

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Ecole Nationale Polytechnique
Département d'Electronique

Laboratoire des Dispositifs de Communication
et de Conversion Photovoltaïque



Thèse de Doctorat

En Électronique
Option : Électricité Solaire

Présentée par :

Sabrina TITRI

Master Systèmes de Communication et Réseaux

Thème

Circuits reconfigurables pour la gestion de l'énergie dans les systèmes photovoltaïques

Membres du jury :

- | | |
|------------------------------------|----------------------------------|
| - M. HADDADI Mourad, Pr. | Président, ENP |
| - M. LARBES Chérif, Pr. | Directeur de thèse, ENP |
| - M. YOUCEF-TOUMI Kamel, Pr. | Co-Directeur de thèse, MIT (USA) |
| - M. AIT CHEIKH Mohamed Salah, Pr. | Examineur, ENP |
| - M. BOUCHAFA Farid, Pr. | Examineur, USTHB |
| - M. MAHRANE Achour, DR | Examineur, UDES |

ENP, 2018

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Ecole Nationale Polytechnique
Département d'Electronique

Laboratoire des Dispositifs de Communication
et de Conversion Photovoltaïque



Thèse de Doctorat

En Électronique
Option : Électricité Solaire

Présentée par :

Sabrina TITRI

Master Systèmes de Communication et Réseaux

Thème

Circuits reconfigurables pour la gestion de l'énergie dans les systèmes photovoltaïques

Membres du jury :

- | | |
|------------------------------------|----------------------------------|
| - M. HADDADI Mourad, Pr. | Président, ENP |
| - M. LARBES Chérif, Pr. | Directeur de thèse, ENP |
| - M. YOUCEF-TOUMI Kamel, Pr. | Co-Directeur de thèse, MIT (USA) |
| - M. AIT CHEIKH Mohamed Salah, Pr. | Examineur, ENP |
| - M. BOUCHAFA Farid, Pr. | Examineur, USTHB |
| - M. MAHRANE Achour, DR | Examineur, UDES |

ENP, 2018

Remerciements

Cette thèse est le fruit d'un long travail de recherche, qui j'en suis convaincue, n'est qu'un modeste pavé dans la longue quête de parfaire encore et toujours les résultats. Je profite donc de l'occasion qui m'est donnée ici pour exprimer toute ma reconnaissance et ma gratitude envers les nombreuses personnes qui m'ont permis de progresser dans cette phase délicate et de finaliser ces travaux.

Les travaux de recherche présentés dans cette thèse ont été menés conjointement entre le *Laboratoire de Dispositifs de Communication et de Conversion Photovoltaïques (LDCCP)* de l'Ecole Nationale Polytechnique (ENP), et la *Division Microélectronique et Nanotechnologies (DMN)* du Centre de Développement des Technologies Avancées (CDTA).

En premier lieu, je tiens à remercier chaleureusement Monsieur ***Cherif LARBES***, Directeur de thèse, Professeur à l'Ecole Nationale Polytechnique, pour la confiance qu'il m'a accordée en acceptant la direction de ma thèse. Je tiens à le remercier plus particulièrement pour sa compétence, son expérience pour avoir assuré de manière agréable le suivi de mon travail, tout au long de ces années de thèse, pour ses nombreux conseils qui m'ont beaucoup aidé et m'ont orienté vers une recherche de qualité. Il a su apporter sa rigueur scientifique et sa maîtrise académique pour la réussite de ce travail de recherche. Qu'il trouve ici toute ma reconnaissance.

Aussi, je tiens à exprimer toute ma gratitude à Monsieur ***Kamal YOUCEF TOUMI***, Professeur au Massachusetts Institute of Technology (MIT/USA), co-directeur de thèse, en le remerciant pour son aide précieuse au travers de ses conseils éclairés et orientations justes, ses suggestions pertinentes et surtout pour tous les échanges scientifiques et les discussions qu'on a pu avoir. Je ne le remercierai jamais assez pour sa disponibilité malgré les lourdes charges de travail auxquelles il est assujéti, tant académiques que professionnelles, tout cela, conjugué aux difficultés liées aux décalages horaires. Qu'il trouve ici également mon humble témoignage quant à sa patience, sa gentillesse ainsi que ses qualités humaines exceptionnelles et inégalées.

