la meilleure solution. L'optimiseur de loups gris peut être résumée par l'organigramme suivant :

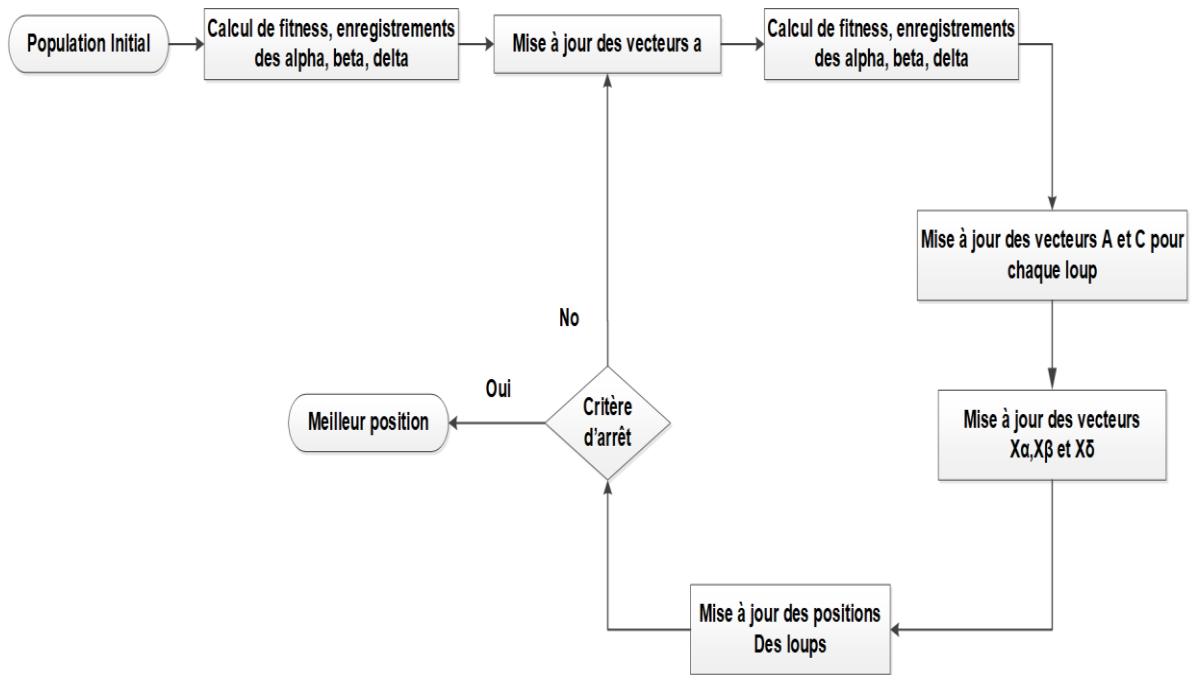


Figure 3.2 Présentation de l'optimiseur de loups gris (OLG)

3.5.2. Algorithme hybride proposé, proposition 02

Le type de l'hybridation dans ce cas est **relais de bas niveau**. La variante hybride proposée est appelée algorithme de luciole hybride avec optimiseur loup gris **HALOLG (HFAGWO)**. L'incorporation de l'optimiseur de loups gris est dans le cas où le mouvement vers la luciole la plus attirante est en échec. L'algorithme hybride se présente deux phases comme suit :

1. Le déplacement de luciole i vers la luciole la plus attractive est assuré par la formule de mise à jour de luciole standard.
2. Le déplacement de luciole i vers une luciole désigné par l'optimiseur de loup gris au lieu de déplacer de façon aléatoire.

Dans cet algorithme, l'algorithme de luciole est utilisé pour faire converger l'essaim vers des zones plus prometteuses dans l'espace de recherche, puis l'optimiseur de loup gris aide à exploiter les zones déjà explorer pour converger vers la solution optimale.

L'algorithme hybride fonctionne mieux que l'algorithme de luciole standard, car il maintient un équilibre entre les capacités d'exploration et d'exploitation offertes par l'algorithme de luciole et l'optimiseur de loup gris. L'algorithme hybride proposé peut être résumé par l'organigramme présenté dans la figure

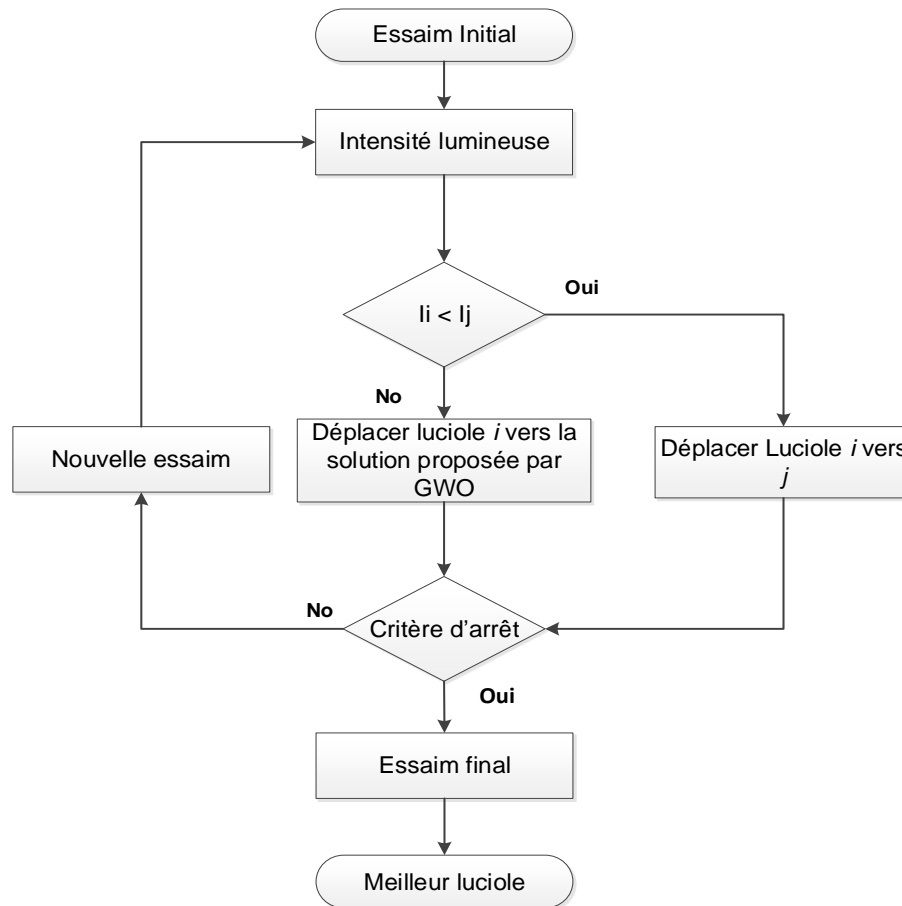


Figure 3.3 Algorithme hybride proposée ALOLG

3.5.3. Résultats d'optimisation avec les fonctions de référence

L'algorithme hybride est testé sous les 23 fonctions de référence citée auparavant pour étudier ses performances par rapport à l'algorithme de luciole, les résultats sont présentés dans le tableau 3.3.

3.6. Hybridation de l'algorithme de luciole avec l'algorithme sinus cosinus

Dans cette section nous proposons d'hybrider l'algorithme de luciole avec l'algorithme d'optimisation sinus cosinus. Nous commençons d'abord par introduisant l'algorithme d'optimisation sinus cosinus. Puis, nous proposons deux algorithmes hybride 1 et 2.

3.6.1. Algorithme d'optimisation sinus cosinus

Mirjalili [71] en 2016 a proposé un nouvel algorithme d'optimisation basé sur les formes du sinus et du cosinus en mathématiques, appelé algorithme sinus cosinus. L'algorithme commence par générer des solutions aléatoires. Il recherche ensuite la meilleure solution en effectuant plusieurs itérations.

Tableau 3.4 Résultats d'optimisation

Fonction	AL		ALOLG	
	μ	σ	μ	σ
F1	1.4E+04	3.8E+03	1.2E-250	00E00
F2	7.4E+02	1.5E+03	1.8E-308	00E00
F3	1.5E+04	4.5E+03	1.7E-131	8.8E-131
F4	6.4E+01	7.7E00	3.E-145	1.5E-144
F5	3.1E+07	1.5E+07	8.9E00	4.6E-02
F6	1.4E+04	3.7E+03	0.95103	0.35929
F7	5.1E00	2.6E00	1.8E-05	2.1E-05
F8	-1.3 E+03	2.6 E+02	-1.3E+136	6.2E+136
F9	1.2E+02	9.2E+01	00E00	00E00
F10	1.9E+01	5.0E-01	8.8E-16	00E00
F11	1.3E+02	3.4E+01	00E00	00E00
F12	5.7E+07	3.4E+07	2.8E-01	2.1E-01
F13	1.1E+08	6.9E+07	5.1E-01	1.9E-01
F14	1.1E+02	1.2E+02	8.2E00	4.6E00
F15	1.2E-01	9.3 E-02	9.9 E-03	1.1 E-02
F16	2.2E-01	5.8E-01	-1.0E00	5.1E-02
F17	1.1 E00	7.7E-01	4.1E-01	4.1 E-02
F18	3.6 E+01	4.2 E+01	9.7 E+00	1.0 E+01
F19	-3.3 E+00	0.3 E-01	-3.7 E+01	1.3 E+01
F20	-1.5 E+00	5.5 E-01	-2.6 E+00	3.8 E-01
F21	-6.4 E-01	3.2 E-01	-4.9 E+00	1.3 E+00
F22	-7.9 E-01	3.1 E-01	-4.4 E+00	1.9 E+00
F23	-9.1 E-01	3.0 E-01	-5.0 E+00	2.2 E+00

Pendant ces itérations, l'espace de recherche est maintenu en ajustant la plage du sinus et du cosinus en fonction de leurs formes mathématiques. Ces boucles sont répétées jusqu'à ce que la condition d'arrêt soit satisfaite. La forme mathématique du SCA est déterminée comme suit :

$$x_i^{t+1} = \begin{cases} x_i^t + r_1 * \sin(r_2) * |r_3 * x_1^t - x_i^t|, r_4 > 0.5 \\ x_i^t + r_1 * \cos(r_2) * |r_3 * x_1^t - x_i^t|, r_4 \leq 0.5 \end{cases} \quad (3.13)$$

3.6.2. Algorithme hybride proposé, proposition 03

Le type d'hybridation, dans ce cas, est un relais de bas niveau. La variante hybride proposée est appelée HALASC1 (HFASCA1). Le premier algorithme hybride est basé sur les règles de mouvement suivantes :

1. le déplacement de luciole i vers la luciole la plus attractive est assuré par l'équation de mouvement de l'algorithme de luciole standard.

2. Le déplacement de luciole i vers une location désignée par l'algorithme d'optimisation sinus cosinus si le mouvement vers la luciole le plus attractant est en échec, suivant l'équation ci-dessous

$$x_i^{t+1} = \begin{cases} x_i^t + r_1 * \sin(r_2) * |r_3 * x_1^t - x_i^t| + 0.5 * |x_1^t - x_i^t|, r_4 > 0.5 \\ x_i^t + r_1 * \cos(r_2) * |r_3 * x_1^t - x_i^t| + 0.5 * |x_1^t - x_i^t|, r_4 \leq 0.5 \end{cases} \quad (3.15)$$

L'organigramme de l'algorithme hybride proposé est montré par la figure 3.4.

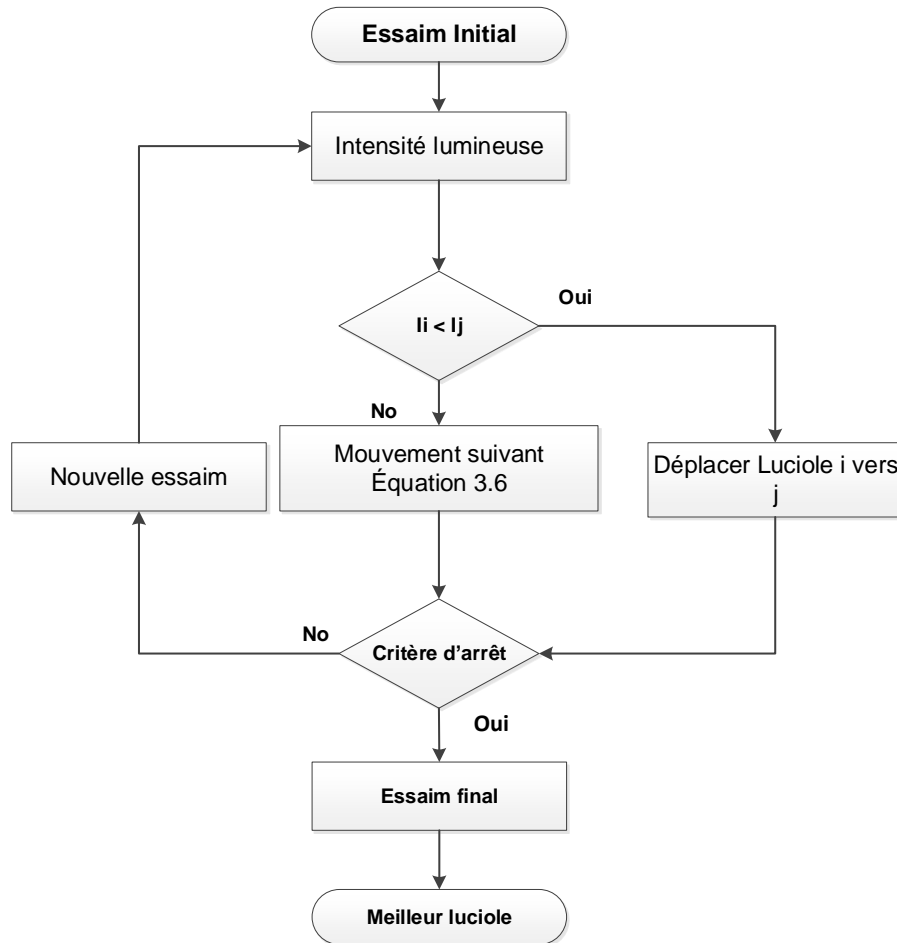


Figure 3.4 Algorithme hybride proposée ALASC1

3.6.3. Algorithme hybride proposé, proposition 04

Le type d'hybridation, dans ce cas, est un relais de bas niveau. La variante hybride proposée est appelée HALASC2 (HFASCA2). Dans ce cas, l'incorporation de l'algorithme de luciole avec l'algorithme d'optimisation sinus cosinus permet d'augmenter la capacité exploratoire de la formule de mise à jour autour de la meilleure luciole. Tout d'abord, nous proposons de modifier l'équation standard du mouvement de luciole à la plus brillante en supprimant le troisième terme de

l'équation 2.7 qui caractérise la randomisation et son remplacement par les équations de l'algorithme d'optimisation sinusoïdale cosinus comme suivant :

$$x_i^{t+1} = \begin{cases} x_i^t + \beta * (x_j^t - x_i^t) + r_1 * \sin(r_2) * |r_3 * x_1^t - x_i^t|, r_4 > 0.5 \\ x_i^t + \beta * (x_j^t - x_i^t) + r_1 * \cos(r_2) * |r_3 * x_1^t - x_i^t|, r_4 \leq 0.5 \end{cases} \quad (3.7)$$

Ensuite, On remplace le mouvement aléatoire des lucioles s'il n'y a pas d'autres solutions meilleur que celle-là par le mouvement inspiré de l'algorithme d'optimisation sinus-cosinus.

$$x_i^{t+1} = \begin{cases} x_i^t + r_1 * \sin(r_2) * |r_3 * x_1^t - x_i^t|, r_4 > 0.5 \\ x_i^t + r_1 * \cos(r_2) * |r_3 * x_1^t - x_i^t|, r_4 \leq 0.5 \end{cases} \quad (3.8)$$

Ce nouvel algorithme hybride est représenté par l'organigramme de la figure 3.5.

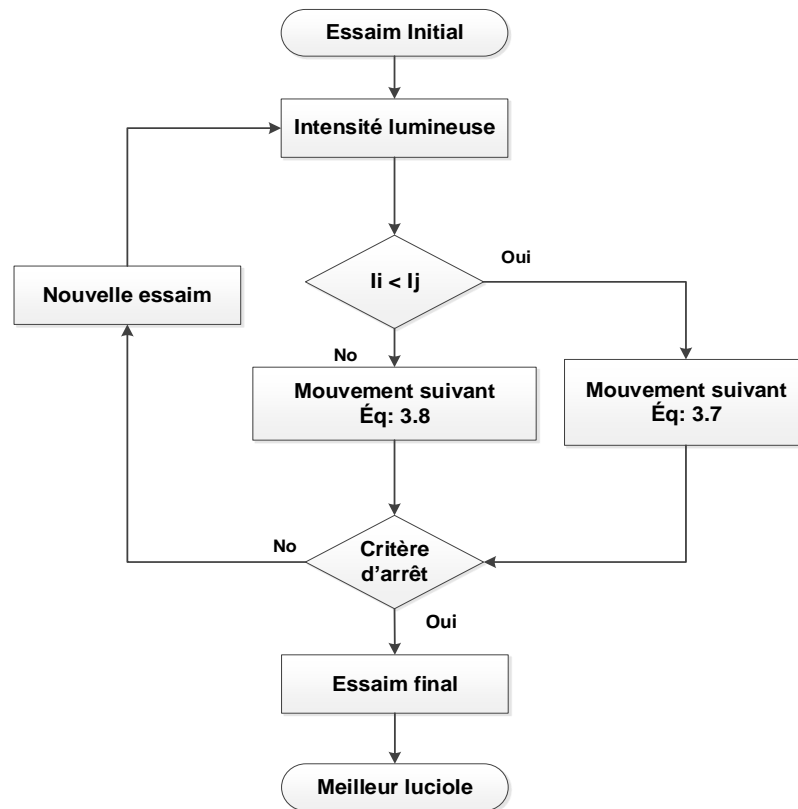


Figure 3.5 Algorithme hybride proposée ALASC2

3.6.4. Résultats d'optimisation avec les fonctions de référence

Les algorithmes hybrides sont testés sous les 23 fonctions de référence citée auparavant pour étudier ses performances par rapport à l'algorithme de luciole, les résultats sont présentés dans le tableau ci-dessous

Tableau 3.5 Résultats d'optimisation

Fonction	AL		ALASC1		ALASC2	
	μ	σ	μ	σ	μ	σ
F1	1.4E+04	3.8E+03	2.4E-302	00E00	2.5E-275	00E00
F2	7.4E+02	1.5E+03	1.5E-313	7.2E-312	1.1E-36	3.2E-36
F3	1.5E+04	4.5E+03	2.7E-146	8.6E-146	3.5E-46	1.1E-45
F4	6.4E+01	7.7E00	7.9E-142	3.9E-141	9.7E-157	4.8E-156
F5	3.1E+07	1.5E+07	7.0E-04	9.0E-04	3.0E-04	4.1 E-04
F6	1.4E+04	3.7E+03	1.1E-08	1.6E-07	1.3E-06	2.2E-06
F7	5.1E00	2.6E00	1.4E-05	1.5E-05	1.8E-05	1.8E-05
F8	-1.3 E+03	2.6 E+02	-1.9E+03	5.0E-04	00E00	00E00
F9	1.2E+02	9.2E+01	00E00	00E00	00E00	00E00
F10	1.9E+01	5.0E-01	8.0E-16	0.0E00	8.8E-16	00E00
F11	1.3E+02	3.4E+01	00E00	00E00	00E00	00E00
F12	5.7E+07	3.4E+07	1.7E-08	3.9E-07	1.0E-06	1.8E-06
F13	1.1E+08	6.9E+07	1.2E-08	3.0E-07	9.8E-06	2.6E-05
F14	1.1E+02	1.2E+02	1.2E01	1.0E-04	1.0E-01	2.7E-11
F15	1.2E-01	9.3 E-02	3.1 E-05	9.8 E-04	3.5E-04	3.7E-05
F16	2.2E-01	5.8E-01	-1.1E00	1.4E-05	-1.0 E00	2.5E-05
F17	1.1 E00	7.7E-01	3.9 E-02	2.2 E-04	3.9 E-01	1.3 E-04
F18	3.6 E+01	4.2 E+01	3.7 E00	1.7 E-04	3.0 E+00	8.4E-06
F19	-3.3 E+00	0.3 E-01	-3.8 E+00	1.1 E-07	-3.8 E+00	5.4 E-02
F20	-1.5 E+00	5.5 E-01	-3.3 E+00	1.5 E-01	-3.1 E+00	1.2 E-01
F21	-6.4 E-01	3.2 E-01	-1.0 E+01	4.8 E-02	-1.0 E+01	2.3E-05
F22	-7.9 E-01	3.1 E-01	-1.0 E+01	6.7 E-02	-1.0 E+01	3.5E-05
F23	-9.1 E-01	3.0 E-01	-1.0 E+01	7.3 E-02	-1.0 E+01	4.8E-05

3.7. Hybridation de l'algorithme de luciole hybride avec l'OLG et l'ASC:

Dans cette section, nous incorporons deux métaheuristiques dans l'algorithme de luciole en deux manières différentes

3.7.1. Algorithme hybride proposé, proposition 05

Le type d'hybridation, dans ce cas, est un relais de bas niveau. La variante hybride proposée est appelée HALOLGASC1 (HFAGWOSCA1)

Dans l'algorithme proposé, nous combinons l'optimiseur de loup gris et l'algorithme d'optimisation du sinus cosinus dans l'algorithme de luciole standard en suivant les règles ci-dessous.

1. La luciole « i » se déplace vers la luciole « j » qui est plus lumineuse en appliquant la formule de mise à jour de l'algorithme de luciole standard.
2. Si ce n'est pas le cas, il se déplace vers un emplacement déterminé par l'optimiseur de loups gris, tel qu'indiqué dans l'équation suivante :

$$x_i^{t+1} = \begin{cases} x_i^t + r_1 * \sin(r_2) * |r_3 * x_{gwo} - x_i^t|, r_4 > 0.5 \\ x_i^t + r_1 * \cos(r_2) * |r_3 * x_{gwo} - x_i^t|, r_4 \leq 0.5 \end{cases} \quad (3.9)$$

La location x_{gwo} est déterminée par l'optimiseur de loup gris. L'algorithme hybride proposé peut être résumé dans l'organigramme suivant :

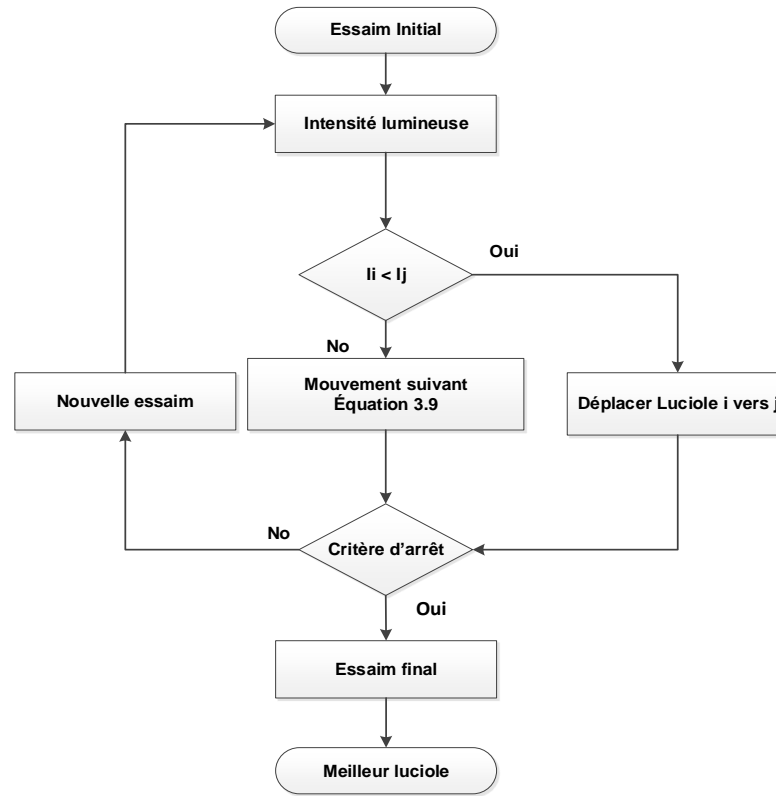


Figure 3.6 Algorithme hybride proposée ALOLGASC1

3.7.2. Algorithme hybride proposé, proposition 06

Dans l'algorithme proposé, nous combinons l'optimiseur du loup gris et l'algorithme d'optimisation sinus cosinus dans l'algorithme de luciole standard. Le type de l'hybridation dans ce cas est le **relais de bas niveau**. La variante hybride proposée est appelée HALOLGASC2 (**HFAGWOSCA2**). Tout d'abord, nous proposons de modifier l'équation standard du mouvement de luciole vers les plus brillants en supprimant le troisième terme de l'équation 2.7 qui caractérise la randomisation et en le remplaçant par les équations de l'algorithme d'optimisation du sinus cosinus comme le montre l'équation suivante :

$$x_i^{t+1} = \begin{cases} x_i^t + \beta * (x_j^t - x_i^t) + r_1 * \sin(r_2) * |r_3 * x_{gwo} - x_i^t|, r_4 > 0.5 \\ x_i^t + \beta * (x_j^t - x_i^t) + r_1 * \cos(r_2) * |r_3 * x_{gwo} - x_i^t|, r_4 \leq 0.5 \end{cases} \quad (3.10)$$

Ensuite, nous remplaçons le mouvement aléatoire des lucioles s'il n'y a pas d'autre solution meilleure que celle-là par le mouvement inspiré de l'algorithme d'optimisation sinus-cosinus.

$$x_i^{t+1} = \begin{cases} x_i^t + r_1 * \sin(r_2) * |r_3 * x_{gwo} - x_i^t|, r_4 > 0.5 \\ x_i^t + r_1 * \cos(r_2) * |r_3 * x_{gwo} - x_i^t|, r_4 \leq 0.5 \end{cases} \quad (3.11)$$

Ce nouvel algorithme est présenté par l'organigramme de la figure 3.7.

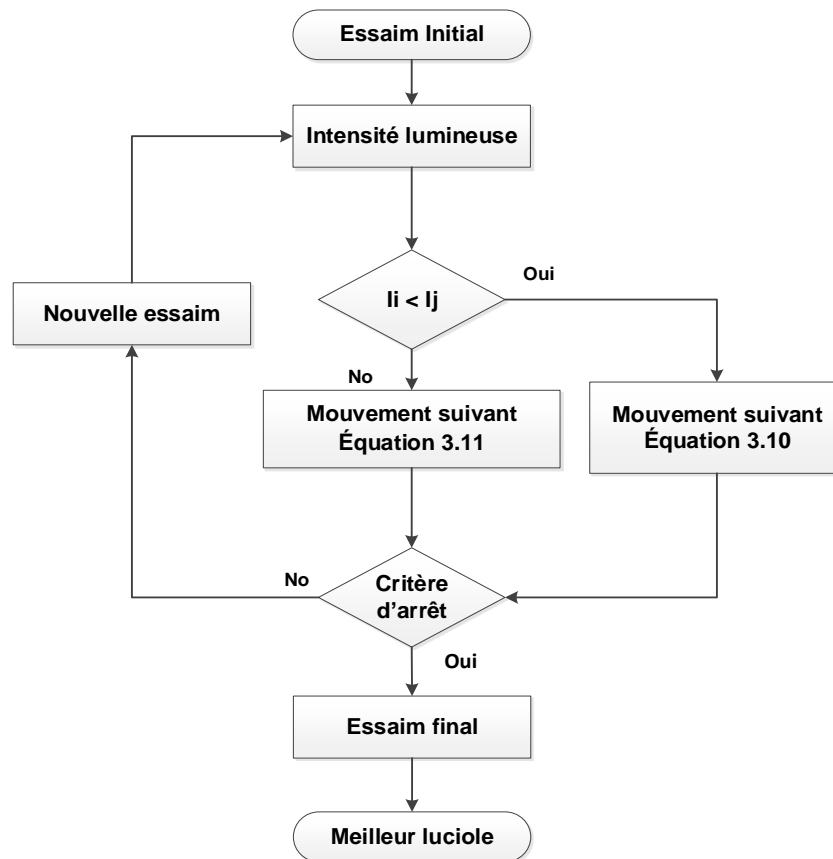


Figure 3.7 Algorithme hybride proposée ALOLGASC2

3.7.3. Résultats d'optimisation avec les fonctions de référence

Les algorithmes hybrides sont testés sous les 23 fonctions de référence citée auparavant pour étudier ses performances par rapport à l'algorithme de luciole, les résultats sont présentés dans le tableau 3.5.

Tableau 3.6 Résultats d'optimisation

Fonction	AL		ALOLGASC1		ALOLGASC2	
	μ	σ	μ	σ	μ	σ
F1	1.4E+04	3.8E+03	1.6E-243	2.0E-243	5.3E-244	7.9E-244
F2	7.4E+02	1.5E+03	1.6E-24	1.5E-24	1.4E-24	1.1E-24
F3	1.5E+04	4.5E+03	1.0E-44	1.3E-44	1.9E-44	4.8E-44
F4	6.4E+01	7.7E00	3.4E-24	2.8E-24	3.4E-24	3.1E-24
F5	3.1E+07	1.5E+07	5.8 E-04	8.2 E-04	3.1 E-04	4.3 E-04
F6	1.4E+04	3.7E+03	8.9E-07	1.1E-06	8.8E-07	1.0E-06
F7	5.1E00	2.6E00	2.7E-05	1.8E-05	2.8E-05	2.2E-05
F8	-1.3 E+03	2.6 E+02	-9.4 E+03	6.6 E+03	-1.3E+136	6.2E+136
F9	1.2E+02	9.2E+01	00E00	00E00	00E00	00E00
F10	1.9E+01	5.0E-01	1.6E+01	7.3E00	8.8E-16	00E00
F11	1.3E+02	3.4E+01	00E00	00E00	00E00	00E00
F12	5.7E+07	3.4E+07	1.2E-06	1.3E-06	1.2E-06	1.1E-06
F13	1.1E+08	6.9E+07	5.9E-06	6.6E-06	4.6E-06	6.9E-06
F14	1.1E+02	1.2E+02	1.0E-01	1.4E-08	1.0E-01	2.2E-11
F15	1.2E-01	9.3 E-02	1.9 E-03	1.4 E-03	3.6 E-04	2.3 E-04
F16	2.2E-01	5.8E-01	-1.0E00	2.2E00	-1.0E00	4.1E-06
F17	1.1 E00	7.7E-01	3.9 E-01	7.2 E-04	3.9 E-01	7.5E-05
F18	3.6 E+01	4.2 E+01	3.0 E+00	1.2 E-04	3.0 E+00	5.7E-05
F19	-3.3 E+00	0.3 E-01	-3.8 E+00	1.0 E-02	-3.8 E+00	5.1 E-02
F20	-1.5 E+00	5.5 E-01	-3.2 E+00	1.0 E-01	-3.2 E+00	6.5 E-02
F21	-6.4 E-01	3.2 E-01	-1.0. E+01	3.9 E-02	-1.0 E+01	4.5E-06
F22	-7.9 E-01	3.1 E-01	-1.0 E+01	6.4 E-02	-1.0 E+01	2.2E-05
F23	-9.1 E-01	3.0 E-01	-1.0 E+01	8.1 E-02	-1.0 E+01	1.3E-05

3.8. Discussions des résultats

On regroupe les résultats précédents dans les mêmes tableaux pour comparer entre les algorithmes proposés.

Pour approfondir les tests sur les algorithmes proposés, nous définissons un test supplémentaire.

- Dans ce test, les expériences sont effectuées avec les fonctions (F_{01} , F_{04} , F_{07} , F_{09} , F_{10} , F_{11} , F_{12}) de test avec une dimension égale à **150**, **200**, **500**.

Les algorithmes ont été initialisés de la même manière de sorte que la comparaison soit la plus pertinente possible. La première génération est distribuée uniformément au sein des intervalles de définition de chaque fonction de test (tableau 3.2).

Pour plus de clarté, les solutions obtenues, qui sont de meilleure qualité, ont été mises en gras.

- **Test 01 : D = 30**

Tableau 3.7 Résultats pour les fonctions Uni-modales (F01 à F07)

Fonction	AL		ALA		ALOLG		ALASC1	
	μ	σ	μ	σ	μ	σ	μ	σ
F1	1.4E+04	3.8E+03	1.9E+03	7.7E+01	1.2E-250	00E00	2.4E-302	00E00
F2	7.4E+02	1.5E+03	1.1E+01	4.5E+00	1.8E-308	00E00	1.5E-313	7.2E-312
F3	1.5E+04	4.5E+03	4.7E+03	2.2E+03	1.7E-131	8.8E-131	2.7E-146	8.6E-146
F4	6.4E+01	7.7E00	2.3E+01	6.1E00	3.E-145	1.5E-144	7.9E-142	3.9E-141
F5	3.1E+07	1.5E+07	3.2E+05	2.6E+05	8.9E00	4.6E-02	7.0E-04	9.0E-04
F6	1.4E+04	3.7E+03	1.5E+03	4.5E02	0.95103	0.35929	1.1E-08	1.6E-07
F7	5.1E00	2.6E00	2.6E-01	2.1E-01	1.8E-05	2.1E-05	1.4E-05	1.5E-05

Tableau 3.8 Résultats pour les fonctions Uni-modales (F01 à F07)

Fonction	ALASC2		ALOLGASC1		ALOLGASC2	
	μ	σ	μ	σ	μ	σ
F1	2.5E-275	00E00	1.6E-243	2.0E-243	5.3E-244	7.9E-244
F2	1.1E-36	3.2E-36	1.6E-24	1.5E-24	1.4E-24	1.1E-24
F3	3.5E-46	1.1E-45	1.0E-44	1.3E-44	1.9E-44	4.8E-44
F4	9.7E-157	4.8E-156	3.4E-24	2.8E-24	3.4E-24	3.1E-24
F5	3.0E-04	4.1 E-04	5.8 E-04	8.2 E-04	3.1 E-04	4.3 E-04
F6	1.3E-06	2.2E-06	8.9E-07	1.1E-06	8.8E-07	1.0E-06
F7	1.8E-05	1.8E-05	2.7E-05	1.8E-05	2.8E-05	2.2E-05

Tableau 3.9 Résultats pour les fonctions multimodales (F08 à F23)

Fonction	AL		ALA		ALOLG		ALASC1	
	μ	σ	μ	σ	μ	σ	μ	σ
F8	-1.3 E+03	2.6 E+02	-9.4 E+03	6.6 E+03	-1.3E+136	6.2E+136	-1.9E+03	5.0E-04
F9	1.2E+02	9.2E+01	6.0E+01	1.3E+01	00E00	00E00	00E00	00E00
F10	1.9E+01	5.0E-01	1.2E+01	1.8E+01	8.8E-16	00E00	8.0E-16	0.0E00
F11	1.3E+02	3.4E+01	5.4E00	00E00	00E00	00E00	00E00	00E00
F12	5.7E+07	3.4E+07	4.6E+04	1.2E+05	2.8E-01	2.1E-01	1.7E-08	3.9E-07
F13	1.1E+08	6.9E+07	2.2E+05	4.3E+05	5.1E-01	1.9E-01	1.2E-08	3.0E-07
F14	1.1E+02	1.2E+02	3.2E00	3.4E00	8.2E00	4.6E00	1.2E01	1.0E-04
F15	1.2E-01	9.3 E-02	1.9 E-02	1.8 E-02	9.9 E-03	1.1 E-02	3.1 E-05	9.8 E-04
F16	2.2E-01	5.8E-01	-1.0E00	3.0E-02	-1.0E00	5.1E-02	-1.1E00	1.4E-05
F17	1.1 E00	7.7E-01	5.4 E-01	3.1E-01	4.1E-01	4.1 E-02	3.9 E-02	2.2 E-04
F18	3.6 E+01	4.2 E+01	1.3 E+01	1.2 E+01	9.7 E+00	1.0 E+01	3.7 E00	1.7 E-04
F19	-3.3 E+00	0.3 E-01	-3.6 E+00	1.9 E+01	-3.7 E+01	1.3 E+01	-3.8 E+00	1.1 E-07
F20	-1.5 E+00	5.5 E-01	-2.2 E+00	4.8 E-01	-2.6 E+00	3.8 E-01	-3.3 E+00	1.5 E-01
F21	-6.4 E-01	3.2 E-01	-5.3 E+00	2.4 E+00	-4.9 E+00	1.3 E+00	-1.0 E+01	4.8 E-02
F22	-7.9 E-01	3.1 E-01	-4.3 E+00	1.9 E+00	-4.4 E+00	1.9 E+00	-1.0 E+01	6.7 E-02
F23	-9.1 E-01	3.0 E-01	-3.9 E+00	2.3 E+00	-5.0 E+00	2.2 E+00	-1.0 E+01	7.3 E-02

Tableau 3.10 Résultats pour les fonctions multimodales (F08 à F23)

Fonction	ALASC2		ALOLGASC1		ALAOLGSC2	
	μ	σ	μ	σ	μ	σ
F8	-1.7E+03	3.0E-04	-9.4 E+03	6.6 E+03	-1.3E+136	6.2E+136
F9	00E00	00E00	00E00	00E00	00E00	00E00
F10	8.8E-16	00E00	1.6E+01	7.3E00	8.8E-16	00E00
F11	00E00	00E00	00E00	00E00	00E00	00E00
F12	1.0E-06	1.8E-06	1.2E-06	1.3E-06	1.2E-06	1.1E-06
F13	9.8E-06	2.6E-05	5.9E-06	6.6E-06	4.6E-06	6.9E-06
F14	1.0E-01	2.7E-11	1.0E-01	1.4E-08	1.0E-01	2.2E-11
F15	3.5E-04	3.7E-05	1.9 E-03	1.4 E-03	3.6 E-04	2.3 E-04
F16	-1.0 E00	2.5E-05	-1.0E00	2.2E00	-1.0E00	4.1E-06
F17	3.9 E-01	1.3 E-04	3.9 E-01	7.2 E-04	3.9 E-01	7.5E-05
F18	3.0 E+00	8.4E-06	3.0 E+00	1.2 E-04	3.0 E+00	5.7E-05
F19	-3.8 E+00	5.4 E-02	-3.8 E+00	1.0 E-02	-3.8 E+00	5.1 E-02
F20	-3.1 E+00	1.2 E-01	-3.2 E+00	1.0 E-01	-3.2 E+00	6.5 E-02
F21	-1.0 E+01	2.3E-05	-1.0. E+01	3.9 E-02	-1.0 E+01	4.5E-06
F22	-1.0 E+01	3.5E-05	-1.0 E+01	6.4 E-02	-1.0 E+01	2.2E-05
F23	-1.0 E+01	4.8E-05	-1.0 E+01	8.1 E-02	-1.0 E+01	1.3E-05

d'après les résultats obtenus, nous remarquons que les algorithmes développés dépassent l'algorithme de luciole standard pour toutes fonctions. Ainsi, nous constatons que l'algorithme de luciole hybride avec l'algorithme sinus cosinus type 1 dépasse toutes les autres techniques pour la majorité des fonctions tests :

- Les fonctions unimodales, sont convenables pour l'analyse comparative d'exploitation (résultats présentés dans les tableaux 3.7 et 3.8), l'algorithme **HALASC1** dépasse les autres techniques pour toutes les fonctions F_{01} , F_{02} , F_{03} , F_{05} , F_{06} , F_{07} , en exception la fonction F_{04} . Par conséquent, ces résultats montrent la performance supérieure de l'algorithme **HALASC1** en termes d'exploitation. Il est à noter que l'algorithme de luciole améliorée présente aussi une performance élevée face à l'algorithme de luciole standard en termes d'exploitation. On remarque aussi que les autres algorithmes développés dépassent l'algorithme de luciole standard pour toutes les fonctions.
- Contrairement aux fonctions unimodales, les fonctions multimodales (F_{08} à F_{23}) ont beaucoup d'optima locaux. Cela permet de comparer et à analyser la capacité d'exploration d'un algorithme. Selon les résultats des tableaux

3.9 et 3.10, l'algorithme **HALASC1** est fourni des résultats très compétitifs sur les fonctions F_{09} , F_{10} , F_{11} , F_{12} , F_{13} , F_{15} , F_{16} , F_{17} , F_{18} , F_{19} , F_{20} , F_{21} , F_{22} et F_{23} . il dépasse tous les autres algorithmes. Donc, l'algorithme **HALASC1** est le plus adapté à l'exploration de l'espace de recherche. Il atteint la valeur optimale théorique avec les fonctions F_{09} , F_{11} . L'algorithme **HALASC1** donne également des résultats acceptables pour la fonction F_8 . Les autres algorithmes hybrides donnent également des résultats acceptables pour toutes les fonctions multimodales sauf pour F_8 .

Les résultats correspondant aux fonctions tests montrent et prouvent les capacités d'exploitation et d'exploration de l'algorithme **HALASC1**. La figure 3.7 illustre les courbes de convergence obtenues avec les fonctions F_{01} , F_{04} , F_{09} , F_{10} , F_{11} et F_{19} . Il est clair que l'algorithme **HALASC1** converge plus rapidement que les autres algorithmes. Nous pouvons constater que **HALASC1** donne un écart type plus petit que les autres algorithmes, ce qui confirme sa stabilité et sa robustesse.

La résolution des problèmes d'optimisation de très haute dimension à l'aide des techniques métaheuristiques, a suscité un intérêt croissant. Il est connu que l'espace de recherche d'un problème augmente de façon exponentielle avec la taille du problème, ce qui entraîne une dégradation des performances de nombreuses métaheuristiques. Donc, pour examiner les performances de nos algorithmes développés sur les fonctions tests de très hautes dimensions. On a utilisé les fonctions F_{01} , F_{04} , F_{07} , F_{10} , F_{11} et F_{12} avec les dimensions $D=150$, $D=300$ et $D=500$. Les résultats obtenus sont comparés en termes de la meilleure valeur optimale moyenne dans les tables 3.11, 3.12 et 3.13 respectivement.

Tableau 3.11 Résultats pour une dimension = 150

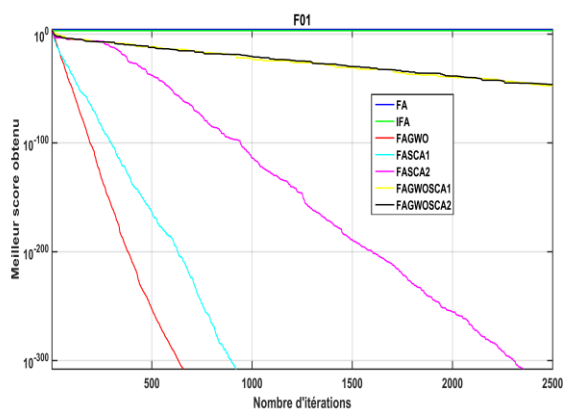
Fonction	AL	ALA	ALOLG	ALASC1	ALASC2	ALOLGASC1	ALOLGASC2
F01	3.0E+05	1.2 E+05	00E00	00E00	00E00	3.0E-48	3.0E-47
F04	9.6E01	6.9E01	5.3E-150	00E00	1.7E-180	7.0E-25	3.1E-25
F07	3.9E04	4.0E02	8.6E-06	4.5E-07	3.4E-05	1.1E-04	1.7E-05
F10	2.1E01	1.9E01	8.8E-16	8.8E-16	8.8E-16	2.0E01	8.8E-16
F11	3.3E03	1.9E03	00E00	00E00	00E00	00E00	00E00
F12	3.5E09	2.6E08	3.4E-01	6.6E-06	1.1E-07	6.8E-08	1.1E-07

Tableau 3.12 Résultats pour une dimension D= 300

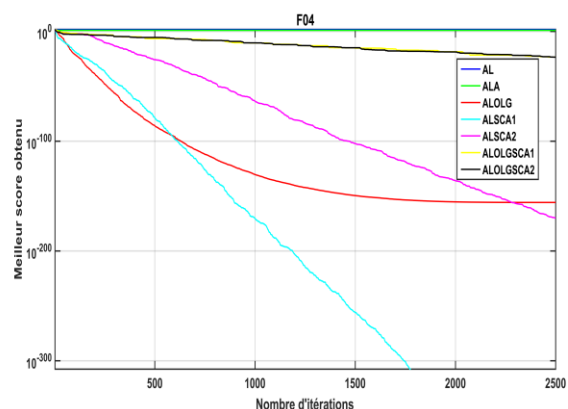
Fonction	AL	ALA	ALOLG	ALASC1	ALASC2	ALOLGASC1	ALOLGASC2
F01	8.9E+05	2.7 E+05	00E00	00E00	00E00	1.8E-45	6.3E- 46
F04	9.9E01	7.4E01	4.2E-152	00E00	5.5E-177	2.3E-24	1.5E-24
F07	2.0E05	2.5E03	4.0E-05	3.3E-06	3.1E-05	1.5E-05	2.6E-05
F10	2.2E01	1.9E01	8.8E-16	8.8E-16	8.8E-16	2.1E01	8.8E-16
F11	8.0E03	2.4E03	00E00	00E00	00E00	00E00	00E00
F12	1.0E10	5.2E08	2.7E-01	3.8E-08	1.5E-08	3.3E-07	1.4E-07

Tableau 3.13 Résultats pour une dimension D= 500

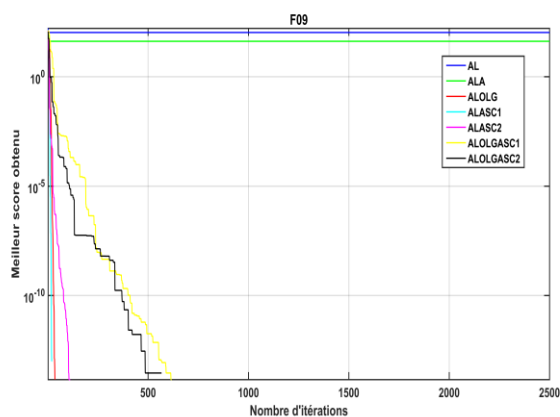
Fonction	AL	ALA	ALOLG	ALASC1	ALASC2	ALOLGASC1	ALOLGASC2
F01	1.5E+06	4.9E+05	00E00	00E00	00E00	1.6E-45	4.6E- 48
F04	9.9E01	7.2E01	3.0E-152	00E00	1.0E-178	3.4E-24	2.6E-25
F07	5.2E04	6.7E03	1.7E-06	7.1E-06	2.2E-05	1.6E-05	8.7E-06
F10	2.1E01	1.9E01	8.8E-16	8.8E-16	8.8E-16	1.7E01	8.8E-16
F11	1.3E04	4.4E03	00E00	00E00	00E00	00E00	00E00
F12	1.6E10	9.4E08	2.7E-01	2.1E-08	2.2E-07	1.3E-06	1.1E-07



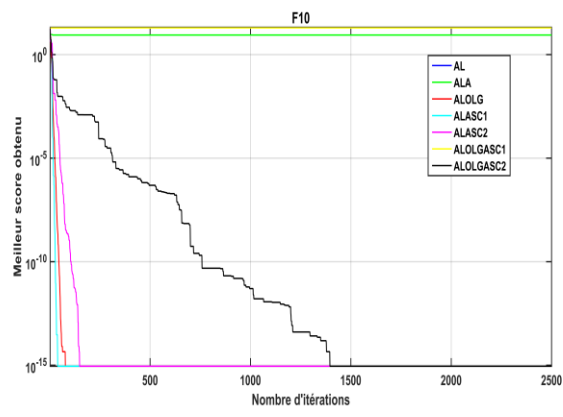
(a)



(b)



(c)



(d)

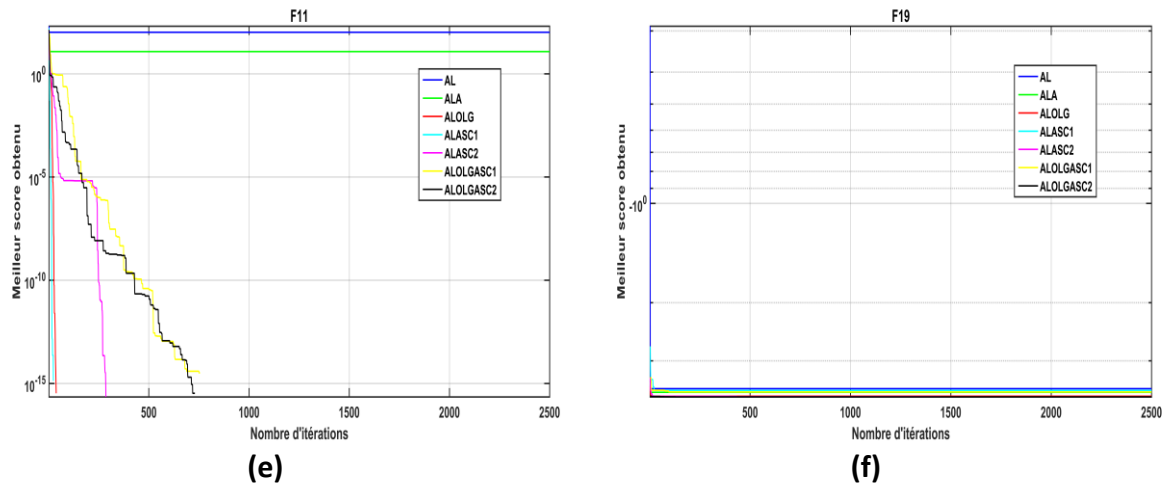


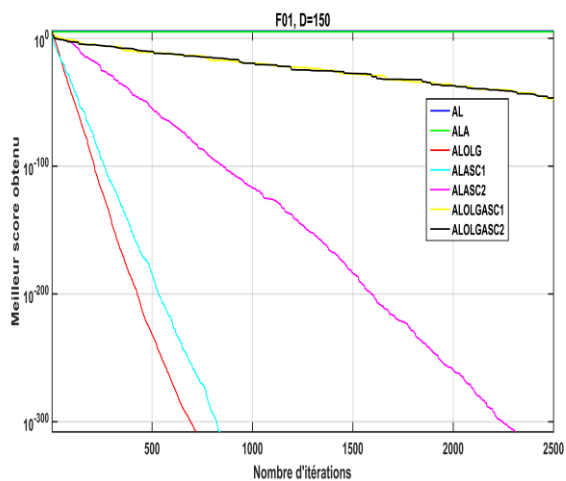
Figure 3.8 Courbes de convergence obtenues avec les fonctions F_{01} , F_{04} , F_{09} , F_{10} , F_{11} , F_{19}

En comparant ces résultats, nous observons que chaque fois que la dimension augmente, les résultats de tous les algorithmes se détériorent.

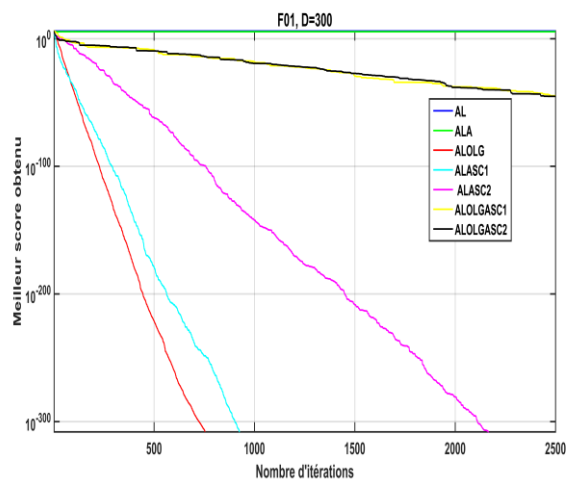
Cependant, contrairement aux autres algorithmes, la détérioration des performances de **HALASC1** est insignifiante. Par exemple, avec F_{10} , les résultats sont d'ordre de 10^{-16} pour les dimensions 150, 300 et 500 respectivement. Il est évident que la dégradation des résultats de **HALASC1** est très faible. On peut aussi remarquer qu'avec la fonction F_{11} , l'algorithme **HALASC1** a obtenu la valeur optimale zéro quelle que soit la dimension du problème. De plus, l'algorithme **HALASC1** maintient une performance supérieure à tous les autres algorithmes développés, ce qui prouve encore une fois l'efficacité de l'algorithme **HALASC1** dans la résolution des problèmes de très hautes dimensions. On remarque ainsi que tous les algorithmes hybrides prouvent leur performance par rapport les algorithmes de luciole standard et améliorée.

La figure 3.8 illustre les courbes de convergence des algorithmes développées obtenues avec les fonctions F_{01} , F_{07} et F_{11} , où les dimensions sont 150, 300 et 500. Il est évident que l'algorithme **HALASC1** a une convergence plus rapide que tous les autres algorithmes développés.

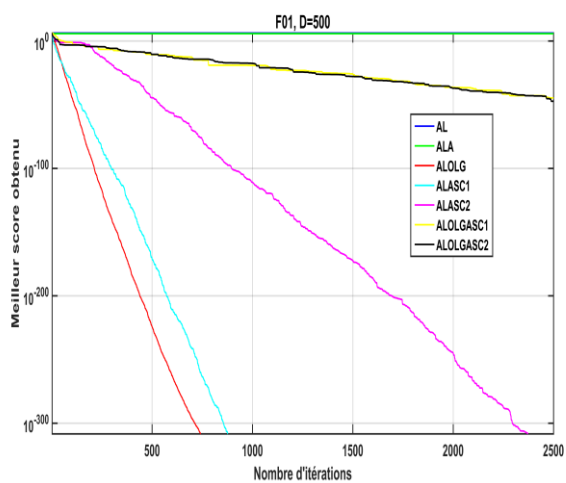
En résumé, les résultats vérifient les performances des algorithmes de lucioles développées dans la résolution de diverses fonctions de référence par rapport à l'algorithme de luciole standard. Pour approfondir l'analyse de la performance des algorithmes proposés, un problème d'identification paramétrique sera discuté dans le chapitre suivant.