Je suis très sensible à l'honneur que m'a fait Monsieur ***Mourad HADDADI***, Professeur à l'Ecole Nationale Polytechnique ENP d'avoir accepté de présider le jury de ma soutenance. Aussi, je tiens à lui exprimer toute ma reconnaissance pour l'intérêt porté à ce modeste travail.

Remerciements

Je remercie aussi vivement les membres du jury : Monsieur **Mohamed Salah AIT CHEIKH**, Professeur à l'Ecole Nationale Polytechnique (ENP), Monsieur **Farid BOUCHAFA**, Professeur à l'Université des Sciences et Technologies Houari Boumèdiene (USTHB) et Monsieur **Achour MAHRANE** Directeur de Recherche à l'Unité de Développement des Equipements Solaires (UDES) pour l'intérêt qu'ils ont porté à mon travail en acceptant de l'examiner et l'honneur qu'ils m'ont fait en participant à ce jury.

Mes sincères remerciements vont à M^{me} **Nouma Izeboudjen**, chef d'équipe *MICS* au sein de la Division Microélectronique et Nanotechnologie pour son aide, ses conseils, ses orientations, ses remarques, ses critiques constructives et ses encouragements en permanence tout au long de ce parcours. Qu'elle trouve ici l'expression de ma profonde gratitude et reconnaissance.

Un remerciement particulier : à M^{me} **Karima Benatchba**, Professeur à l'Ecole Nationale Supérieure d'Informatique (ENSI), pour ses orientations et sa précieuse aide, particulièrement celle liée au domaine des métagheuristiques et du bio inspirée ; à Mme **Fathya CHEKIRED**, chercheur au niveau de l'Unité de Développement des Equipements Solaires (UDES), pour son aide et ses orientations.

Enfin, je réserve une place particulière à ma **famille** en remerciant chaleureusement :

Mes chers parents sans qui je n'en serais pas là aujourd'hui et à qui je dois tout. Je les remercie pour leur soutien indéfectible et leurs encouragements tout au long de mes études. Qu'ils trouvent, dans ce travail, le témoignage de ma fierté et de ma gratitude la plus sincère ; **Mon mari**, pour la confiance qu'il m'a témoignée, ses perpétuels encouragements, sa compréhension, sa patience légendaire supportant, chaque jour, mes états d'âme. Je tiens à le remercier particulièrement pour son soutien moral ininterrompu, principal levier de ressourcement et de motivation pour le parachèvement de ce travail. Enfin, je le remercie également pour ses précieux conseils à la relecture de mon manuscrit facilitant la finalisation de cette thèse qui a sans doute évité quelques crises de nerfs de dernière minute ;

Mon fils, ma joie de vivre, pour sa compréhension...

Enfin, à toutes les personnes qui ont contribué de près ou de loin, directement ou indirectement, à l'aboutissement de ce travail, qu'elles trouvent ici le témoignage de ma profonde reconnaissance.

Sabrina TITRI

ملخص:

في هذا العمل اقترحنا أولاً تصنيف جديد للتحكم في تتبع نقطة الاستطاعة القسوى "MPPT" اعتماداً على المعايير والأساليب التي يعتبرها المصمم. بالتالي تم اقتراح وحدات تحكم جديدة على أساس أساليب مستوحاة من البيولوجيا "MBI" وعلى وجه التحديد خوارزمية النمل (ACO) وخوارزمية الخفافيش (BSO). أولاً تم تطوير مختلف خطوات التصميم المقترحة حسب نمط ACO_MPU_MPPT وBSO_MPPT المقترحة بتقييم أداء وحدات التحكم التي تم الحصول عليها من حيث سرعة التتبع والدقة والاستقرار والمتانة وتم إجراء الاختبارات في ظل تباينات بطيئة وسريعة للظروف الجوية (درجة الحرارة والإشعاع الشمسي) تحت مختلف أنماط التضليل الجزئي. أظهرت النتائج التي تم الوصول إليها أن وحدات التحكم المقترحة تعطي أفضل أداء في ظل الظروف الجوية المتغيرة. بالإضافة إلى ذلك، فإنها تمكن تتبع نقطة الاستطاعة القسوى بسهولة. علاوة على ذلك، لإثبات تفوق ومثالة وحدات التحكم المقترحة تم تحليل النتائج التي تم الوصول إليها ومقارنتها مع الآخرين التي تم الحصول عليها بالطرق التقليدية (P&O_MPPT) وطرق الحوسبة الناعمة (ANN_MPPT، FLC_MPPT، FL_GA_MPPT، ANFIS_MPPT). تم تنفيذ وحدات التحكم (BIM_MPPT) على دارة FPGA وقد تحقق ذلك من خلال اعتماد منهج تصميم جديد يعتمد على أساس (HLS). ثم من أجل تسهيل النماذج الأولية السريعة لمختلف وحدات التحكم، تم اقتراح منصتين: الأولى مكرسة بتصميم وتقييم أداء وحدات التحكم MPPT من خلال اختبارات ومحاكات. أما المنصة الثانية فهي مصممة للنماذج الأولية IP_MPPT المتخصصة لتنفيذ نظام ضوئي.

كلمات مفتاحية: أنظمة شمسية، نقطة تتبع الاستطاعة القسوى، أساليب مستوحاة من البيولوجيا، خوارزمية النمل والخفافيش، FPGA، HLS.

Abstract:

In this work, we first proposed a new classification of MPPT controllers depending on the criteria and methods considered by the designer. Then, new MPPT controllers based on Bio- inspired methods (BIM), precisely the Ant Colony Optimization algorithm and Bat Swarm Optimization Algorithm (BSO MPPT) are proposed. First, the different steps of the design of the proposed ACO_NPU MPPT and BSO_MPPT controllers are developed. To evaluate the performances of the obtained controllers, in terms of its tracking speed, accuracy, stability and robustness, tests are carried out under slow and rapid variations of weather conditions (Irradiance and Temperature) and under different partial shading patterns. Moreover, to demonstrate the superiority and robustness of the proposed controllers, the obtained results are analyzed and compared with others obtained from the Conventional Methods (P&O_MPPT) and the Soft Computing Methods with Artificial intelligence (ANN_MPPT, FLC_MPPT, ANFIS_MPPT, FL_GA_MPPT). The obtained results show that the proposed controllers give the best performances under variables atmospheric conditions. In addition, they can easily track the global maximum power point (GMPP) under partial shading conditions. Secondly, a hardware implementation of the Bio inspired MPPT controllers on FPGA circuit is achieved by adopting a new design approach based on the high level synthesis (HLS) concept. Then, in order to facilitate the rapid prototyping of various MPPT controllers, two platforms are proposed, the first one, dedicated to develop and evaluation the performance of MPPT controllers through tests and simulations. The second one is designed for Rapid prototyping of IP_MPPT controllers dedicated for hardware implementation of photovoltaic systems.

Keywords: Photovoltaic systems, MPPT, bio inspired methods, ACO_NPU, BSO, FPGA, HLS.