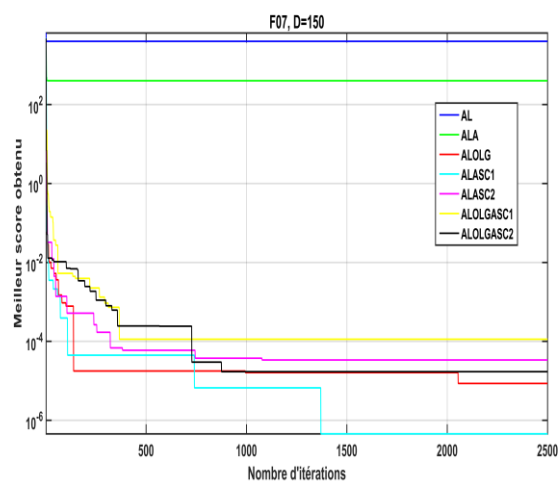
(a)



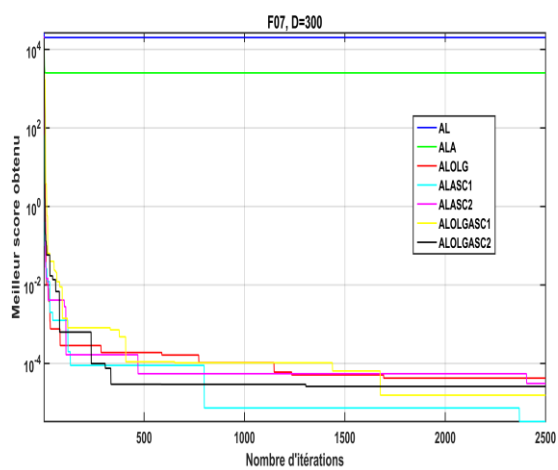
(b)



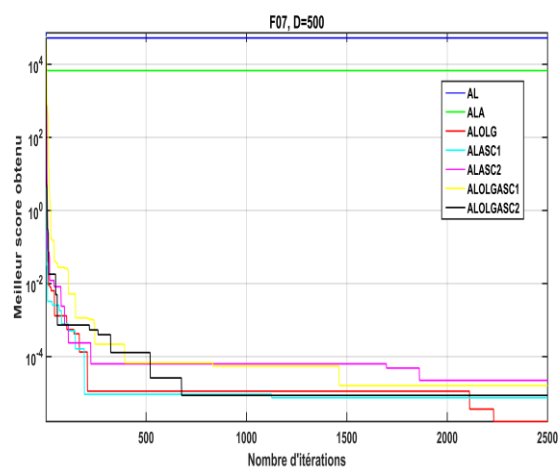
(c)



(d)



(e)



(f)

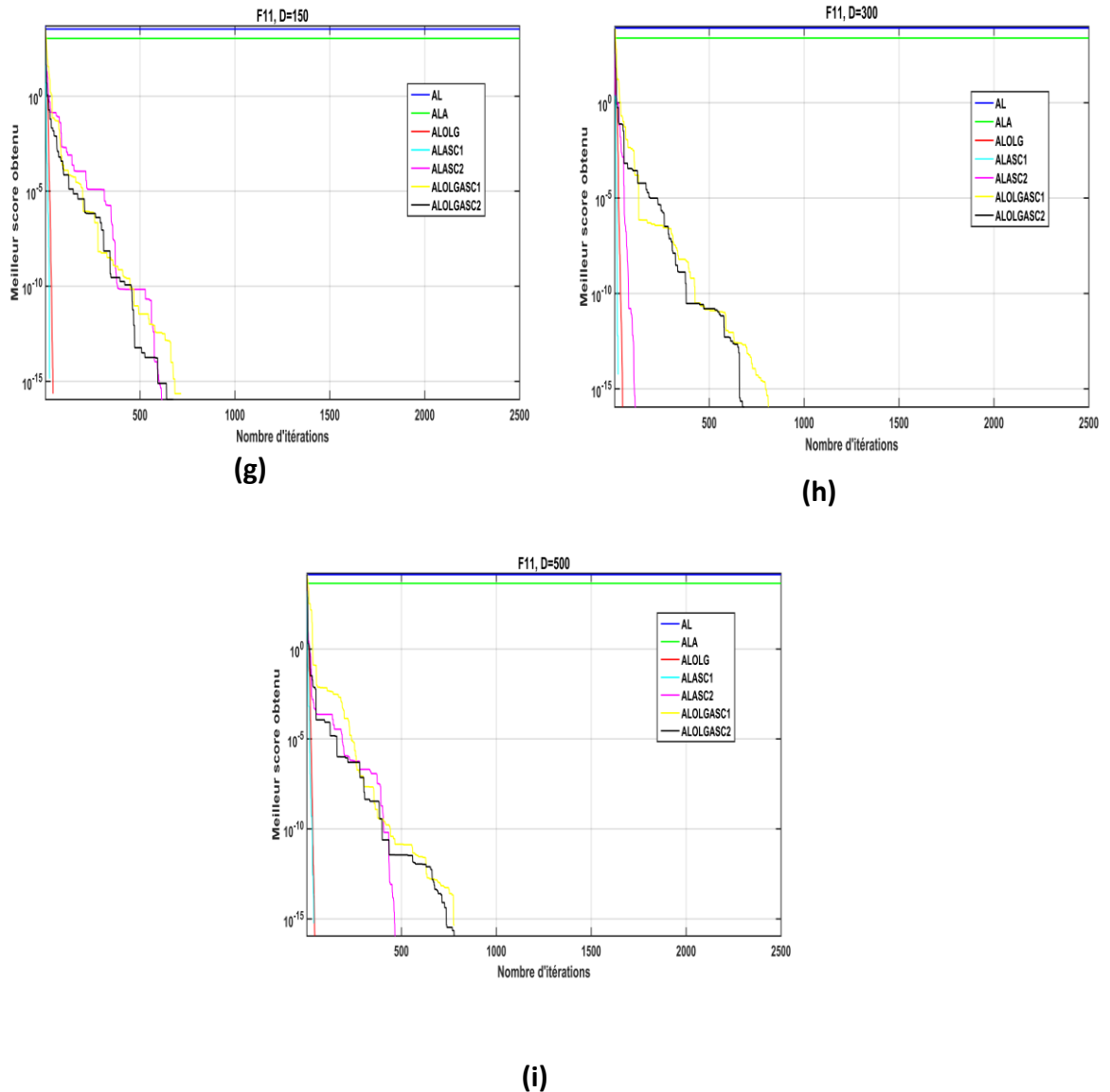


Figure 3.9 Courbes de convergence sur les dimensions 150, 300 et 500 obtenues avec F_{01} , F_{07} et F_{11}

3.9. Conclusion

Nous avons proposé dans ce chapitre, six améliorations sur l'algorithme de luciole standard :

- une orientation de mouvement des lucioles vers le centroïde de l'essaim dans le cas où le mouvement vers la luciole plus le plus brillant est en échec.
- Une hybridation de l'algorithme de luciole avec l'optimiseur de loup gris.
- Deux hybridations de l'algorithme de luciole avec l'algorithme sinus cosinus type1 et type2.

- Deux hybridations de l'algorithme de luciole avec l'optimiseur de loup gris et l'algorithme de sinus cosinus type1 et type2.

Nous avons testé nos algorithmes par des fonctions de référence. Les résultats obtenus montrent la supériorité des algorithmes proposés par rapport à l'algorithme de luciole standard.

CHAPITRE 4

APPLICATION A L'IDENTIFICATION PARAMETRIQUE DE LA MACHINE ASYNCHRONE

4.1. Introduction

Dans ce chapitre, la détermination des paramètres de la machine asynchrone repose sur le modèle de référence. Le critère de comparaison est optimisé par les algorithmes de luciole proposés.

Ce type d'application est semblable à ce qui est arrivé au cours des dernières décennies avec d'autres algorithmes évolutionnaires tels que l'algorithme génétique (AG) [17], [78], l'optimisation de l'essaim de particules (PSO) [18], l'algorithme de colonie artificielle d'abeilles (ABC) [79], l'alimentation bactérienne (BF) [80], algorithme de saut de grenouille mélangé [34]. La plupart des techniques n'ont estimé que les paramètres électriques [29], [49], [81]–[86]. Les paramètres mécaniques sont déterminés séparément par la méthode de ralentissement. Contrairement à la méthode adoptée [17], [78] qui permet d'estimer simultanément les paramètres électriques et mécaniques, en n'utilisant que le courant de démarrage et la tension de phase correspondante.

Nous confirmons les programmes que nous avons développés en utilisant des données simulées et des données expérimentales issues de deux moteurs différents.

4.2. Identification à l'aide de modèle de référence

Le but de cette méthode est la détermination des paramètres électriques et mécaniques simultanément en utilisant le courant mesuré au démarrage d'une phase et la simple tension correspondante.

La méthode est basée sur la minimisation de l'erreur quadratique « S » calculée à la sortie du modèle (équation 4.1) entre la sortie de la machine réelle et la sortie du modèle de la machine

$$S = \sum_{i=1}^n (I_{am} - I_{ac})^2 \quad (4.1)$$

Où :

- n : la taille de l'échantillon.
- I_{am} : le courant mesuré.
- I_{ac} : le courant calculé.

La méthode d'identification peut être résumée comme suit :

1. L'entrée du système réel et le modèle mathématique correspondant sont alimentés par la même tension sinusoïdale.
2. La sortie du moteur réel est mesurée et la sortie du modèle de machines avec une solution candidate sélectionnée par l'algorithme d'optimisation, est calculée par la résolution numérique du système d'équations différentielles en utilisant la méthode Runge-Kutta du quatrième ordre.
3. L'erreur quadratique de sortie entre le courant mesuré au démarrage et le courant calculé sera calculée.
4. Le critère d'arrêt est mis en œuvre :
 - 4.1. S'il est vérifié, la meilleure solution est atteinte.
 - 4.2. Sinon, répéter l'étape (2) avec une autre solution sélectionnée par la méthode d'optimisation choisie.
5. Fin de la méthode d'identification.

4.3. Identification paramétrique de la machine asynchrone

Les données simulées permettent de vérifier l'adaptation des algorithmes développés ainsi que les différents résultats obtenus. Les données expérimentales relevées sur deux moteurs différents confirment les résultats obtenus.

4.3.1. Identification l'aide des données simulées

À l'aide de données simulées nous proposons de :

- valider les algorithmes développés en application réelle.
- 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 quadratique permet une approximation convenable de la fonction coût à minimiser, quelques raisons pour cela :

- ✓ une fonction quadratique est parmi les plus simples des fonctions avec minimum bien déterminé.
- ✓ Une fonction générale développée autour d'un minimum local $\theta(k)$ est bien approché par une fonction quadratique (série de Taylor d'ordre 2).
- ✓ Même éloigné du minimum, il semble préférable d'utiliser une approximation du type quadratique. Cette approximation est plus efficace que l'approximation linéaire. La raison est liée au fait que le développement de la fonction cout en série de Taylor au second ordre permet de conserver une bonne précision sur un voisinage plus étendu qu'avec l'approximation au premier ordre.
- ✓ Les méthodes basées sur une fonction quadratique peuvent être invariantes lorsqu'on fait une transformation linéaire de variables (mise à l'échelle).

Les données simulées sont obtenues par résolution numérique du système non-linéaire (équation **1.19**) de la MAS par la méthode de Runge-Kutta d'ordre 4 en utilisant les paramètres d'une machine connue alimentée par une tension sinusoïdale. Ces données sont illustrées par la figure 4.2 qui représente le courant de démarrage et la tension simple correspondante.

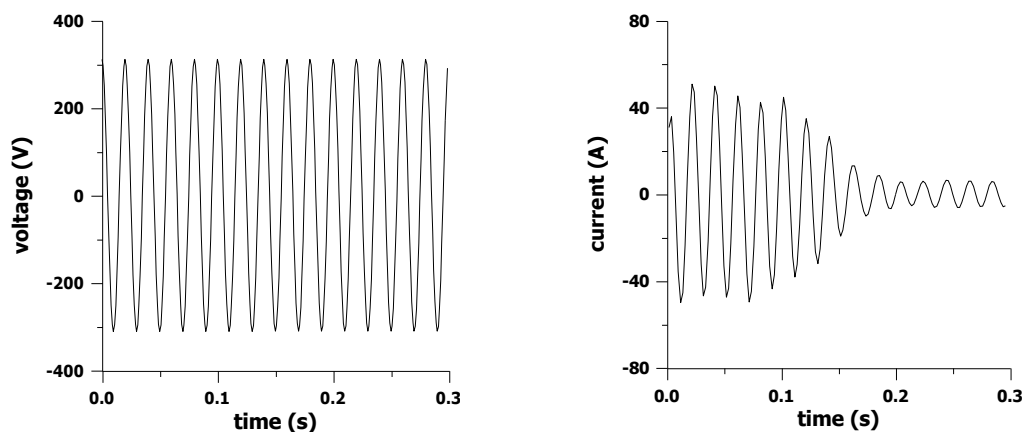


Figure 4.1 (a) Tension d'alimentation (b) Courant statorique calculé correspondant

Pour identifier les paramètres de la machine connue, nous utilisons comme algorithme d'optimisation:

- algorithme de luciole standard et trois variantes [87].

- Algorithme de luciole améliorée [88].
- Algorithme de luciole hybride avec algorithme sinus cosinus 1 et 2.
- Algorithme de luciole hybride avec optimiseur de loup gris.
- Algorithme de luciole hybride avec optimiseur de loup gris et algorithme de sinus cosinus 1 et 2.

La configuration de tous algorithmes utilisés dans cette section est fixée comme suit :

- la taille d'essaim est : 100 lucioles.
- Les différents paramètres sont :
 - coefficient d'attraction : 2 ;
 - coefficient d'absorption : 2.

Les résultats d'identification sont résumés dans les tableaux 4.1 et 4.2

Tableau 4.1 Identification l'aide des données simulées

Paramètres		ALS	ALV1	ALV2	ALV3	ALA
σ	0.09	0.8999	0.09	0.09	0.09	0.09
T_r (ms)	123	123	123	123	123	123
L_s (mH)	159	159	159	159	159	159
T_s (ms)	54	54	54	54	54	54
J (Kg.m ²)	0.038	0.038	0.038	0.038	0.038	0.038
fr (Nm.s/Rd)	0.001	0.00099	0.001	0.001	0.001	0.001

Tableau 4.2 Identification l'aide des données simulées

Paramètres	ALOLG	ALASC1	ALASC2	ALOLGASC1	ALOLGASC2
σ	0.8999	0.09	0.09	0.09	0.09
T_r (ms)	123	123	123	123	123
L_s (mH)	159	159	159	159	159
T_s (ms)	54	54	54	54	54
J (Kg.m ²)	0.038	0.038	0.038	0.038	0.038
fr (Nm.s/Rd)	0.00099	0.001	0.001	0.001	0.001

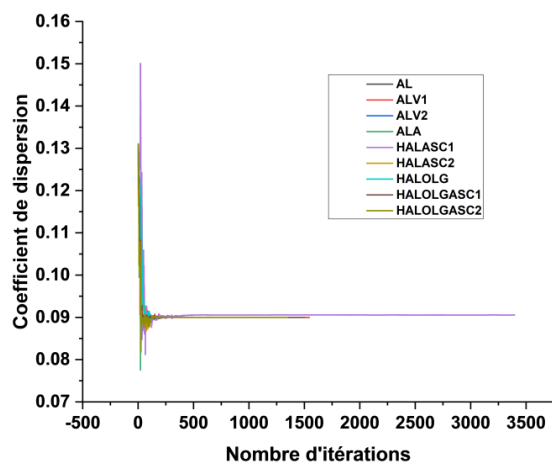
Nous constatons que :

- les essais simulés montrent la convergence vers le même vecteur paramètre de tous les algorithmes développés.
- La vitesse de convergence de l'algorithme de luciole améliorée est nettement plus rapide que celle des autres algorithmes de luciole standard et leurs variantes.
- La vitesse de convergence de l'algorithme de luciole hybride avec algorithme sinus cosinus est nettement plus rapide que celle des

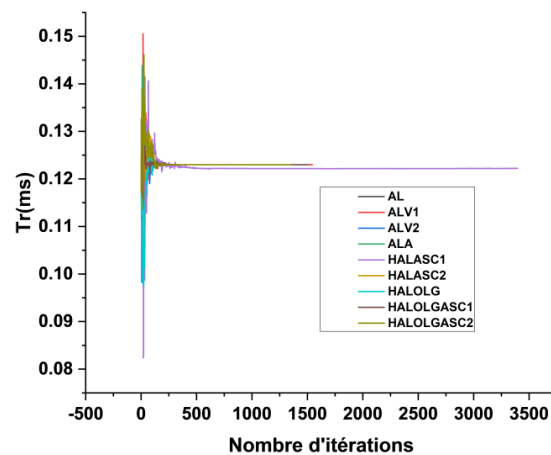
autres algorithmes de luciole standard et leurs variantes.

- La vitesse de convergence de l'algorithme de luciole hybride avec l'optimiseur de loup gris est nettement plus rapide que celle des autres algorithmes de luciole standard et leurs variantes

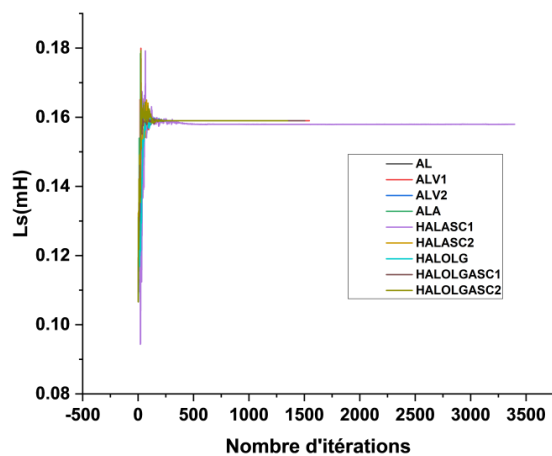
L'évolution des paramètres donnée par les figures 4.2 montre la convergence et la supériorité des différentes proposées.



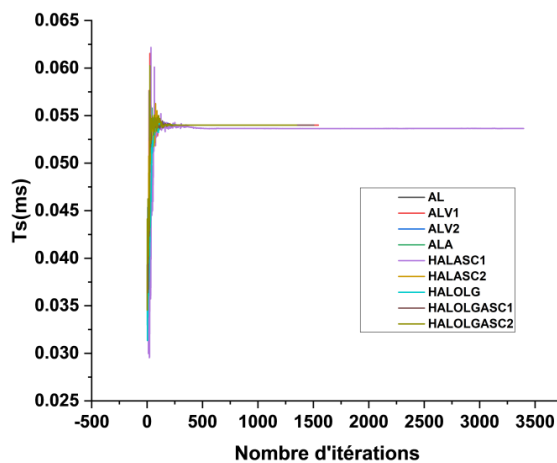
(a)



(b)



(c)



(d)

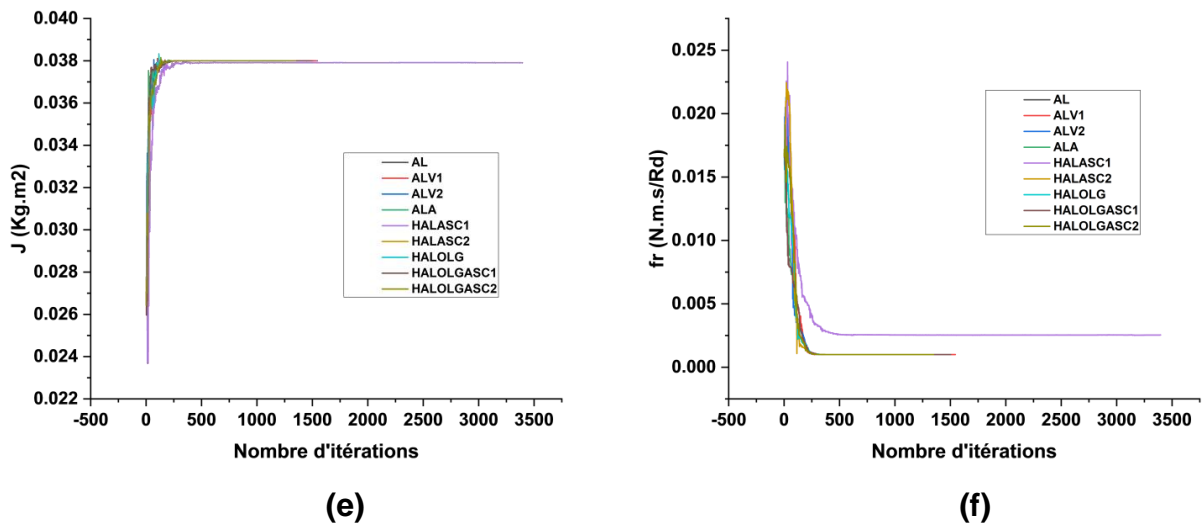


Figure 4.2 Evolution des paramètres de la machine connue

4.3.2. Identification l'aide des données expérimentales

Les données expérimentales ont été collectées sur un oscilloscope numérique à partir de tests transitoires sur deux moteurs asynchrones triphasés qui sont réalisés avec les caractéristiques suivantes :

- Moteur **M1** : 2 pôles, 220/380V, 3kw.
- Moteur **M2** : 4 pôles, 220/380V, 1.5kw.

Cette méthode d'identification nécessite une entrée et une sortie de chaque moteur, l'entrée est la tension du stator et la sortie est leur courant correspondant au même instant, les mesures nécessaires sont obtenues en utilisant 20 kHz comme fréquence d'échantillonnage. Les courbes 4.3 à 4.4 représentent le courant de démarrage et la tension simple correspondante pour chaque moteur.

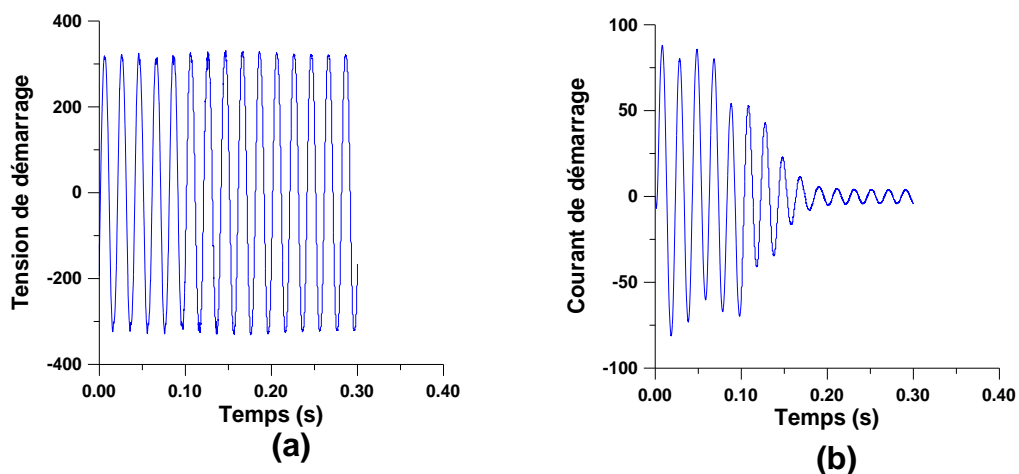


Figure 4.3 Moteur M1 (a) tension mesuré (b) courant mesuré

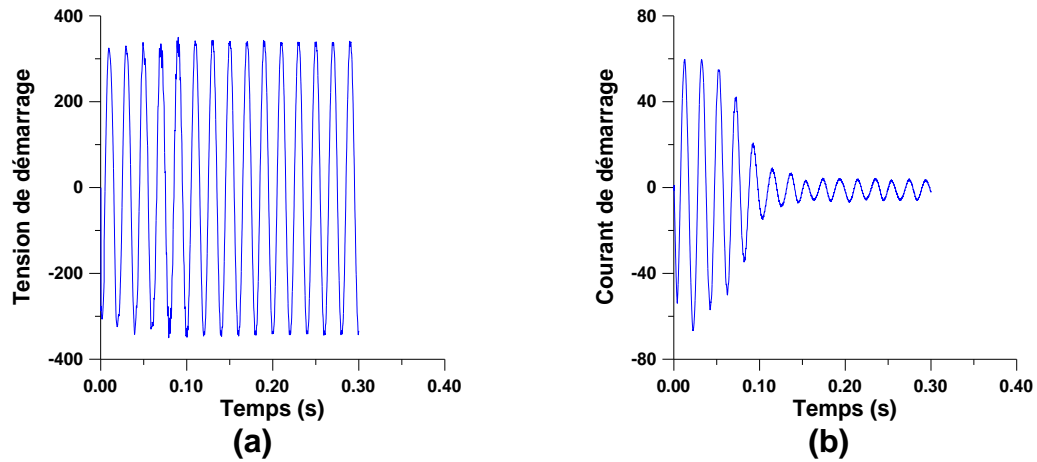


Figure 4.4 Moteur M2 (a) tension mesuré (b) courant mesuré

4.3.2.1. Identification des paramètres de moteur M1

4.3.2.1.1. Modèle à une Cage

L'application des algorithmes développés sur le moteur **M1** conduit aux paramètres donnés respectivement par les tableaux suivants

Tableau 4.3 Identification des paramètres de moteur M1

Paramètres (M3)	ALS	ALV1	ALV2	ALV3	ALA
σ	0.0343	0.0349	0.0336	0.0334	0.0339348
T_r (ms)	195.7	192.4	200.7	202.4	198.56050
L_s (mH)	246.9	242.9	252.8	254.7	250.32197
T_s (ms)	144.2	141.8	147.5	148.5	146.11275
J (Kg.m ²)	0.0223	0.0223	0.0223	0.0223	0.02231066
fr (Nm.s/Rd)	0.00261	0.00262	0.00266	0.00272	0.00263174
Nombre d'itération	848	636	798	1493	275
Temps de calcul (s)	135332	101499	127354	171338	19319
Erreur quadratique obtenue	45725	45719	45712	45721	45709

Tableau 4.4 Identification des paramètres de moteur M1

Paramètres	ALOLG	ALASC1	ALASC2	ALOLGASC1	ALOLGASC2
σ	0.0343	0.0337	0.0341	0.0332	0.0343
T_r (ms)	195.7	193.6	201.9	203.1	197.8
L_s (mH)	246.9	244.7	251.9	254.9	241.9
T_s (ms)	144.2	143.2	146.9	149.3	143.9
J (Kg.m ²)	0.0223	0.0223	0.0223	0.0223	0.0223
fr (Nm.s/Rd)	0.00261	0.00262	0.00266	0.00266	0.00261
Nombre d'itération	416	287	324	409	453
Temps de calcul	29536	20377	23328	29244	32616
Erreur quadratique obtenue	45717	45710	45715	45711	45731

Après une analyse des tableaux (4.3 à 4.4), on constate que :

- tous les algorithmes convergent vers la valeur optimale des paramètres
- La vitesse de convergence de l'algorithme de luciole améliorée est nettement plus rapide que celle de l'algorithme de luciole standard.
- La vitesse de convergence des algorithmes de luciole hybrides est nettement plus rapide que l'algorithme de luciole standard.
- La vitesse de convergence de l'algorithme de luciole hybride avec algorithme sinus cosinus est nettement plus rapide que celle des autres algorithmes de luciole hybrides.
- La vitesse de convergence de l'algorithme de luciole hybride avec l'optimiseur de loup gris est nettement plus rapide que celle des autres algorithmes de luciole standard.

Pour chaque algorithme, la superposition illustrée par les figures (4.5, 4.7, 4.9, 4.11, 4.13) entre le courant mesuré et ceux calculés à l'aide des paramètres estimés de moteur M1, a montré que le courant calculé était proche du courant mesuré dans les deux états transitoires et permanent. Ces résultats indiquent la proximité des valeurs des paramètres de la machine par rapport aux valeurs nominales et la validité de la méthode d'identification. L'erreur entre les courants mesurés et celui-ci calculés à l'aide des paramètres estimés, pour le moteur M1 est montrée par les figures (4.6, 4.8, 4.10, 4.12, 4.14).

L'évolution des paramètres estimée du moteur M1 par :

- algorithme de luciole amélioré et par l'algorithme de luciole hybride avec algorithme de sinus cosinus type 1 et 2 est donné par la figure 4.15.
- Algorithme de luciole standard et amélioré par la figure 4.16.

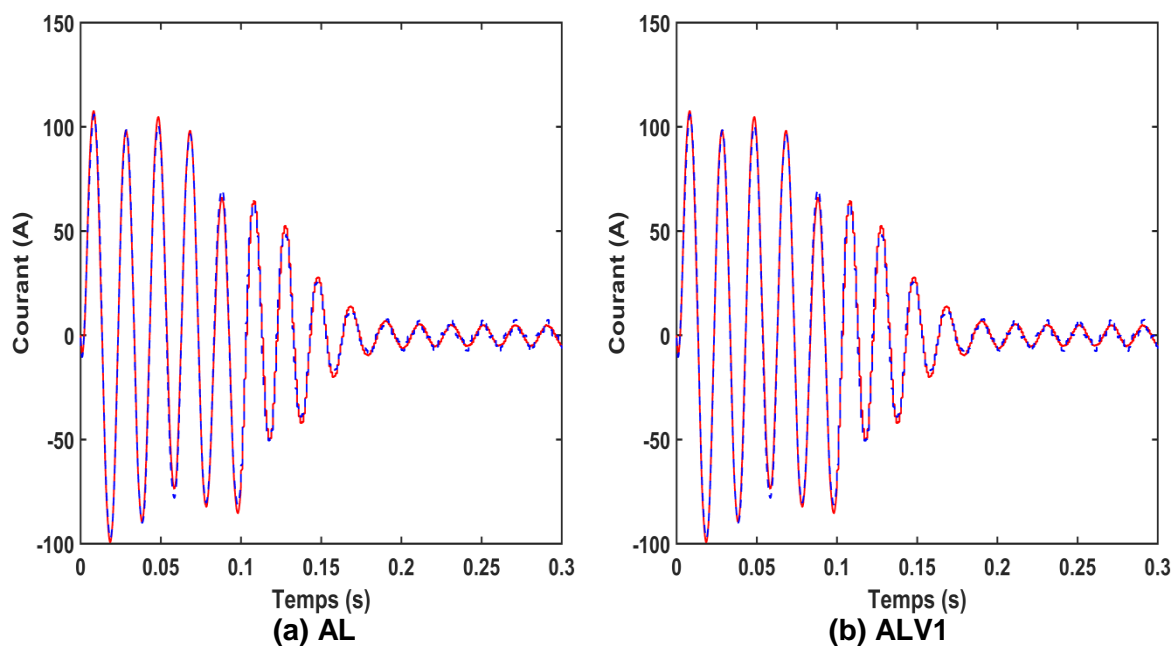


Figure 4.5 Superposition du courant mesuré - - - et calculé - - - à l'aide des paramètres estimés par AL et ALV1 du moteur M1

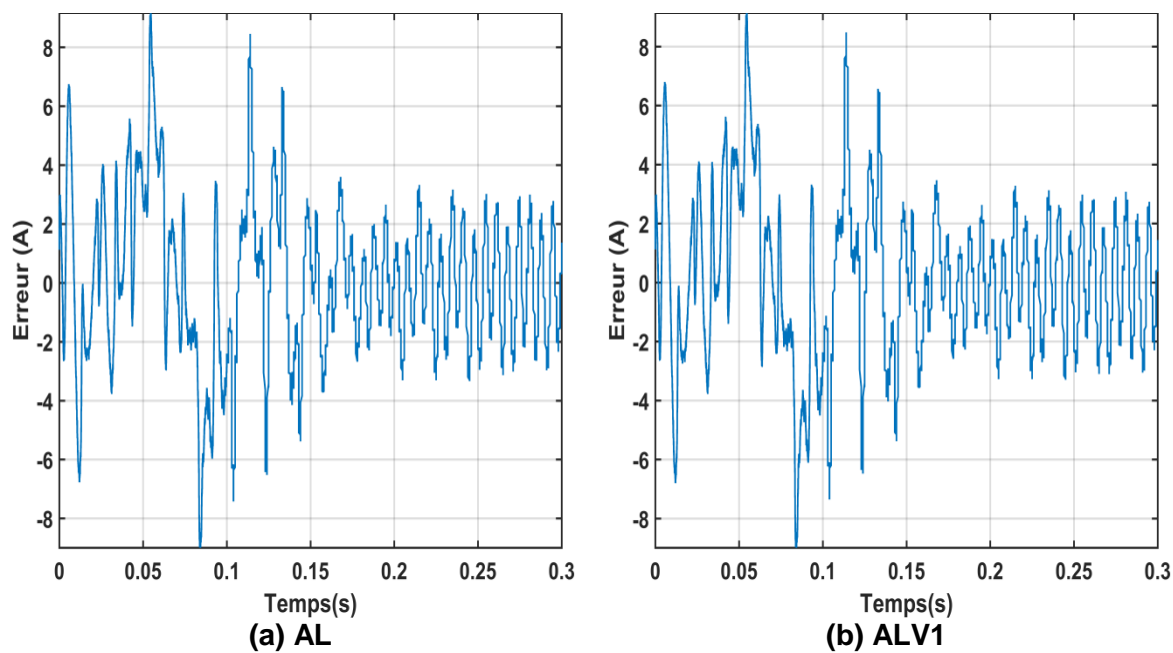


Figure 4.6 Erreur entre le courant mesure et celui calcule avec les paramètres estimés par AL et ALV1 du moteur M1

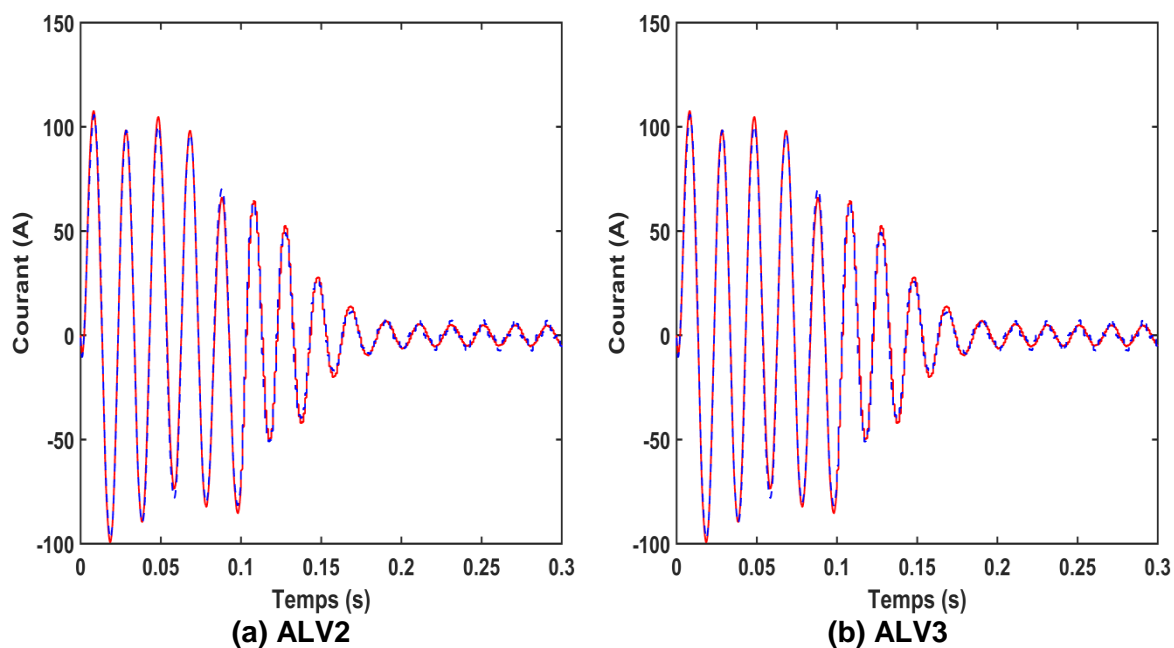


Figure 4.7 Superposition du courant mesuré - - - et calculé - - - à l'aide des paramètres estimés par ALV2 et ALV3 du moteur M1

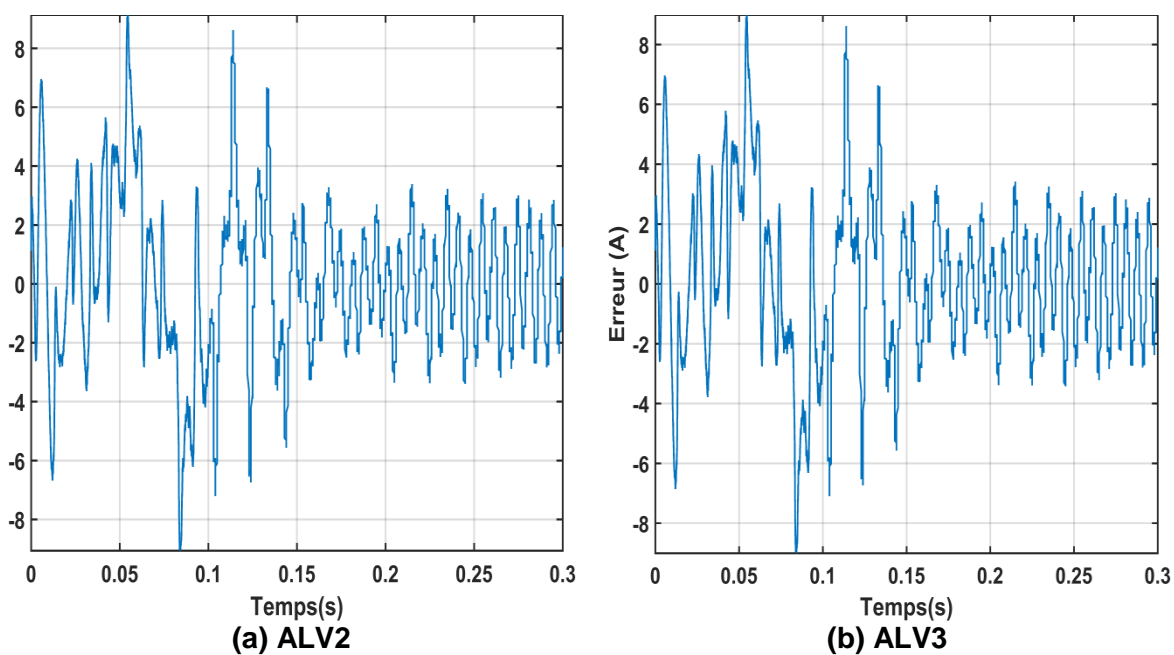


Figure 4.8 Erreur entre le courant mesure et celui calcule avec les paramètres estimés par ALV2 et ALV3 du moteur M1

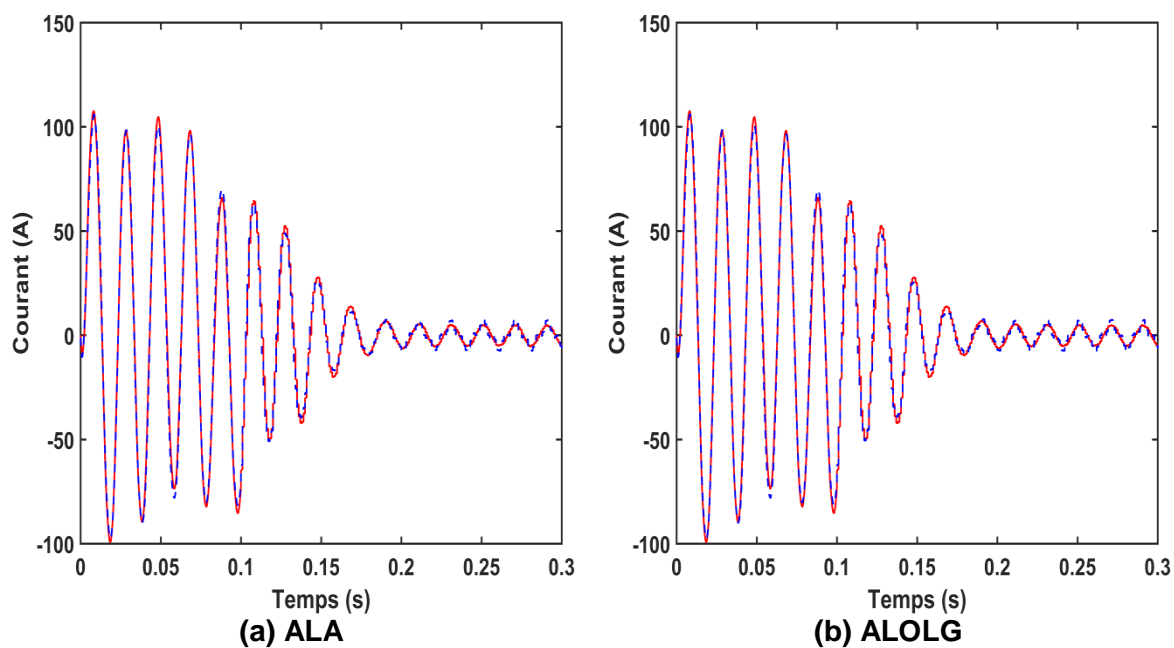


Figure 4.9 Superposition du courant mesuré - - - et calculé - - - à l'aide des paramètres estimés par ALA et ALOLG du moteur M1

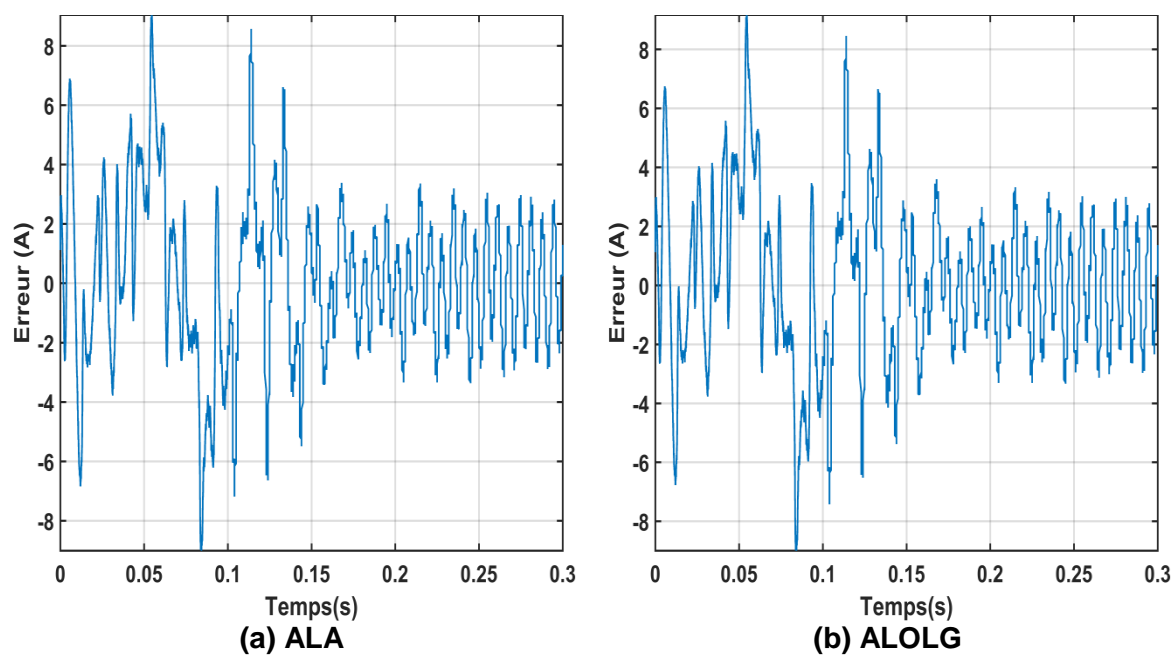


Figure 4.10 Erreur entre le courant mesure et celui calculé avec les paramètres estimés par ALA et ALOLG du moteur M1

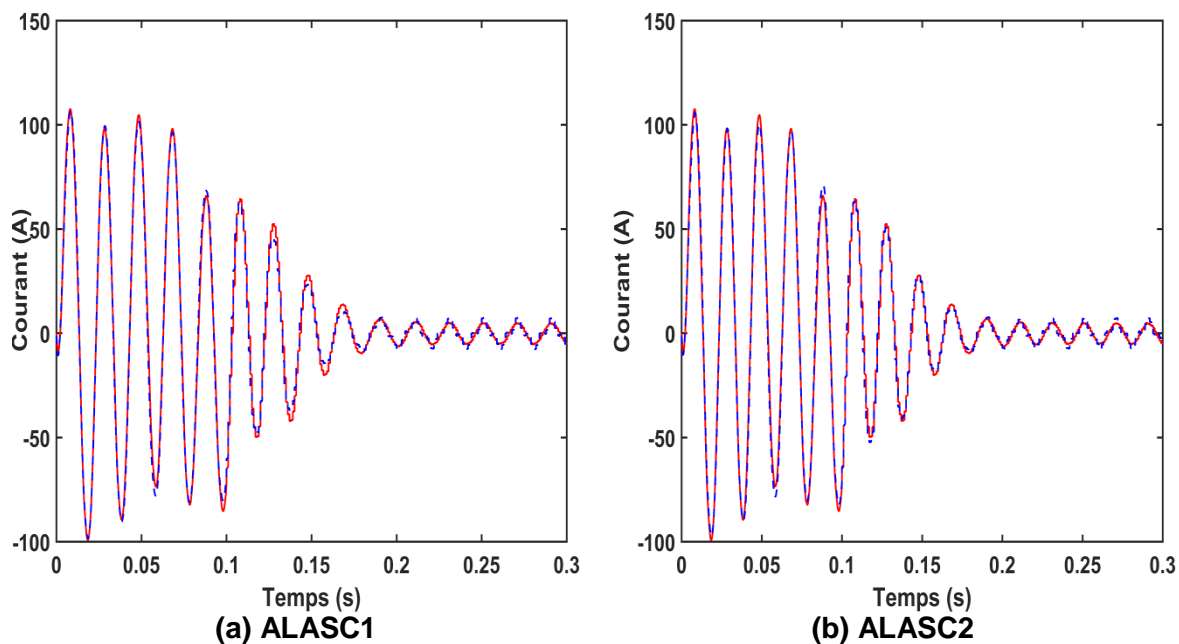


Figure 4.11 Superposition du courant mesuré - - - et calculé - - - à l'aide des paramètres estimés par ALASC1 et ALASC2 du moteur M1

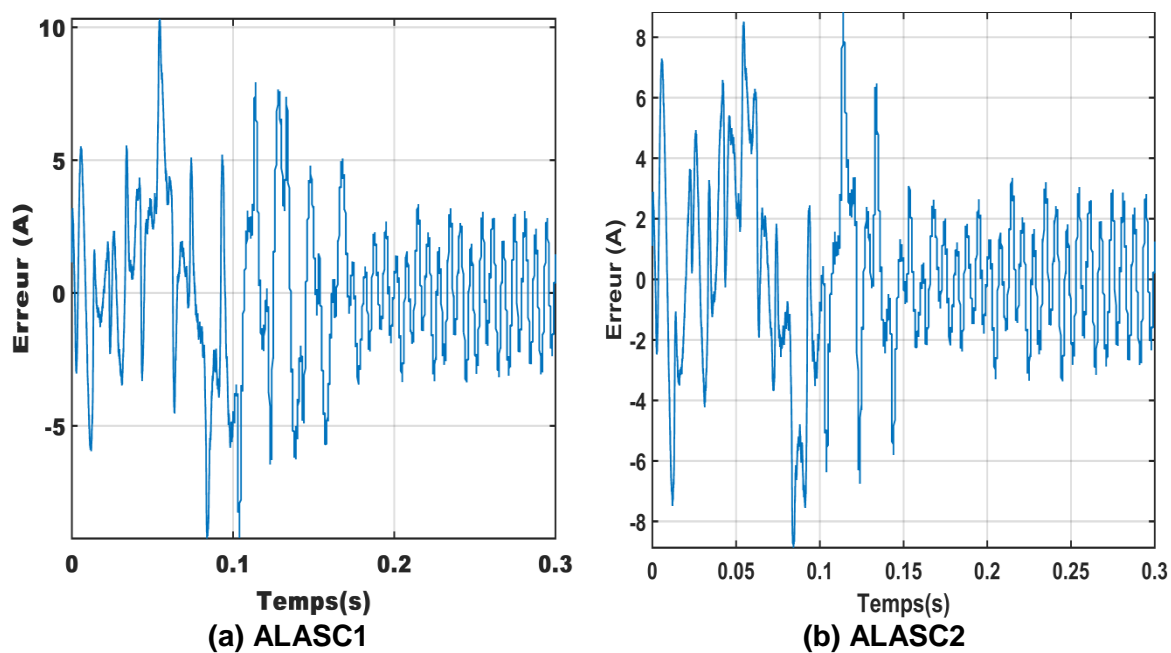


Figure 4.12 Erreur entre le courant mesure et celui calculé avec les paramètres estimés par ALASC1 et ALASC2 du moteur M1

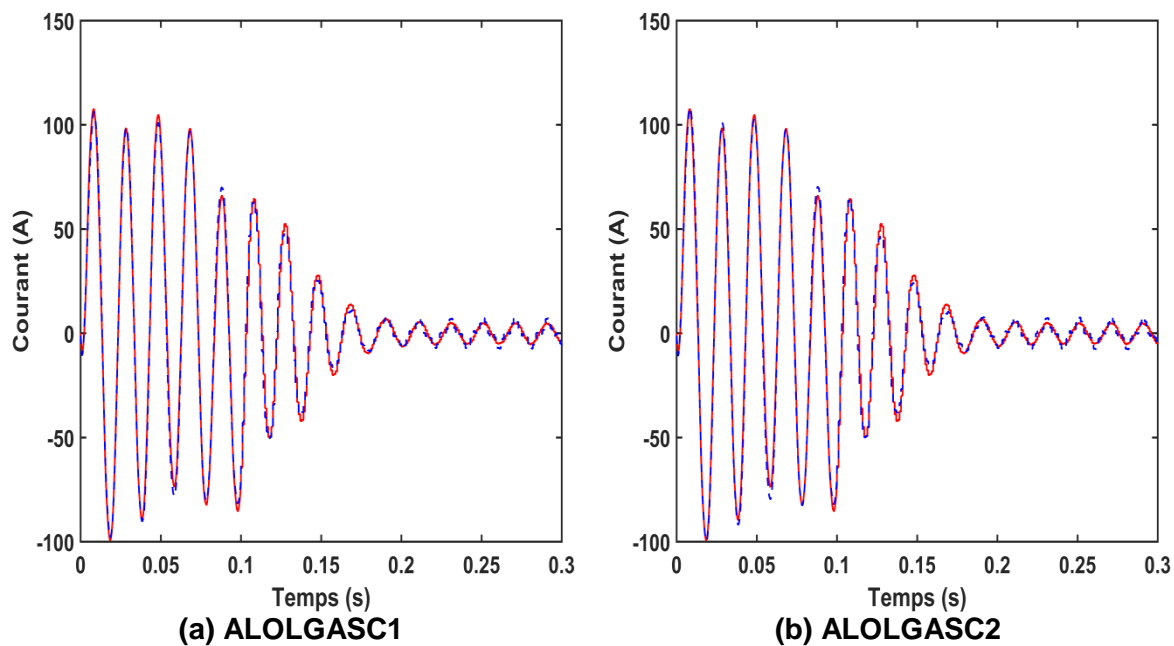


Figure 4.13 Superposition du courant mesuré - - - et calculé - - - à l'aide des paramètres estimés par ALOLGASC1 et ALOLGASC2 du moteur M1

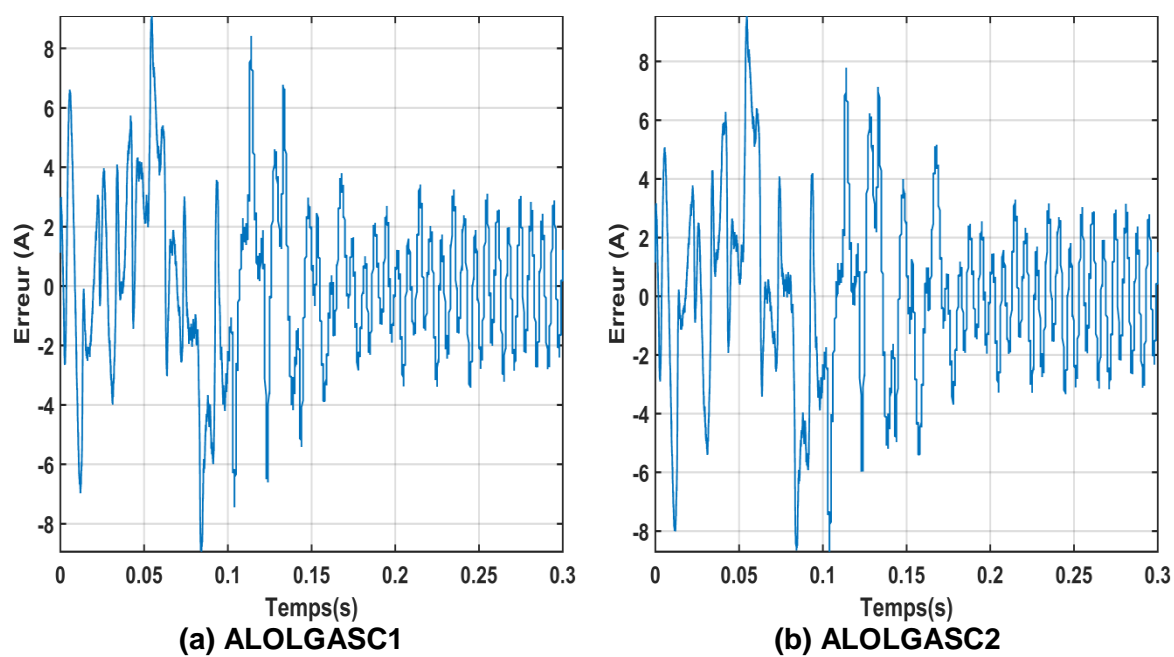
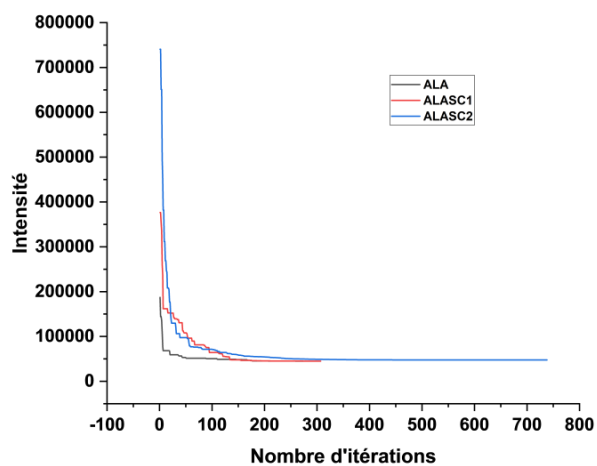
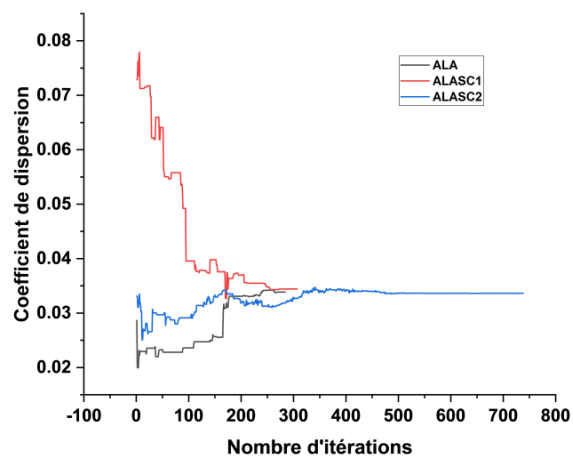


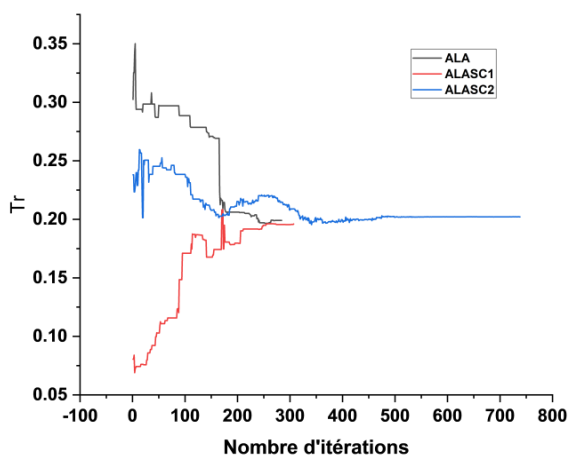
Figure 4.14 Erreur entre le courant mesure et celui calculé avec les paramètres estimés par ALOLGASC1 et ALOLGASC2 du moteur M1



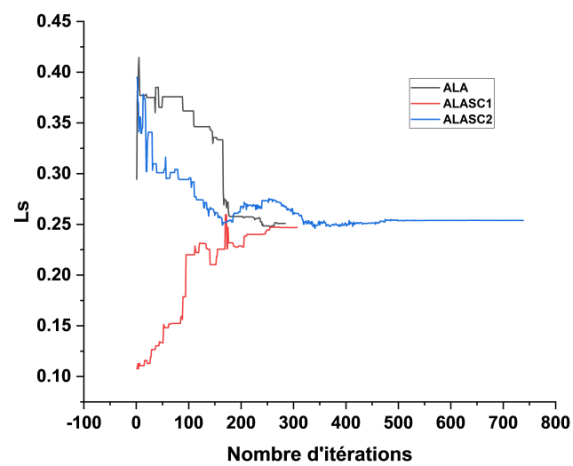
(a)



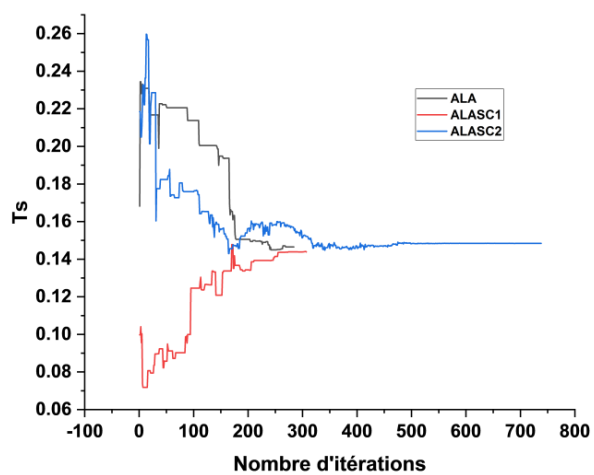
(b)



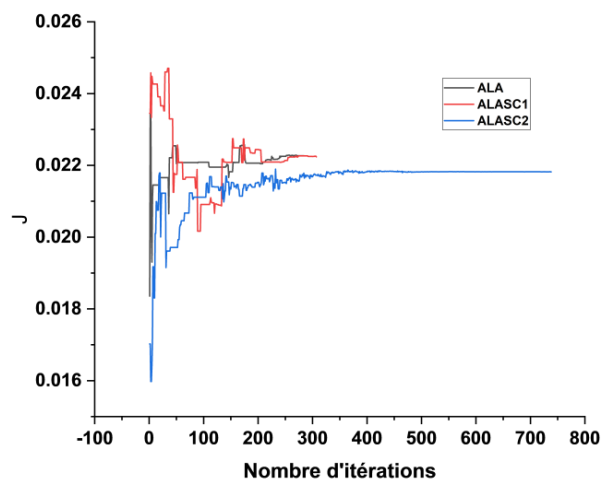
(c)



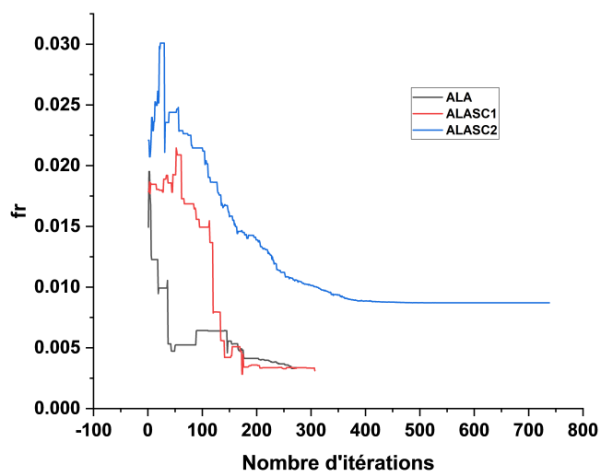
(d)



(e)

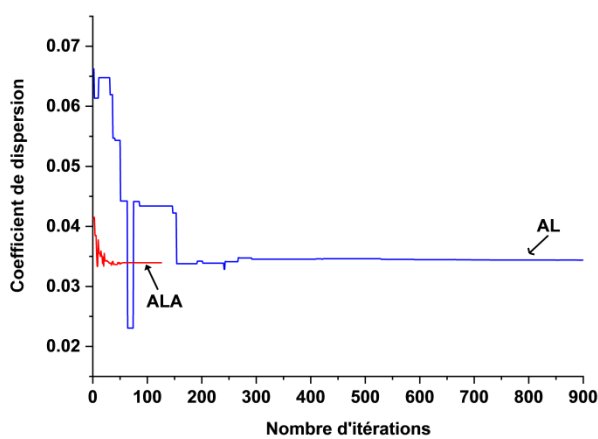


(f)

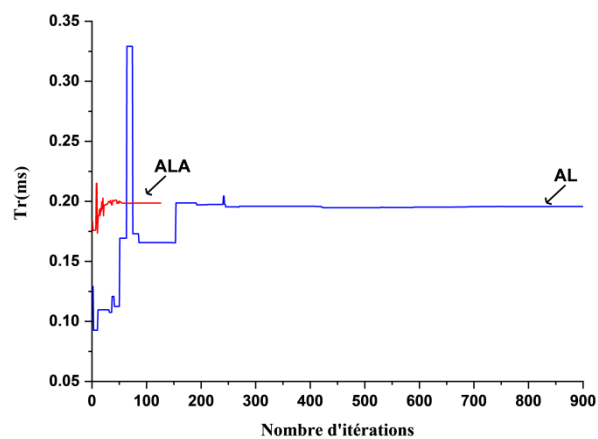


(g)

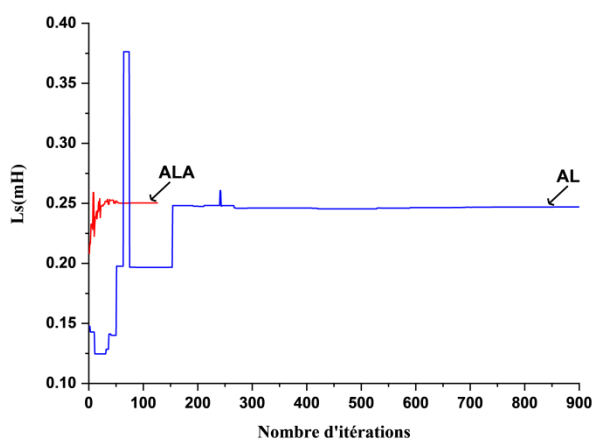
Figure 4.15 Evolution des paramètres de la machine M1 par ALA, ALASC1,2



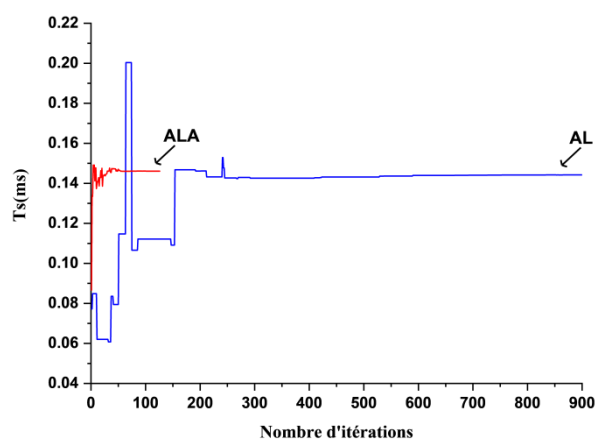
(a)



(b)



(c)



(d)

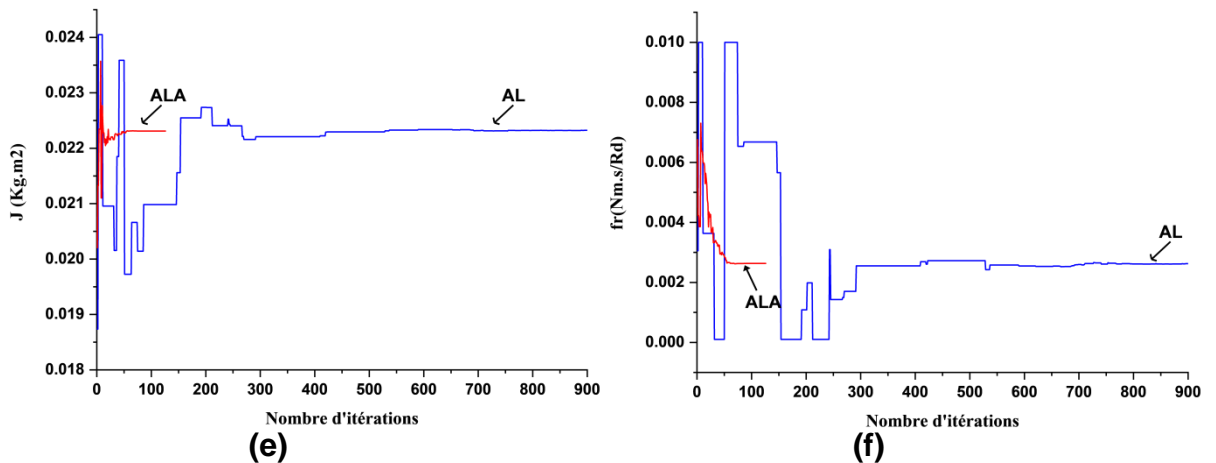


Figure 4.16 Evolution des paramètres de la machine M1 par AL, ALA.

4.3.2.1.2. Modèle à deux cages

Pour le modèle à deux cages, l'optimisation est assurée seulement par quatre algorithmes :

- algorithme de luciole amélioré ;
- Algorithme de luciole hybride avec l'optimiseur de loup gris ;
- Algorithme sinus et cosinus type 1 ;
- Algorithme sinus et cosinus type 2.

Le tableau 4.5 donne les paramètres estimés pour le moteur **M1**, son examen montre que les tous les essais convergent vers le même vecteur paramètre.

Tableau 4.5 Paramètres du Moteur M1 en considérant la deuxième cage

Paramètres	ALA	ALOLG	ALASC1	ALASC2
σ_1	0.1932	0.1931	0.1932	0.1931
σ_2	0.0323	0.03212	0.0323	0.03212
T_{r1} (ms)	23.2	23.2	23.2	23.2
T_{r2} (ms)	184.8	184.8	185.1	184.8
L_s (mH)	245.4	245.4	245.4	245.4
T_s (ms)	151.8	151.8	152.1	151.8
P_1	0.8462	0.846	0.842	0.846
J (Kg.m ²)	0.0231	0.0233	0.0233	0.0233
fr (Nm.s/Rd)	0.00266	0.003	0.00266	0.003
Nombre d'itération	195	365	235	278
Temps de calcul (s)	40755	77015	49821	58659
Erreur quadratique obtenue	35606	36294	35867	36294

Les figures 4.17 et 4.19 montrent que le courant calculé avec le vecteur paramètre estimé se superpose avec le courant mesuré aussi bien durant le régime transitoire que le régime permanent avec une erreur acceptable due au

problème des mesures et aussi au modèle. L'erreur entre le courant mesuré et celui-ci calculé est illustrée par les figures 4.18 et 4.20.

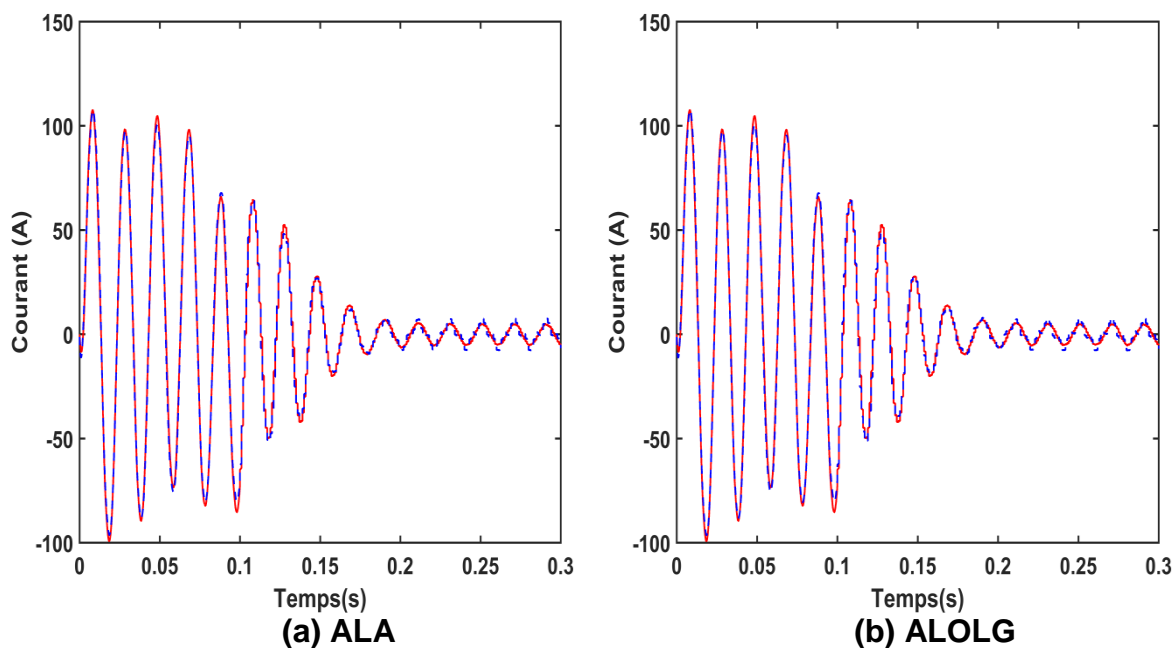


Figure 4.17 Superposition du courant mesuré - - - et calculé - - - à l'aide des paramètres estimés par ALA et ALOG du moteur M1

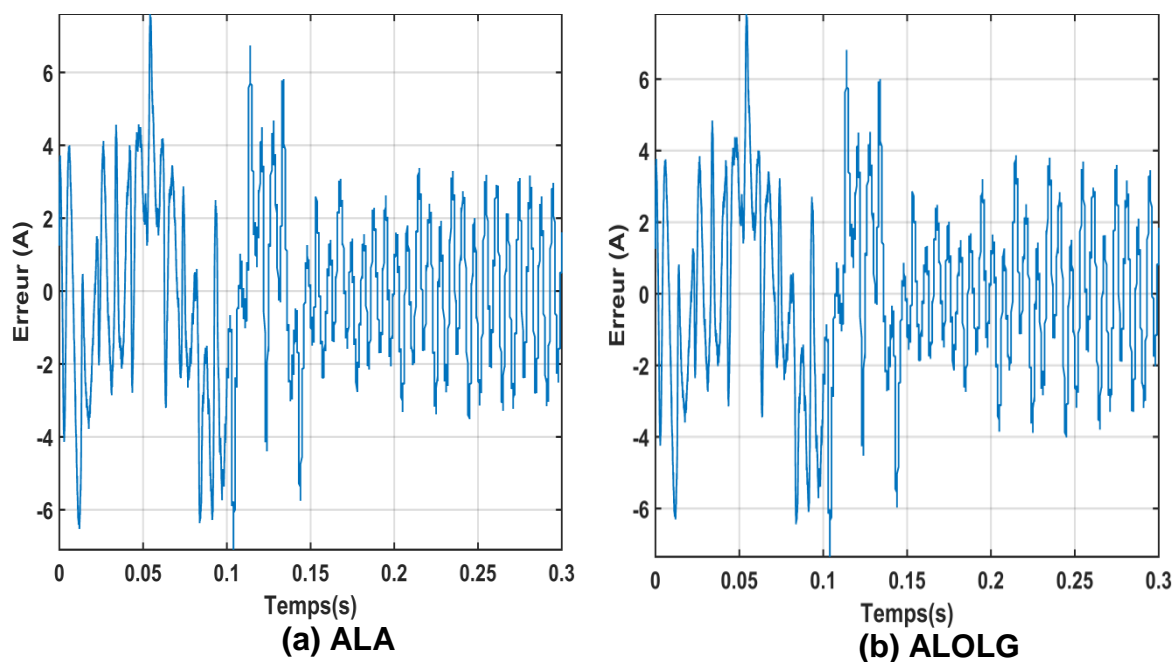


Figure 4.18 Erreur entre le courant mesuré et celui calculé avec les paramètres estimés par ALA et ALOG du moteur M1

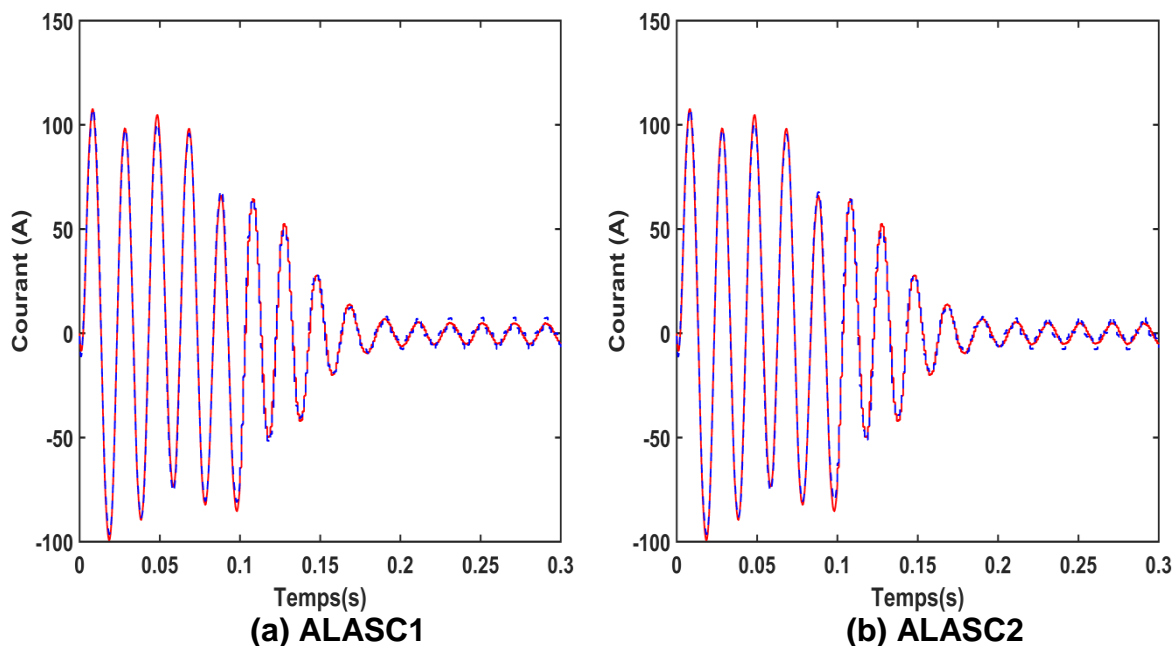


Figure 4.19 Superposition du courant mesuré - - - et calculé - - - à l'aide des paramètres estimés par ALASC1 et ALASC2 du moteur M1

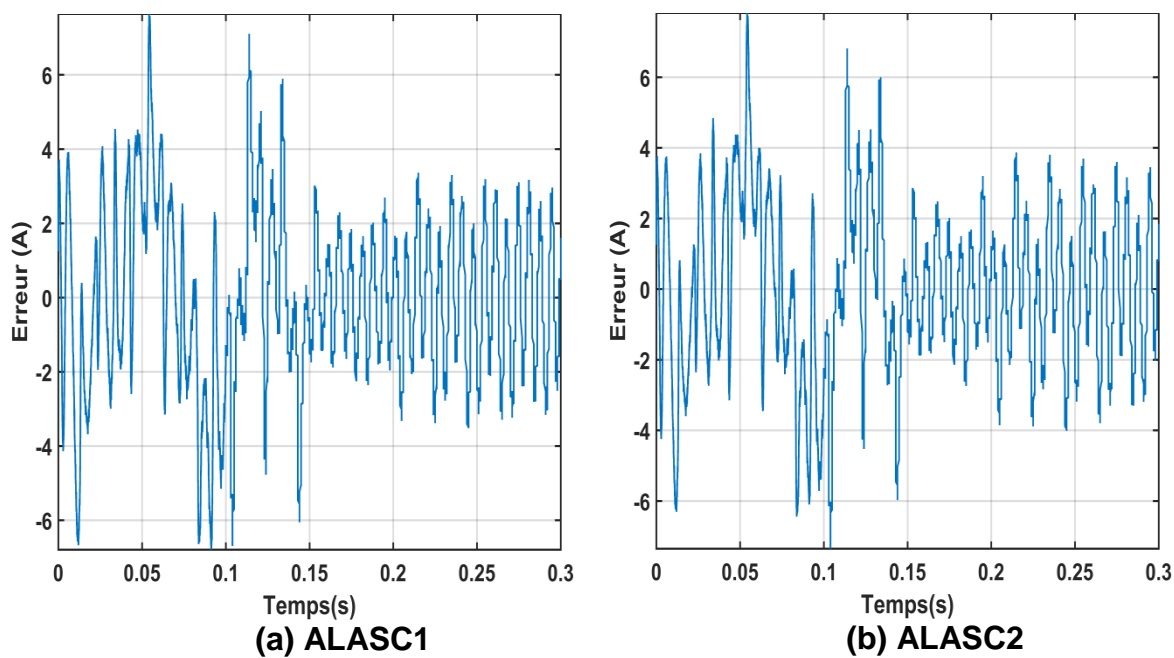
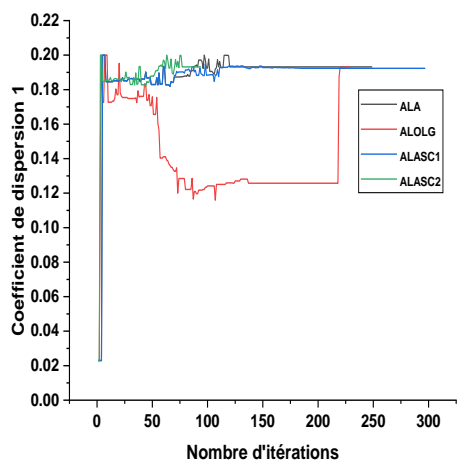
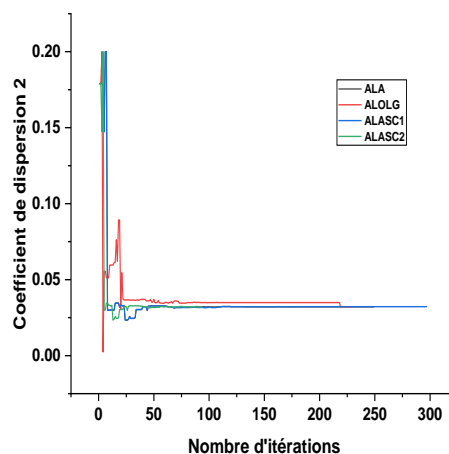


Figure 4.20 Erreur entre le courant mesuré et celui calculé avec les paramètres estimés par ALASC1 et ALASC2 du moteur M1

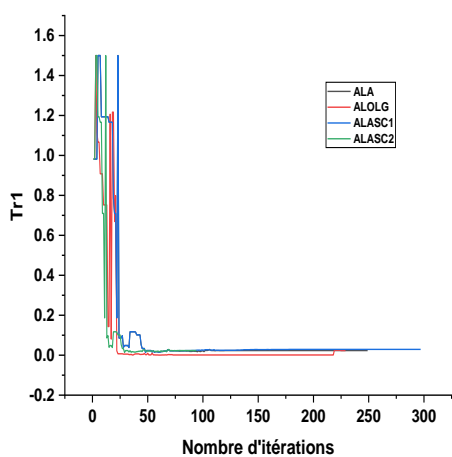
La figure 4.21 montre l'évolution des paramètres estimée du moteur M1.



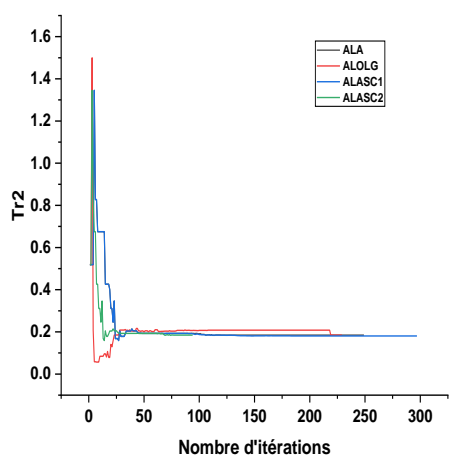
(a)



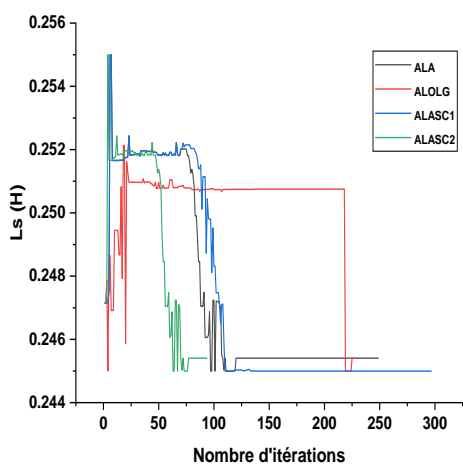
(b)



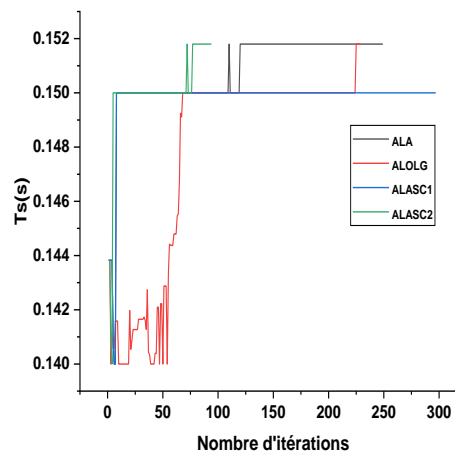
(c)



(d)



(e)



(f)

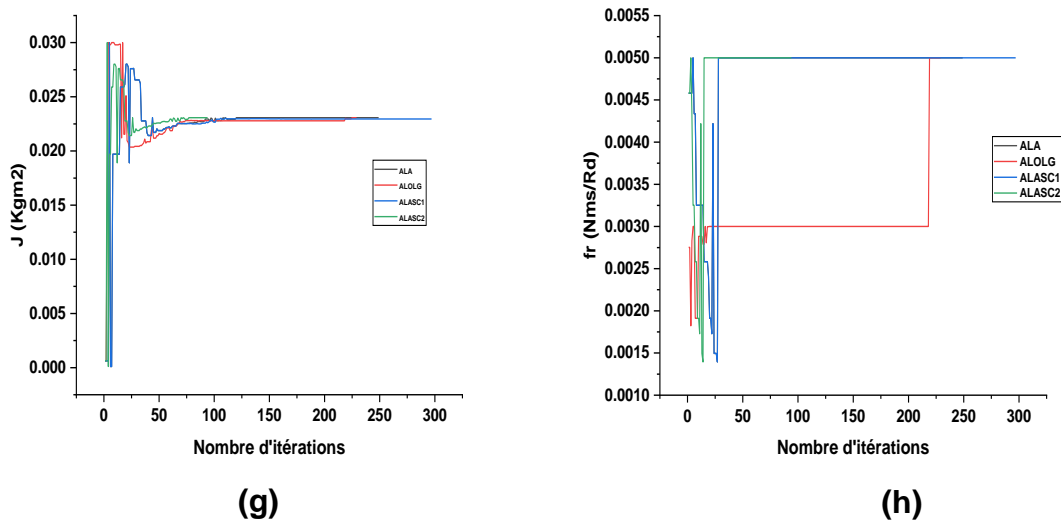


Figure 4.21 evolution des paramètres estimé avec les algorithmes ALA, ALOLG, ALASC1 et ALASC2

4.3.2.2. Identification des paramètres de moteur M2

4.3.2.2.1. Modèle à une Cage

L'application des algorithmes développés sur le moteur M2 conduit aux paramètres donnés respectivement par les tableaux 4.6 et 4.7.

Tableau 4.6 Identification des paramètres de moteur M2

Paramètres (M1)	AL	ALV1	ALV2	ALV3	ALA
σ	0.0689	0.0690	0.0696	0.0692	0.0689
Tr (ms)	135.6	135.4	134	135.1	135.6
Ls (mH)	190.3	190.0	188.2	189.6	190.3
Ts (ms)	87.5	87.4	86.6	87.2	87.3
J (Kg.m ²)	0.0353	0.0353	0.0353	0.0353	0.0353
fr (Nm.s/Rd)	0.0099	0.0099	0.0099	0.0099	0.0099
Nombre d'itération	512	499	507	531	337
Erreur quadratique obtenue	59198	59214	59391	59225	59196

On constate que :

- tous les algorithmes convergent vers la valeur optimale des paramètres.
- La vitesse de convergence de l'algorithme de luciole hybride avec algorithme sinus cosinus est nettement plus rapide que celle des autres algorithmes de luciole hybrides.

Tableau 4.7 Identification des paramètres de moteur M2

Paramètres	ALOLG	ALASC1	ALASC2	ALOLGASC1	ALOLGASC2
σ	0.069	0.0689	0.0689	0.069	0.0658
T_r (ms)	135.4	135.7	135.7	135.4	143.2
L_s (mH)	190.0	190.3	190.3	190.0	199.8
T_s (ms)	87.5	87.5	87.5	57.5	91.7
J (Kg.m ²)	0.0353	0.0353	0.0353	0.0353	0.0351
fr (Nm.s/Rd)	0.099	0.099	0.099	0.099	0.0123
Nombre d'itération	427	349	389	415	461
Erreur quadratique	59220	59189	59193	59225	60148

Les figures (4.23, 4.25, 4.27, 4.29 et 4.31) montrent la superposition des courant mesuré et calculé. Le erreur entre le courant mesuré et ceux calcule à l'aide des paramètres estimés, sont également données par les figures (4.24, 4.26, 4.28, 4.30 et 4.32). L'évolution des paramètres estimés est montrée dans l'appendice B.

4.3.2.2.2. Modèle à deux cages

Le tableau 4.8 donne les paramètres estimés pour le moteur M2, son examen montre que les tous les essais convergents vers le même vecteur.

Tableau 4.8 Paramètres du Moteur M2 en considérant la deuxième cage

Paramètres	ALA	ALOLG	ALASC1	ALASC2
σ_1	0.4339	0.434	0.434	0.4351
σ_2	0.06085	0.0609	0.0615	0.06102
T_{r1} (ms)	61.5	61.6	62.1	61.8
T_{r2} (ms)	161.8	161.8	162.3	162.0
L_s (mH)	210.0	210.0	212.1	209.8
T_s (ms)	100.0	100.0	101.9	99.8
P_1	0.619	0.6198	0.6232	0.6198
J (Kg.m ²)	0.0349	0.0349	0.0349	0.0349
fr (Nm.s/Rd)	0.020	0.020	0.020	0.020
Nombre d'itération	737	854	719	811
Erreur quadratique obtenue	35606	35867	36294	36294

4.3.3. Résultats et Discussions

Pour valider les résultats des algorithmes développés par rapport à l'algorithme de luciole standard, en temps de calcul et nombre d'itérations pour une meilleure convergence (vitesse de convergence) vers la meilleure solution optimale, on voit dans les figures précédentes que notre objectif est atteint, et cela comme suit :

- La vitesse de convergence de l'algorithme de luciole améliorée est plus rapide que l'algorithme standard.
- La vitesse de convergence des algorithmes hybride est plus rapide que l'algorithme de luciole standard.
- La vitesse de convergence de l'algorithme de luciole hybride avec l'algorithme sinus cosinus est plus rapide que toutes les autres.

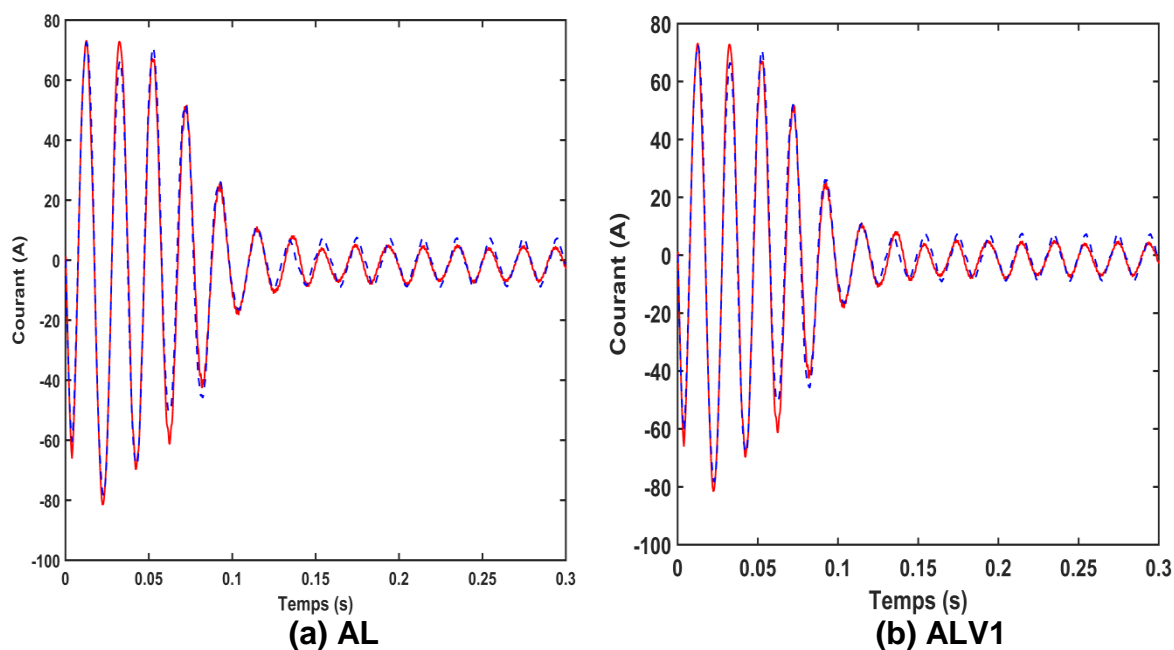


Figure 4.22 Superposition du courant mesuré - - - et calculé - - - à l'aide des paramètres estimés par AL et ALV1 du moteur M2

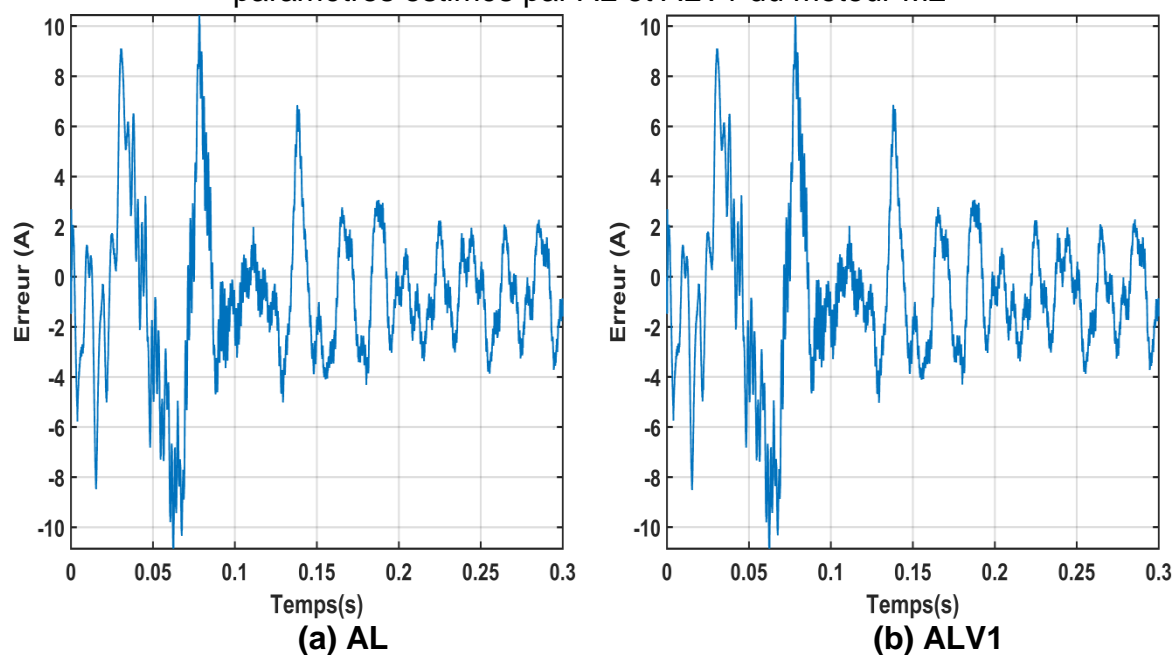


Figure 4.23 Erreur entre le courant mesure et celui calcule avec les paramètres estimes par AL et ALV1 du moteur M2

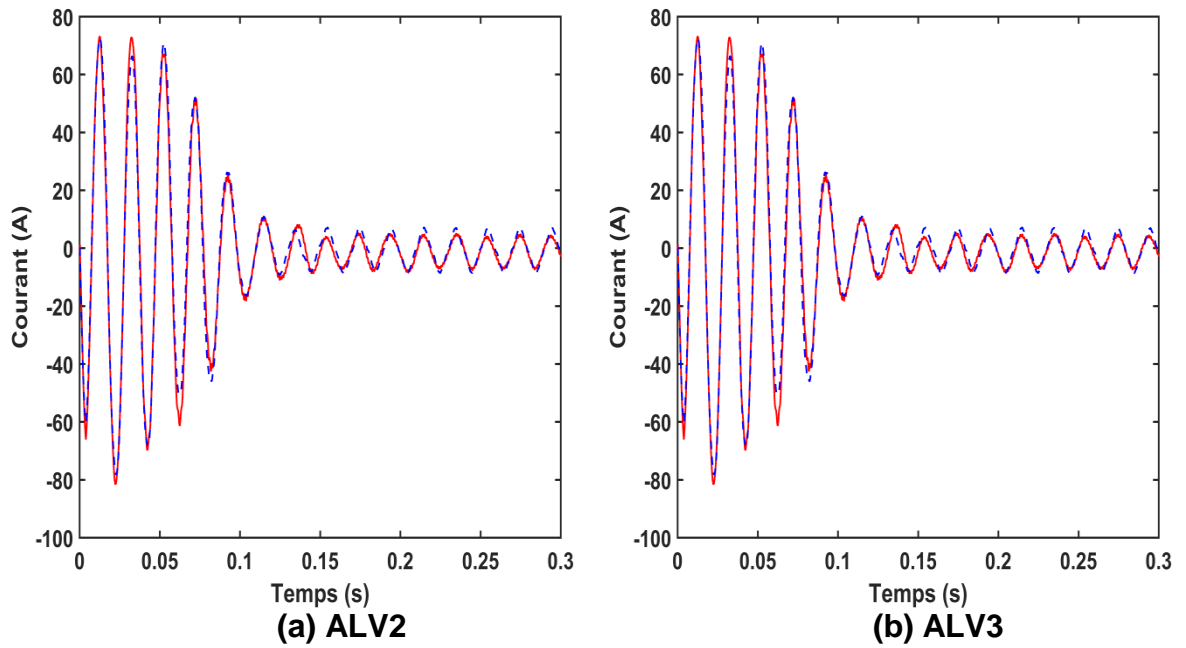


Figure 4.24 Superposition du courant mesuré - - - et calculé - - - à l'aide des paramètres estimés par ALV2 et ALV3 du moteur M2

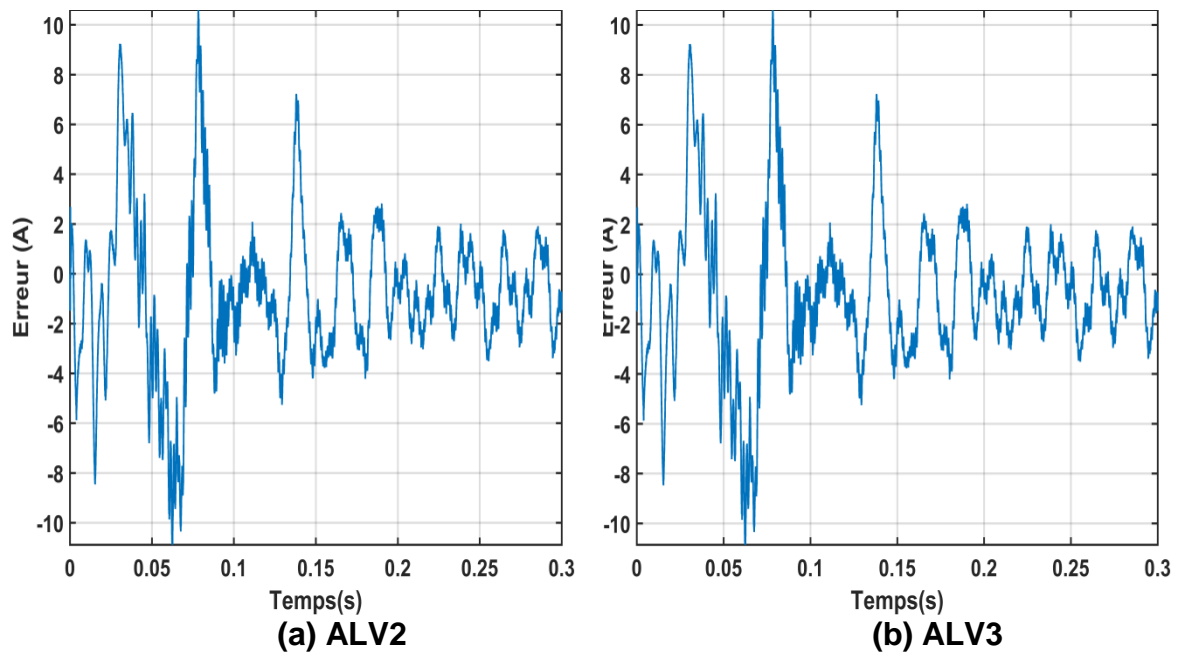


Figure 4.25 Erreur entre le courant mesure et celui calcule avec les paramètres estimés par ALV2 et ALV3 du moteur M2

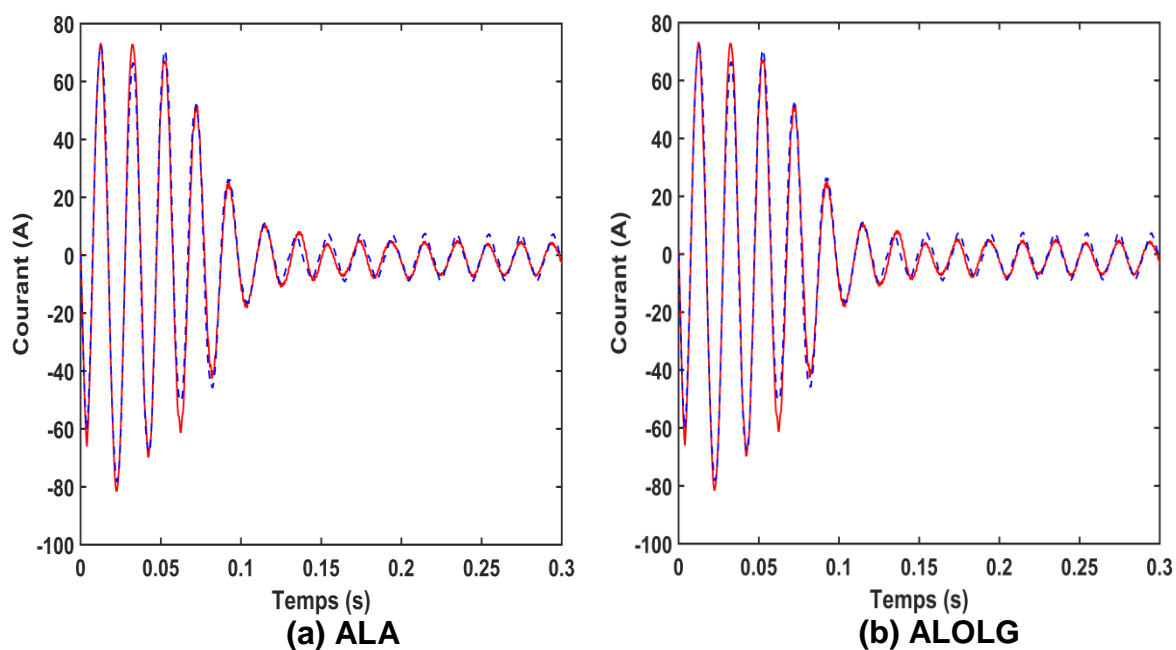


Figure 4.26 Superposition du courant mesuré - - - et calculé - - - à l'aide des paramètres estimés par ALA et ALOLG du moteur M2

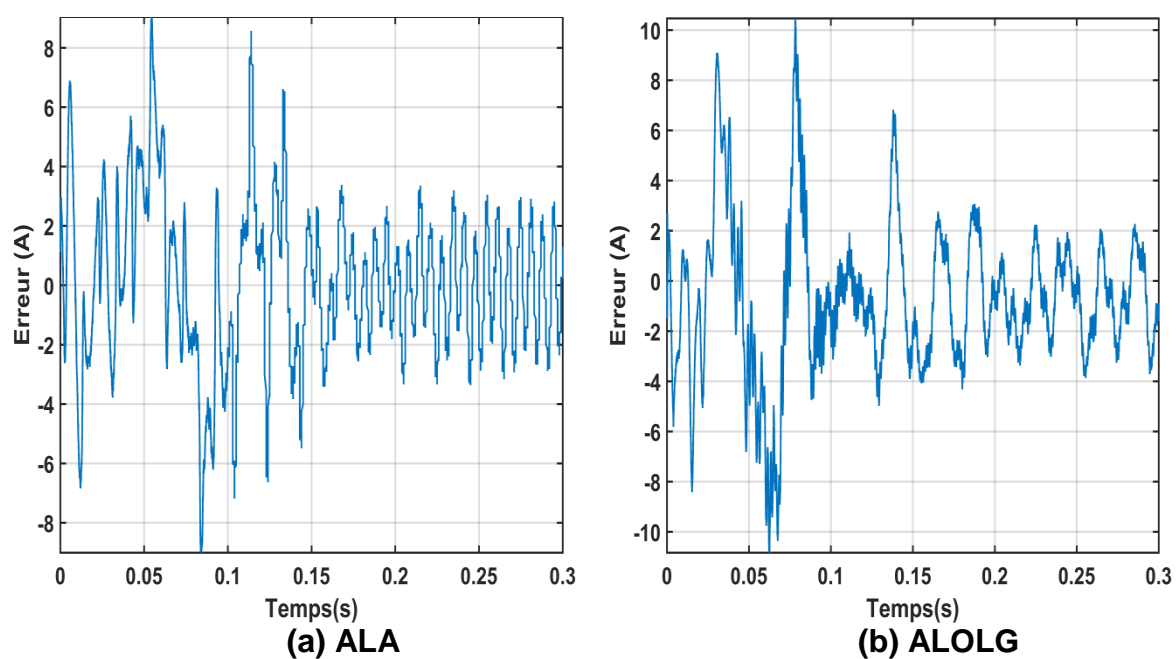


Figure 4.27 Erreur entre le courant mesure et celui calculé avec les paramètres estimés par ALA et ALOLG du moteur M2

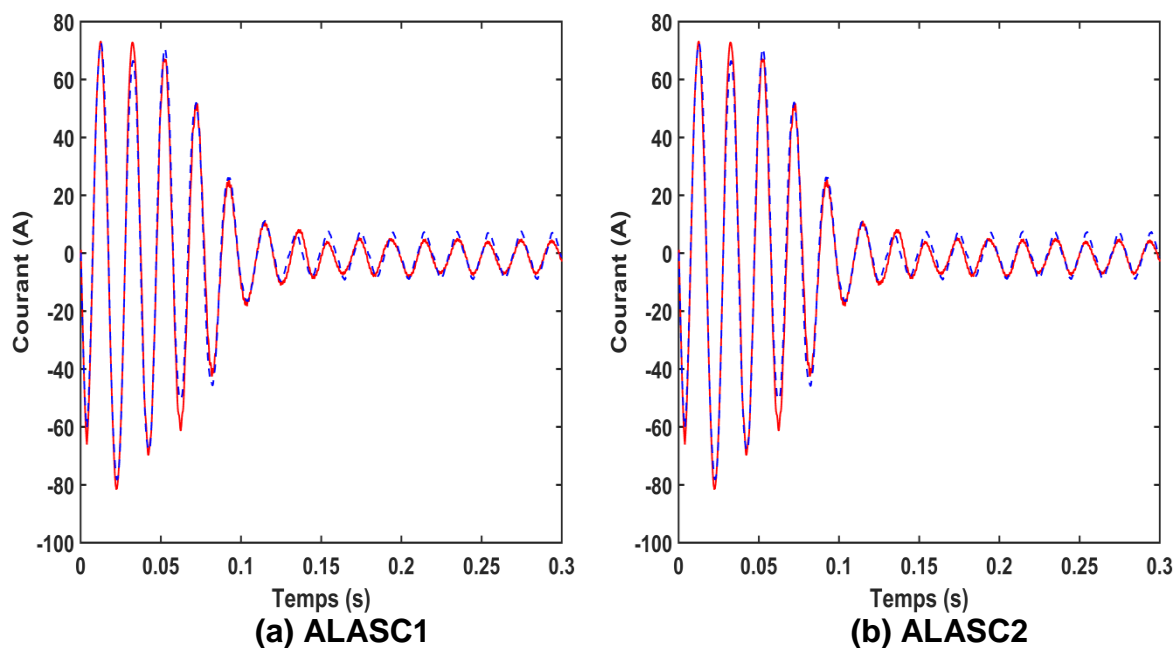


Figure 4.28 Superposition du courant mesuré - - - et calculé - - - à l'aide des paramètres estimés par ALASC1 et ALASC2 du moteur M2

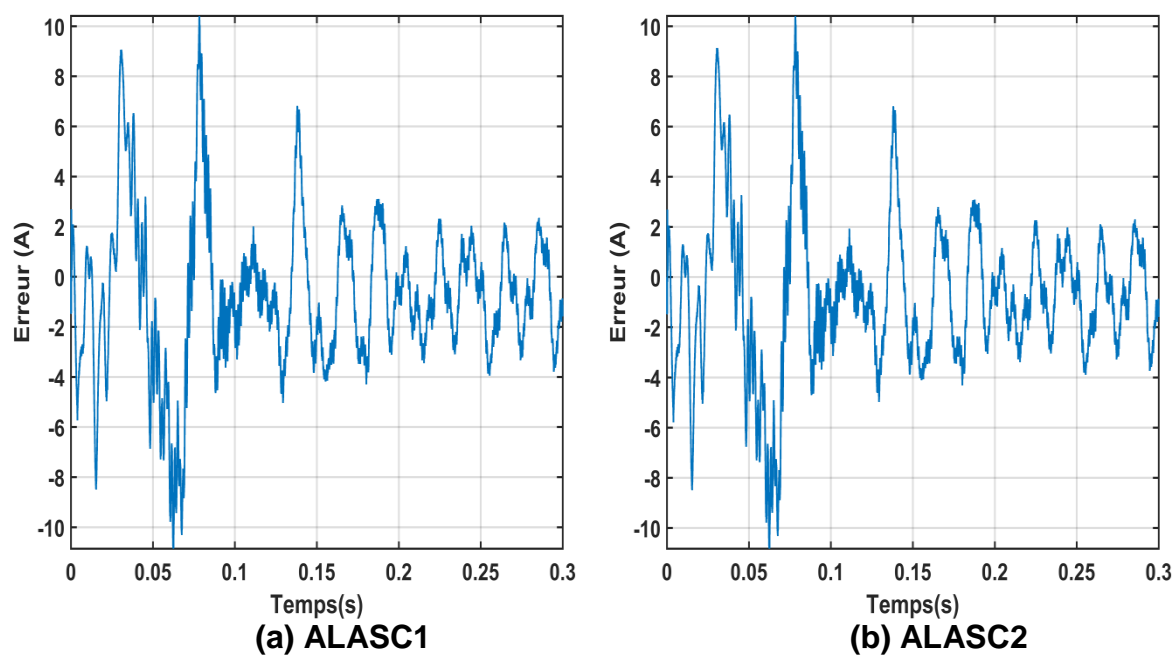


Figure 4.29 Erreur entre le courant mesure et celui calculé avec les paramètres estimés par ALASC1 et ALASC2 du moteur M2

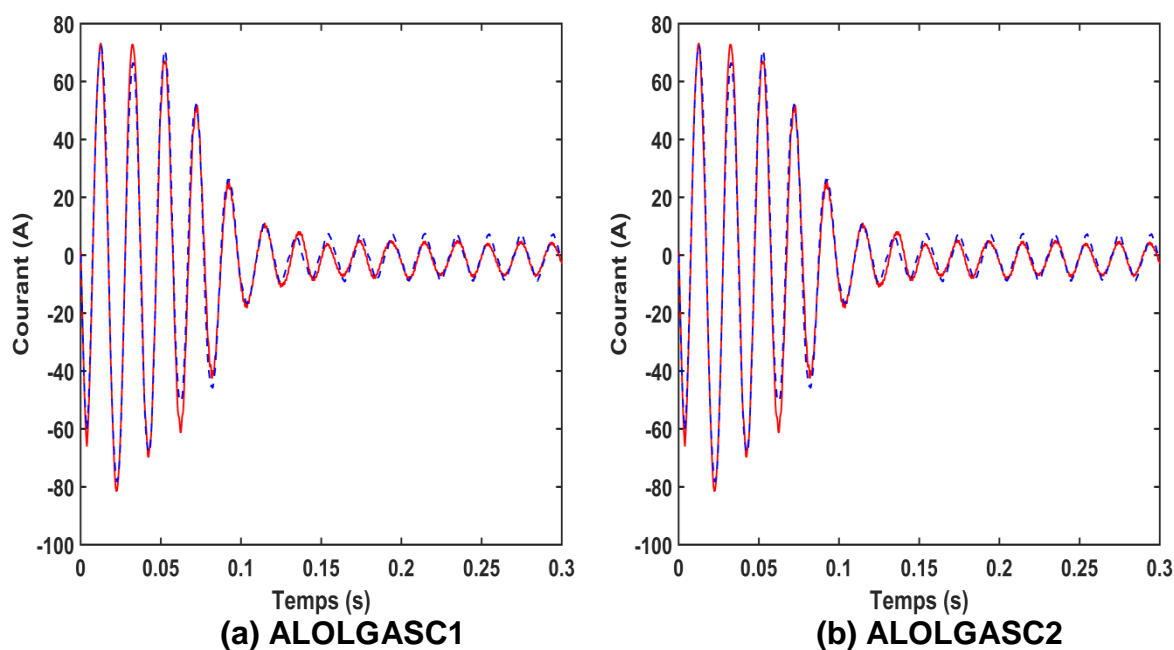


Figure 4.30 Superposition du courant mesuré - - - et calculé - - - à l'aide des paramètres estimés par ALOLGASC1 et ALOLGASC2 du moteur M2

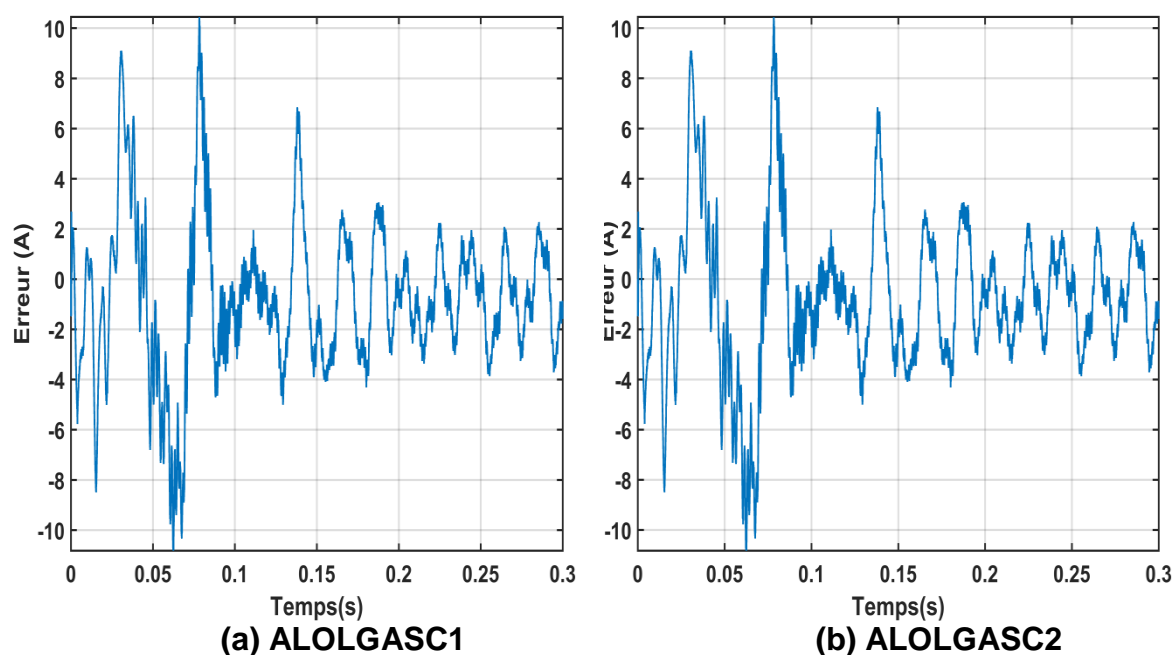


Figure 4.31 Erreur entre le courant mesure et celui calcule avec les paramètres estimés par ALOLGASC1 et ALOLGASC2 du moteur M2

L'algorithme de luciole standard a une bonne exploration de l'espace de recherche, mais a une mauvaise exploitation des zones déjà trouvée ce qui ralentit la vitesse de convergence. Tandis que L'algorithme de luciole amélioré et les algorithmes hybrides imposent une exploration progressive de l'espace de

recherche tout en intensifiant l'exploitation des zones les plus prometteuses. Ainsi, elles imposent un compromis équilibré entre l'exploitation des meilleures solutions disponibles à un moment donné et l'exploration de l'espace de recherche est réalisée.

L'examen des tableaux (4.3 à 4.9) montre que tous les algorithmes convergent vers le même vecteur des paramètres, et ceci en partant d'une population initiale aléatoire située dans le même espace de recherche pour les deux, ce qui est l'optimum global. Donc, les algorithmes de lucioles développées sont capables de fournir des résultats très compétitifs. Ce qui prouve que l'utilisation du courant d'alimentation d'une phase statorique et de la tension simple correspondante est suffisante pour la détermination des paramètres électriques et mécaniques de la machine. Aussi, l'analyse des résultats permet de montrer que :

- L'erreur simple est plus importante dans le régime permanent que dans le régime transitoire.
- Le nombre d'itérations du modèle à deux cages du moteur M1 est moins inférieur que le modèle à une cage, ce qui n'est pas le cas dans le moteur M2.
- Les valeurs des paramètres trouvés par le deuxième modèle (Deux cages) sont très proches à celle du premier modèle (une Cage) pour les deux moteurs M1 et M2.
- $\sigma_1 > \sigma_2$, $T_{r1} < T_{r2}$ ce qui implique $R_{r1} > R_{r2}$ et $L_{r1} < L_{r2}$.
- Le modèle à deux cages demande un effort de calcul plus grand que le modèle à une cage.

Pour évaluer l'effet de la deuxième cage sur la précision et la minimisation de l'erreur quadratique. On définit la notion d'erreur pourcentage qu'est le rapport entre la somme des erreurs absolues (entre le courant mesuré et celui-ci calculé) et la somme de courant mesuré correspondante. Le pourcentage d'erreur dans ces tables est défini comme suit:

$$\text{Pourcentage d'erreur} = \frac{\sum_{i=0}^n (I_m - I_s)^2}{\sum_{i=0}^n I_m^2} \times 100\%.$$

Le tableau 4.9 montre le résultat de cette étude. Cette table montre que l'augmentation de nombre de rotors dans le modèle de la machine améliore la précision des paramètres obtenus. Le table 4.10 donne une comparaison de l'erreur quadratique obtenue par le modèle à une cage et le modèle à deux cages pour les deux moteurs M1 et M2.

Tableau 4.9 Le Pourcentage d'erreur des deux modèles pour le moteur M1

Pourcentage erreur	Moteur M1	Moteur M2
Une Cage	5.55	14.76
Deux cages	2.27	13.82

Tableau 4.10 Comparaison entre le modèle à cage et à deux cages

	Une Cage	Deux cages	Taux d'amélioration
Moteur M1	45700	35606	22.10 %
Moteur M2	59000	49700	15.76 %

On remarque que le taux d'amélioration du moteur M1 est plus grand que le moteur M2, cela ne due que Le moteur M2 présent plus d'harmonique que le moteur M1.

4.4. Conclusion

Dans le présent chapitre, une technique d'identification paramétrique de la machine asynchrone est fondée sur la méthode du modèle de référence a été proposée. L'optimisation est réalisée à l'aide des algorithmes de luciole proposés au chapitre précédent. Les données simulées aussi bien que les données expérimentales de deux moteurs différents, montrent la supériorité des algorithmes proposés.

CONCLUSION GENERALE

Les travaux de recherche présentés dans cette thèse concernent le développement des nouvelles améliorations des méthodes d'optimisation globales qui s'appuient sur l'algorithme de luciole. Deux contributions sont proposées.

Premièrement, la contribution s'appuie sur l'orientation du mouvement la luciole vers le centroïde de l'essaim ce qui permet de contourner de problème de piégeage aux optimums locaux. C'est ce qui permet d'obtenir une version de l'algorithme de luciole ayant une bonne exploitation des zones prédéterminées dans l'espace de recherche.

Deuxièmement, la contribution sert à hybrider l'algorithme de luciole avec l'optimiseur de loup gris d'une part ou / et l'algorithme de sinus cosinus d'autre part. Tout d'abord, nous avons testé notre algorithme proposé par un jeu de fonctions de référence pour valider et pour prouver leur supériorité contre l'algorithme de luciole standard.

Puis, nous appliquons les algorithmes proposés sur un problème réel pour permettre de confirmer les améliorations obtenues de l'algorithme de luciole standard.

L'application traitée dans cette thèse rentre dans le cadre de l'identification paramétrique de la machine asynchrone. La méthode d'identification basée sur la méthode du modèle de référence est utilisée, et qui s'appuie sur les trois étapes suivantes :

- le choix du modèle de la machine;
- le choix des signaux d'entrée-sortie;
- le choix du critère d'évaluation, avec un premier choix de la méthode d'optimisation.

Dans un premier temps, nous avons adopté un modèle (système d'équations) de la machine asynchrone ayant quatre paramètres électriques mesurables. Dans un deuxième temps, pour tenir compte de l'effet de peau et pour affiner le modèle, nous avons assimilé le rotor à deux cages, ce qui

augmente les paramètres à identifier à sept paramètres électriques mesurables. Le modèle à double cage augmente la précision, mais il augmente aussi, la complexité du modèle. Les paramètres électriques et mécaniques sont déterminés simultanément à l'aide de la méthode du modèle de référence. Le courant du démarrage et la tension simple correspondante sont considérés comme des signaux d'entrées-sorties, tandis que l'erreur quadratique entre les courants mesurés et ceux calculés est choisi comme critère de similitude.

Nous avons testé la technique d'identification en utilisant l'algorithme de luciole standard et les algorithmes proposés, nous avons alors validé notre approche à l'aide des données simulées, puis des données expérimentales pour identifier différents paramètres des moteurs asynchrones. Nous avons vérifié que les algorithmes convergent vers le même vecteur des paramètres. Ce résultat nous permet donc de confirmer l'unicité de la solution.

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 encourageant à envisager son application pour le diagnostic préventif des machines asynchrones, comme perspectives de ce travail.

APPENDICE A
LISTE DES SYMBOLES ET DES ABREVIATIONS

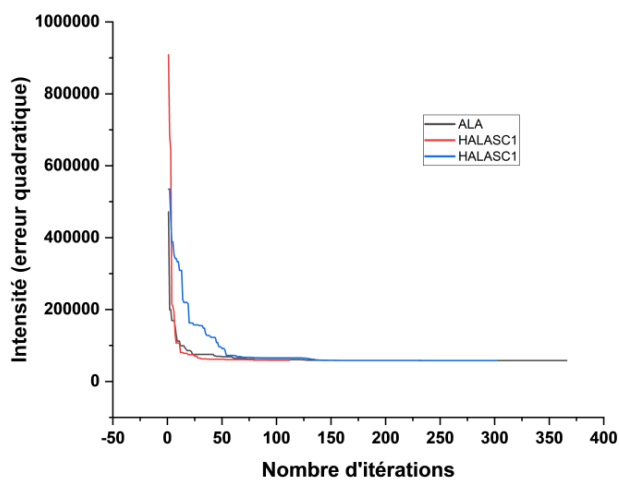
SYMBOLS

V_{dr}, V_{qr} :	Tension rotorique liée aux axes d-q
V_{ds}, V_{qs} :	Tension statorique liée aux axes d-q
I_{dr}, I_{qr} :	Courant rotorique lié aux axes d-q
I_{ds}, I_{qs} :	Courant statorique lié aux axes d-q
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
C_{em} :	Couple electromagnetique.
C_r :	Couple résistant.
J :	Moment d'inertie de l'ensemble des parties tournantes
B :	Coefficient de frottement.
σ :	Coefficient de dispersion
T_r :	Constante de temps rotorique
T_s :	Constante de temps statorique
$\varphi_{dr}, \varphi_{qr}$:	Flux rotorique
Ω :	Vitesse mecanique (Rd/s)
$\varphi_{ds}, \varphi_{qs}$:	Flux statorique
R_s, R_r :	Resistances statorique et rotorique
ω_e :	Vitesse electrique (Rd/s)
L_m :	Inductance mutuelle (H)
L_s, L_r :	Inductions statorique et rotorique (H)

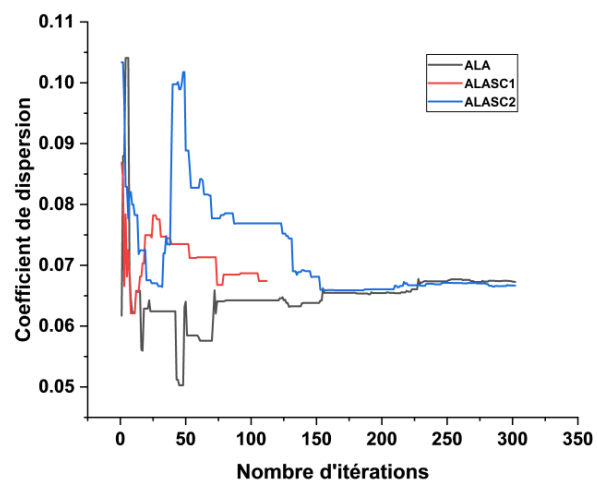
ABBREVIATIONS

- MAS : Machine asynchrone triphasée
- AL : Algorithme de luciole standard
- ALA : Algorithme de luciole amélioré
- OLG : Optimiseur de loup gris
- ASC : Algorithme d'optimisation sinus cosinus
- ALOLG : Algorithme de luciole hybride avec l'optimiseur de loup gris
- ALASC1 : Algorithme de luciole hybride avec l'algorithme de sinus cosinus type 1
- ALASC2 : Algorithme de luciole hybride avec l'algorithme de sinus cosinus type 2
- ALOLGASC1 : Algorithme de luciole hybride avec l'optimiseur de loup gris et l'algorithme de sinus cosinus type 1
- ALOLGASC2 : Algorithme de luciole hybride avec l'optimiseur de loup gris et l'algorithme de sinus cosinus type 2

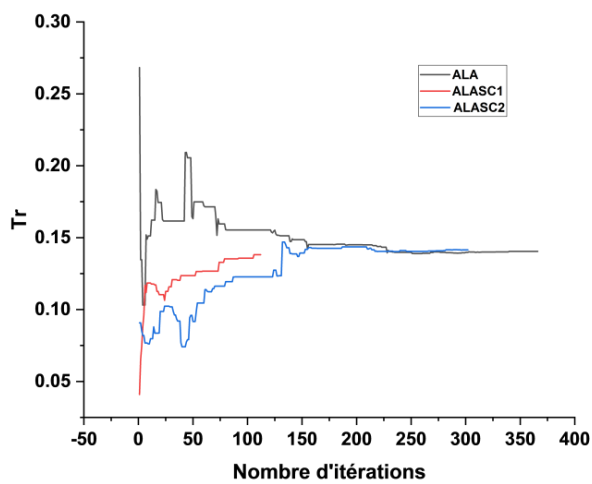
APPENDICE B
EVOLUTION DES PARAMETRES ESTIME DU MOTEUR M2 PAR
L'ALGORITHME DE LUCIOLE AMELIORE ET HYBRIDE (**MODELE A CAGE**)



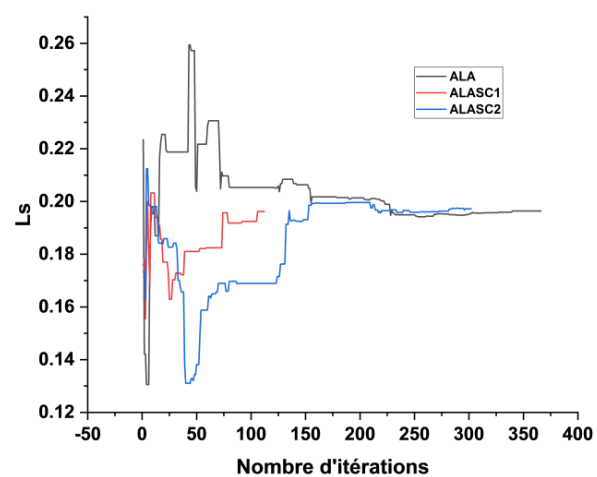
(a)



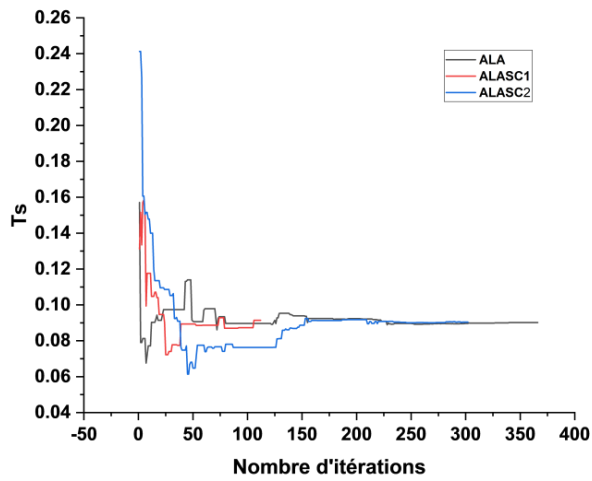
(b)



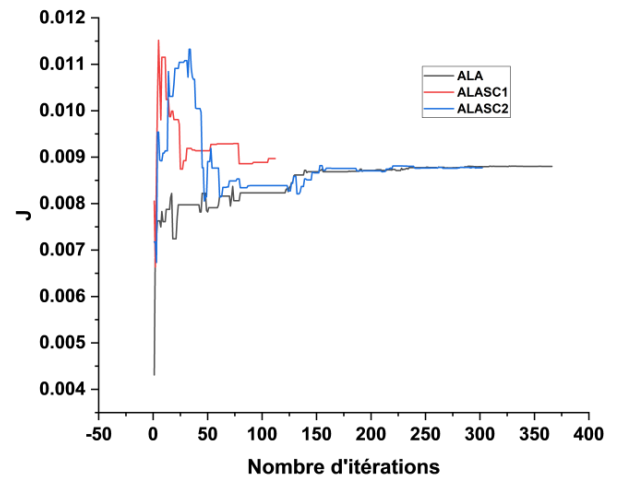
(c)



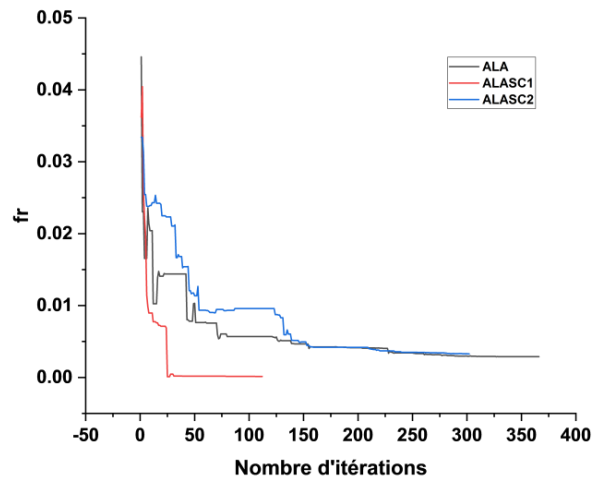
(d)



(e)



(f)



(g)

Figure B.1 evolution des paramètres estimés par : algorithme de luciole amélioré, algorithme de luciole hybride avec l'algorithme de sinus cosinus type 1 et type 2

APPENDICE C

SUPERPOSITION ET ERREUR ENTRE LE COURANT MESURE ET LE COURANT CALCULE PAR LES PARAMETRES ESTIME DU MOTEUR M2 PAR L'ALGORITHME DE LUCIOLE AMELIORE ET HYBRIDE (**MODELE A DOUBLE CAGE**)

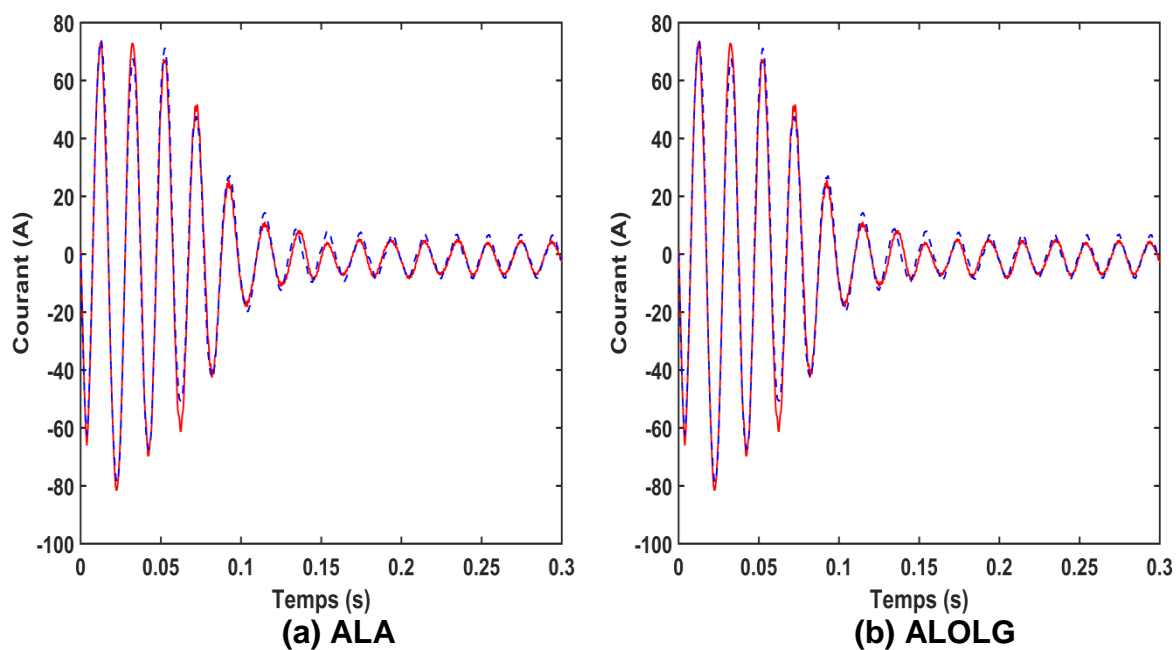


Figure C.1 Superposition du courant mesuré - - - et calculé - - - à l'aide des paramètres estimés par ALA et ALOGG du moteur M2

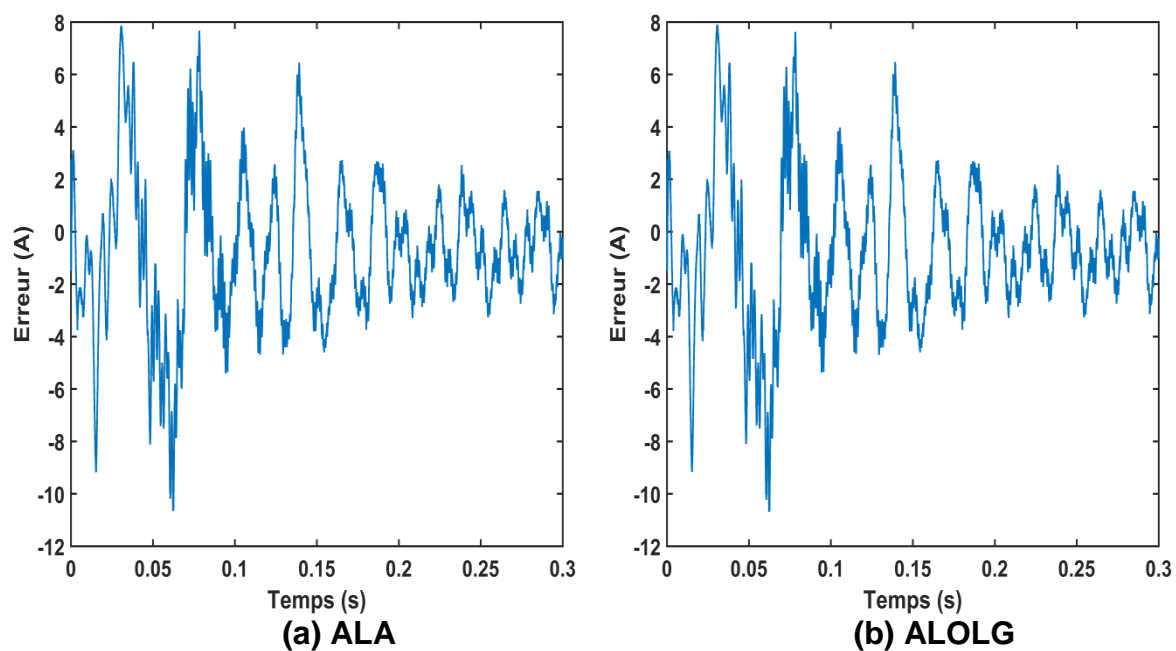


Figure C.2 Erreur entre le courant mesuré et celui calculé avec les paramètres estimés par ALA et ALOGG du moteur M2

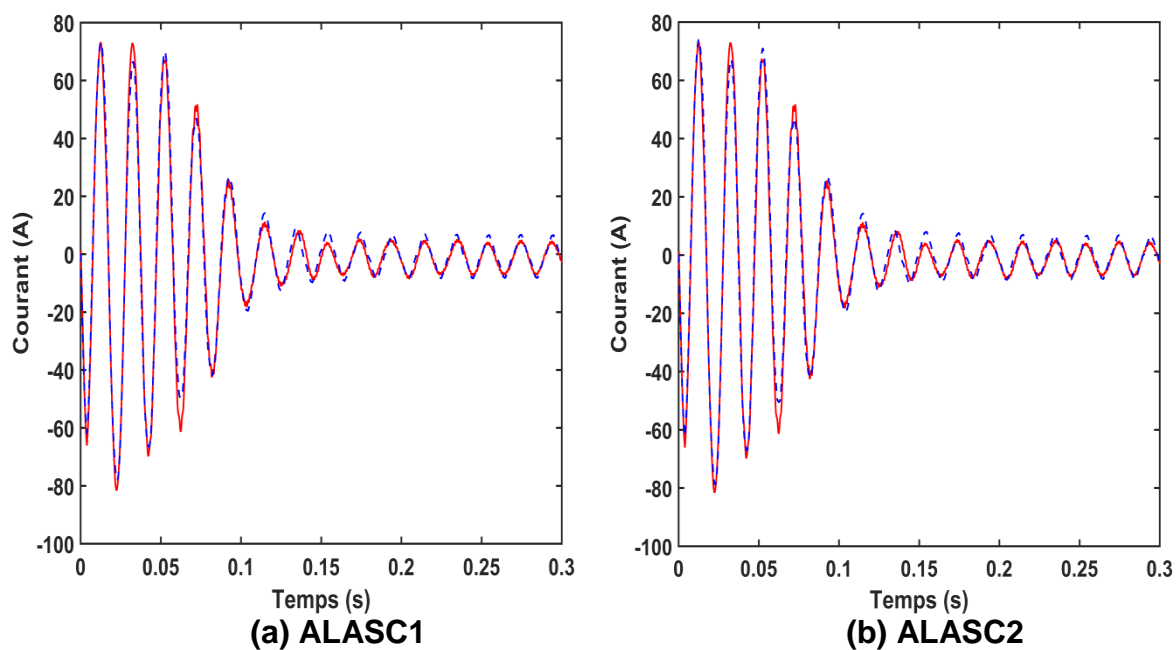


Figure C.3 Superposition du courant mesuré - - - et calculé - - - à l'aide des paramètres estimés par ALASC1 et ALASC2 du moteur M2

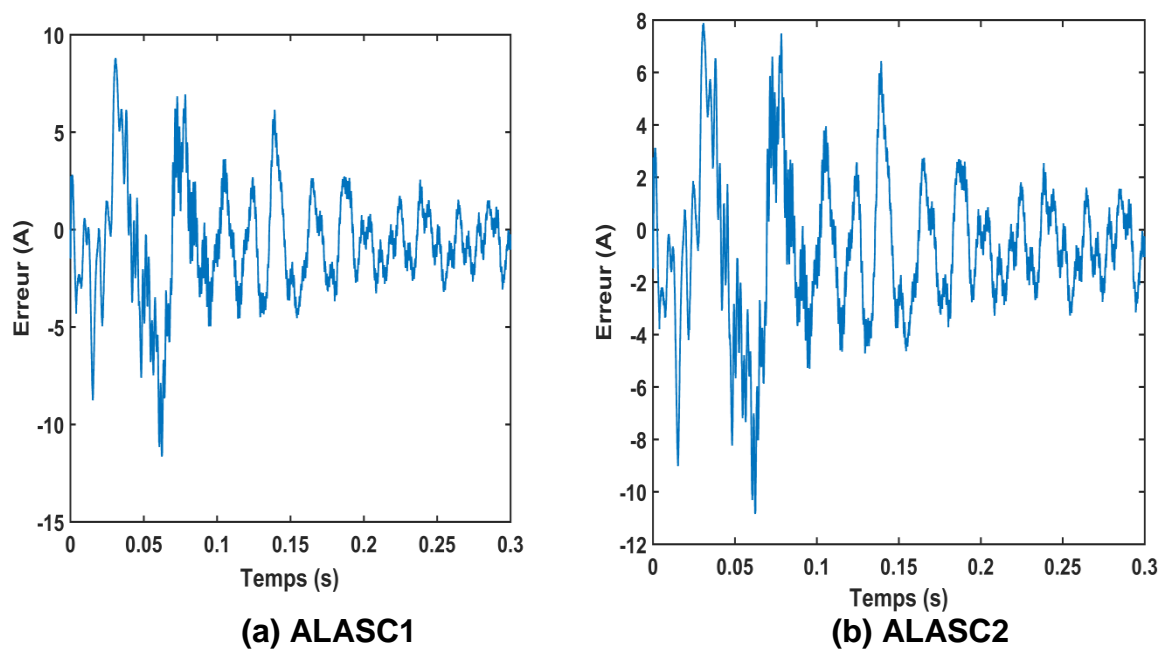


Figure C.4 Erreur entre le courant mesuré et celui calculé avec les paramètres estimés par ALASC1 et ALASC2 du moteur M2

REFERENCES

- [1] E. Walter, E. Walter, and L. Pronzato, *Identification de modèles paramétriques à partir de données expérimentales*. Paris: Masson, 1994.
- [2] Luenberger, D.G., "Linear and non linear programming.", Addison Wesley, 1984.
- [3] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Non linear programming theory and Algorithms*. 3rd edition, John Wiley & Sons inc, Sep, 2013.
- [4] E.-G. Talbi, "Metaheuristics : From Design to Implementation", Edition Wiley-Blackwell, 2009.
- [5] Talbi, EG. "A Taxonomy of Hybrid Metaheuristics". *Journal of Heuristics*, vol. 8, p. 541–564, 2002, <https://doi.org/10.1023/A:1016540724870>.
- [6] X. S. Yang, "Firefly algorithms for multimodal optimisation," in *Proc. 5th Symposium on Stochastic Algorithms, Foundations and Applications*, vol. 5792, pp. 169–178, 2009.
- [7] Ł. Szymon and Z. Sławomir, "Firefly algorithm for continuous constrained optimization tasks.", in *ICCCI*, 2009.
- [8] B. Computing, C. Science, and E. Sarajevo, "Firefly Algorithm for Constrained Optimization Problems," *Recent Res. Telecommun. Informatics, Electron. Signal Process. Firefly*, no. lii, pp. 139–144, 2013.
- [9] P. Moscato and A. M. Norman, "memetic approach for the traveling salesman problem: Implementation of a computational ecology for combinatorial optimization on message-passing systems," In *proceedings of the international conference on parallel computing and transputer applications*, 1992.
- [10] H. Ming-Huwi, "Vector quantization using the firefly algorithm for image compression", *Expert Syst Appl*, vol. 39, no. 1, pp. 1078–1091, 2012.
- [11] J. Senthilnath, S. N. Omkar, and V. Mani, "Clustering using firefly algorithm: performance study, *Swarm Evol*," *Comput*, vol. 1, no. 3, pp. 164–171, 2011.
- [12] X. S. Yang, S. S. H. Sayed, and H. G. Amir, "Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect," *Appl. Soft Comput*, vol. 12, 3, pp. 1180–1186, 2012.
- [13] S. L. Tilahun and H. C. Ong, "Modified firefly algorithm," *Journal of Applied*

- Mathematics, vol. 2012, Article ID 467631, 12 pages, 2012.
- [14] T. Hassanzadeh and M. R. Meybodi, "A new hybrid algorithm based on Firefly Algorithm and cellular learning automata," *20th Iranian Conference on Electrical Engineering (ICEE2012)*, pp. 628-633, 2012, doi: 10.1109/IranianCEE.2012.6292431.
- [15] M. Shafaati and H. Mojallali, "Modified Firefly Optimization for IIR System Identification," *J. Control Eng. Appl. Informatics*, vol. 14, no. 4, pp. 59–69, 2012.
- [16] S. M. Farahani, A. Abshouri, B. Nasiri, and M. Meybodi, "A Gaussian firefly algorithm," *Int. J. Mach. Learn. Comput.*, vol. 1, no. 5, pp. 448–454, 2011.
- [17] E. Boudissa and Bounekhla M., "Genetic Algorithm with Dynamic Selection Based on Quadratic Ranking Applied to Induction Machine Parameters Estimation," *Electr. Power Components Syst.*, vol. 40, no. 10, pp. 1089–1104, 2012.
- [18] B. M'hamed, O. Ahmed, T. Ali, E. m. Kadidja and B. Elghalia, "Using two PSO-structures approaches to estimate induction machine parameters," *2009 13th European Conference on Power Electronics and Applications*, pp. 1-8, 2009.
- [19] M. A. Awadallah, "Parameter estimation of induction machines from nameplate data using particle swarm optimization and genetic algorithm techniques," *Electr. Power Components Syst.*, vol. 36, no. 8, pp. 801–814, Aug. 2008, doi: 10.1080/15325000801911393.
- [20] Vas, P., "Artificial-Intelligent-Based Electrical Machines and Drives. Application of Fuzzy, Neural, Fuzzy-Neural and Genetic-Algorithm-Based Techniques", Oxford Univ. Press, Oxford, 1999.
- [21] CARON Jean-Pierre and HAUTIER Jean-Paul, *Modélisation et commande de la machine asynchrone* - CARON Jean-Pierre - HAUTIER Jean-Paul, TECHNIP. 1995.
- [22] P. L. Alger, "Induction Machines", Gordon and Broach Science Publishers, Second Edition, 1970.
- [23] J. Chatelain, "Machines électriques: Traité d'électricité", Volume X, Presses Ecole Polytechnique Romandes, Lausanne, 1983.
- [24] A. Gastli, "Identification of induction motor equivalent circuit parameters using the single-phase test," *IEEE Trans. Energy Convers.*, vol. 14, no. 1,

- pp. 51–56, 1999, doi: 10.1109/60.749147.
- [25] A. Trentin, P. Zanchettan, P. Wheeler, J. Clare, R. Wood, and D. Katsis, “A New Methods for Induction Motors parameters Estimation using GA and Transient speed Measurements”, in Industry application conference, 41st IAS annual meeting, Conference record of the 2006 IEEE, vol. 5, pp. 2435–2440, 2006.
- [26] B. Abdelhadi, A. Benoudjit, and N. Nait-Said, “Application of genetic algorithm with a novel adaptive scheme for the identification of induction machine parameters,” *IEEE Trans. Energy Convers.*, vol. 20, no. 2, pp. 284–291, Jun. 2005, doi: 10.1109/TEC.2004.841508.
- [27] M. Bounekhla and M. E. Zaim, Rezzoug, A., “Identification of saturated induction motor parameters from transient stator current measurement”. Toulouse, EPE, 2003.
- [28] M. Bounekhla, M. E. Zaim, and A. Rezzoug, “Comparative study of three minimization methods applied to the induction machine parameters dentification using transient stator current”, *Electr. Power Components Syst*, vol. 33, no. 8, pp. 913–930, 2005.
- [29] N. Benaïdja and N. Khenfer, “Identification of asynchronous machine parameters by evolutionary techniques,” *Electr. Power Components Syst.*, vol. 34, no. 12, pp. 1359–1376, Dec. 2006, doi: 10.1080/15325000600748897.
- [30] IEE, “Test Procedure for Polyphase Induction Motors and Generators.” IEEE, New York City, NJ, USA.
- [31] IEC Standards 60034-28, Rotating Electrical Machines–Part 28: Test Methods for Determining Quantities of Equivalent Circuit Diagrams for Three-Phase Low-Voltage Cage Induction Motors. IEC: Geneva.
- [32] J. Pedra and L. Sainz, “Parameter estimation of squirrel-cage induction motors without torque measurements,” *IEE Proc. - Electr. Power Appl.*, vol. 153, no. 2, p. 263, 2006, doi: 10.1049/ip-epa:20050383.
- [33] E. G. Talbi, “A taxonomy of hybrid meta-heuristics,” Rapport AS-183 du Laboratoire d’Informatique Fondamentale de Lille. Université des Sciences et Technologies de Lille, *Journal of combinatorial optimisation*, 1-45, 1999.
- [34] M. Gomez-Gonzalez, F. Jurado, and I. Pérez, “Shuffled frog-leaping algorithm for parameter estimation of a double-cage asynchronous

- machine,” *IET Electr. Power Appl.*, vol. 6, no. 8, p. 484, 2012, doi: 10.1049/iet-epa.2011.0262.
- [35] H. R. Mohammadi and A. Akhavan, “Parameter Estimation of Three-Phase Induction Motor Using Hybrid of Genetic Algorithm and Particle Swarm Optimization,” *Journal of Engineering*, vol. 2014, Article ID 148204, 6 pages 20143. doi: 10.1155/2014/148204.
- [36] L. A. Pereira, M. Perin, L. F. A. Pereira, J. R. Ruthes, F. L. M. de Sousa, and E. C. P. de Oliveira, “Performance estimation of three-phase induction motors from no-load startup test without speed acquisition,” *ISA Trans.*, vol. 96, pp. 376–389, Jan. 2020, doi: 10.1016/j.isatra.2019.05.028.
- [37] M. Akbaba, M. Taleb, and A. Rumeli, “Improved estimation of induction machine parameters,” *Electr. Power Syst. Res.*, vol. 34, no. 1, pp. 65–73, Jul. 1995, doi: 10.1016/0378-7796(95)00958-K.
- [38] I. Kostov, V. Spasov, and V. Rangelova, “Application of genetic algorithms for determining the parameters of induction motors,” *Teh. Vjesn.*, vol. 16, no. 2, pp. 49–53, 2009.
- [39] B. Abdelhadi, A. Benoudjit, and N. Nait Said, “Identification of induction machine parameters using a new adaptive genetic algorithm,” *Electr. Power Components Syst.*, vol. 32, no. 8, pp. 767–784, Aug. 2004, doi: 10.1080/15325000490466645.
- [40] L. Monjo, F. Córcoles, and J. Pedra, “Parameter estimation of squirrel-cage motors with parasitic torques in the torque-slip curve,” *IET Electr. Power Appl.*, vol. 9, no. 5, pp. 377–387, May 2015, doi: 10.1049/iet-epa.2014.0208.
- [41] S. Yamamoto, H. Hirahara, A. Tanaka, and T. Ara, “A Simple Method to Determine Double-Cage Rotor Equivalent Circuit Parameters of Induction Motors From No-Load and Locked-Rotor Tests,” *IEEE Trans. Ind. Appl.*, vol. 55, no. 1, pp. 273–282, 2019, doi: 10.1109/TIA.2018.2864105.
- [42] R. Natarajan and V. K. Misra, “Parameter estimation of induction motors using a spreadsheet program on a personal computer,” *Electr. Power Syst. Res.*, vol. 16, no. 2, pp. 157–164, Mar. 1989, doi: 10.1016/0378-7796(89)90008-4.
- [43] M. H. Haque, “Determination of NEMA design induction motor parameters from manufacturer data,” *IEEE Trans. Energy Convers.*, vol. 23, no. 4, pp. 997–1004, 2008, doi: 10.1109/TEC.2008.2001451.

- [44] Minoux, M., "Programmation Mathématiques: Théorie et algorithmes.", Edition Dunod, Paris, 1983.
- [45] S. Tahi, R. Ibtiouane, and Bounekhla, "M.," "Design optimization of two synchronous reluctance machine with maximized torque and power factor", Progress In Electromagnetics Research B, vol. 35, pp. 369–387, 2011. Doi:10.2528/PIERB11091101
- [46] M. E. Zaim, K. Dakhouché and M. Bounekhla, "Design for torque ripple reduction of a three-phase switched-reluctance machine," in IEEE Transactions on Magnetics, vol. 38, no. 2, pp. 1189-1192, March 2002, doi: 10.1109/20.996304.
- [47] M. Bounekhla, M. E. Zaim, A. Rezzoug, and A. Ahmed, M., "The Hooke & Jeeves algorithm approach in the identification of the induction machine parameters", ICEM, Helsinki, 2000.
- [48] M. Bounekhla, M. E. Zaim, A. Rezzoug and M. Ait-Ahmed, "Application of Rosenbrock algorithm to motor parameters identification using transient stator current measurements," 2000 Eighth International Conference on Power Electronics and Variable Speed Drives (IEE Conf. Publ. No. 475), pp. 158-162, 2000. Doi: 10.1049/cp: 20000238.
- [49] R. Thangaraj, M. Pant, C. Thanga Raj, and A. K. Nagar, "In Situ efficiency determination of induction motor: a comparative study of evolutionary techniques", Appl. Artif. Intell., vol. 25, no. 2, pp. 116–140, 2011.
- [50] M. Srinivas, M. Patnaik, "Genetic Algorithm: A Survey," IEEE Comput., vol. 27, pp. 17–26, Juin 2014.
- [51] Renders, J. M., "Algorithmes génétiques et réseaux de neurone". Edition Hermès, 1995.
- [52] Colette, Y., Siarry, P, "Optimisation multiobjectif". Edition Eyrolles, 2000.
- [53] T. A. Feo and M. G. C. Resende, "A probabilistic heuristic for a computationally difficult set covering problem," Oper. Res. Lett., vol. 8, no. 2, pp. 67–71, 1989.
- [54] T. A. Feo and M. G. C. Resende, "Greedy randomized adaptive search procedures," J. Glob. Optim., vol. 6, no. 2, pp. 109–133, 1995.
- [55] J. P. Hart and A. W. Shogan, "Semi-greedy heuristics : An empirical study," Oper. Res. Lett., vol. 6, no. 3, pp. 107–114, 1989.
- [56] P. Festa and M. Resende, "An annotated bibliography of GRASP, Part I :

- Algorithms," *Int. Trans. Oper. Res.*, vol. 16, no. 1, pp. 1–24, 2009.
- [57] P. Festa and M. Resende, "An annotated bibliography of GRASP, Part II: Applications," *Int. Trans. Oper. Res.*, vol. 16, no. 2, p. 131 172, 2009.
- [58] Mladenovic, N., "A variable neighborhood algorithm - a new metaheuristic for combinatorial optimization," in *Abstracts of papers presented at Optimization Days, Montréal, Canada*, p. 112, 1999.
- [59] N. Mladenovic, "A variable neighborhood algorithm," *Journal of Computers & Operations Research*, vol. 24, no. 11, p. 1097-1100, 1997.
- [60] Stützle, Thomas. "Local search algorithms for combinatorial problems - analysis, improvements, and new applications." *DISKI* (1999).
- [61] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol.4, pp. 1942-1948,1995, doi: 10.1109/ICNN.1995.488968.
- [62] D. M., M. V., and C. A, "The Ant System: Optimization by a Colony of Cooperating Agents," *IEEE Trans. Syst. Man, Cybern. - Part B*, vol. 26, no. 1, pp. 29–41, 1996.
- [63] D. M. and Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, 1997.
- [64] M. Dorigo, "The Ant Colony Optimization Meta-Heuristic", In D. Corne, M. Dorigo, F. Glover (eds) *New Ideas in Optimization*, McGraw-Hill, London, UK, pp 11–32, 1999.
- [65] M. Hadj Sadok, H. Salhi, A. Guessoum, and Bounekhla, M., "Induction machine parametric identification by the ants *pachycondyla apicalis* algorithm", *Mediterr. J. Meas. Control*, vol. 3, no. 2, pp.49-58, 2007.
- [66] Yang, X.S., "Nature-inspired Metaheuristic Algorithms", UK, Luniver Press, 2008.
- [67] X. S. Yang, "Firefly algorithms for multimodal optimisation," in *Proc. 5th Symposium on Stochastic Algorithms, Foundations and Applications*, vol. 5792, pp. 169–178, 2009.
- [68] X. S. Yang, *Engineering Optimisation: An Introduction with Metaheuristic Applications*. USA: John Wiley and Sons, Jun 2010.
- [69] Yang, X. S., "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimisation (NICSO 2010)*, vol. 284, Edition.

- Springer, Berlin, SCI, pp. 65–74, 2010.
- [70] S. Mirjalili, S. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.
- [71] S. Mirjalili, "A Sine Cosine Algorithm for solving optimization problems", *Knowledge-Based Syst.*, vol. 96, pp. 120–133, 2016. doi: /10.1016/S0305-0548(97)00031-2.
- [72] W. C. Jackson and J. D. Norgard, "A Hybrid Genetic Algorithm with Boltzmann Convergence Properties," *J Optim Theory Appl*, vol. 136, no. 3, pp. 431–443, 2008, doi: /10.1007/s10957-007-9308-8
- [73] Y. G. Xu, G. R. Li, and Z. P. Wu, "A Novel hybrid genetic algorithm using local optimizer based on heuristic pattern move," *Appl. Artif. Intell.*, vol. 15, no. 7, pp. 601– 631, 2001.
- [74] O. A. Mohammed and G. F. Uler, "A hybrid technique for the optimal design of electromagnetic devices using direct search and genetic algorithms," in *IEEE Transactions on Magnetics*, vol. 33, no. 2, pp. 1931-1934, March 1997, doi: 10.1109/20.582670.
- [75] E. G. Talbi, "Hybrid metaheuristics for multi-objectif optimization," *J. Algorithms Comput. Technol.*, vol. 9, no. 1, 2015, doi: 10.1260/1748-3018.9.1.4
- [76] X. S. Yang, "Firefly algorithm, stochastic test functions and design optimization," *Int. J. Bio Inspired Comput.*, vol. 2, no. 2, pp. 78–84, 2010.
- [77] F., Jr. Iztok, X.S., Yang, F., Iztok, J., Brest. "Memetic firefly algorithm for combinatorial optimization." *arXiv: Optimization and Control*, 2012.
- [78] E. Boudissa and Bounekhla, "M.," "A real-coded genetic algorithm applied to induction machine parameters identification trough an output error", *Mediterr. J. Meas. Control*, vol. 6, no. 3, p. 109 119, 2010.
- [79] Jamadi, Mohammad and Farshad Merrikh- Bayat. "New Method for Accurate Parameter Estimation of Induction Motors Based on Artificial Bee Colony Algorithm." *ArXiv abs/1402.4423*, n. pag, 2014.
- [80] V. P. Sakthivel, R. Bhuvaneshwari, and S. Subramanian, "Bacterial foraging technique based parameter estimation of induction motor from manufacturer data, *Elect. Power Compon.*," *Syst*, vol. 38, pp. 657–674, 2010.
- [81] S. Ansuji, F. Shokooch and R. Schinzingler, "Parameter estimation for induction machines based on sensitivity analysis," in *IEEE Transactions on*

- Industry Applications, vol. 25, no. 6, pp. 1035-1040, Nov.-Dec. 1989, doi: 10.1109/28.44239.
- [82] Y. Koubaa, "Asynchronous machine parameters estimation using recursive method," *Simul. Model. Pract. Theory*, vol. 14, no. 7, pp. 1010–1021, 2006.
- [83] D. Lindenmeyer, H. W. Dommel, A. Moshref, and P. Kundur, "An induction motor parameter estimation method," *Int. J. Electr. Power Energy Syst.*, vol. 23, pp. 251–262, 2001.
- [84] J. Stephan, M. Bodson, and J. Chiasson, "Real time estimation of induction motor parameters," *IEEE Trans. Industry Appl.*, vol. 30, 3, pp. 746–759, 1994.
- [85] Trentin, A., Zanchetta, P., Wheeler, P.W., Clare, J.C., Wood, R., & Katsis, D., "A New Method for Induction Motors Parameter Estimation Using Genetic Algorithms and Transient Speed measurements", in *Conference Record of the 2006 IEEE Industry Applications Conference Forty-First IAS Annual Meeting*, 5, 2435-2440, 2006.
- [86] A. A. Mohamed, "Parameter estimation of Induction Machines from nameplate data using particle swarm optimisation and genetic algorithm techniques", *Electr. Power components Syst.*, vol. 36, pp. 801–814, 2008.
- [87] N. Dif, E. Boudissa, M. Bounekhla, I. Dif, "Firefly algorithm improvement with application to induction machine," *Revue roumaine des sciences techniques -serie electrotechnique et energetique*, vol. 65, no.1, pp. 35-40, 2020.
- [88] N. Dif, E. Boudissa, M. Bounekhla, "Determination of single cage model parameters of induction machine using a basic firefly and their variants," in *Conference ICAID 2017, The Second International Conference on Applied Automation and Industrial Diagnostics*, 2017.