Résumé:

Dans ce travail, nous avons en premier lieu proposé une nouvelle classification des contrôleurs MPPT selon les critères et les méthodes envisagées par le concepteur. Puis, de nouveaux contrôleurs basé sur les méthodes Bio Inspirés (MBI), plus précisément l'algorithme de colonie de fourmis (ACO_MPPT) et l'algorithme de chauve-souris (BSO_MPPT) sont proposés. En premier, les différentes étapes de conception des deux contrôleurs proposés ACO_NPU_MPPT et BSO_MPPT sont développées. Afin d'évaluer les performances des deux contrôleurs en terme de vitesse de convergence, précision, stabilité et robustesse, des tests sont élaborés sous variations de température et d'éclairement (rapide, lente) et tenant compte de l'ombrage partiel. De plus, afin de démontrer la supériorité et la robustesse des contrôleurs proposés, les résultats obtenus sont analysés et comparés avec les contrôleurs (P&O_MPPT) des Méthodes Conventionnelles, et (ANN_MPPT, FLC_MPPT, ANFIS_MPPT, FL_GA_MPPT) des méthodes d'Intelligence Artificielles. Les résultats obtenus montrent que les contrôleurs proposés donnent de meilleures performances sous variation des conditions atmosphériques. De plus, sous un ombrage partiel, ils peuvent suivre le GMPP facilement. En second lieu, une implémentation hardware des contrôleurs Bio Inspirés sur circuits FPGA est réalisée en adoptant le concept de l'approche de conception basée sur la synthèse de haut niveau. Aussi, afin de faciliter le prototypage rapide des contrôleurs MPPT, deux plateformes sont proposées, la première, dédiée pour le développement et l'évaluation des performances des contrôleurs MPPT au travers de tests et simulations. La seconde quant à elle est conçue pour le prototypage rapide d'IP_MPPT.

Mots-clés : systèmes photovoltaïques, MPPT, méthodes bio inspirées, ACO_NPU, BSO, FPGA, HLS.

TABLE DES MATIÈRES

LISTE DES ABRÉVIATIONS

LISTE DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION GÉNÉRALE	22
PARTIE I : LES TECHNIQUES DE RECHERCHE DU POINT DE PUISSANCE MAXIMALE	
CHAPITRE 1 : LES CONCEPTS DE BASE DES SYSTÈMES PHOTOVOLTAÏQUES	30
1.1 INTRODUCTION	30
1.2 L'ÉFFET PHOTOVOLTAÏQUE	30
1.2.1 Principe d'une cellule photovoltaïque	30
1.2.2 Caractéristique électrique d'une cellule photovoltaïque	32
1.2.3 Modélisation d'une cellule photovoltaïque	33
1.3 LE GÉNÉRATEUR PHOTOVOLTAÏQUE	34
1.3.1 Structure d'un générateur photovoltaïque	34
1.3.2 Caractéristique Courant-Tension et Puissance-Tension	35
1.3.2.1 Influence de l'éclairement sur les courbes I-V et P-V	35
1.3.2.2 Influence de la température sur les courbes I-V et P-V	36
1.3.2.3 Influence d'un éclairement non homogène : Ombrage partiel	37
1.4 CONNEXION GÉNÉRATEUR PHOTOVOLTAÏQUE-CHARGE	39
1.4.1 Connexion directe	39
1.4.2 Connexion indirecte via un étage d'adaptation	39
1.5 PRINCIPE DE RECHERCHE DU POINT DE PUISSANCE MAXIMALE	40
1.6 CONCLUSION	41
CHAPITRE 2 : LES MÉTHODES DE RECHERCHE DU POINT DE PUISSANCE MAXIMALE MPPT : ÉTAT DE L'ART ET PERSPECTIVES	44
2.1 INTRODUCTION	44
2.2 LES MÉTHODES DE RECHERCHE DU POINT DE PUISSANCE MAXIMALE MPPT	44
2.2.1 Critères de sélection des commandes MPPT	44
2.2.2 Classification des méthodes MPPT	47
2.3 PROPOSITION D'UNE NOUVELLE APPROCHE DE CLASSIFICATION DES CONTRÔEURS MPPT	50

TABLE DES MATIÈRES

2.4 LES MÉTHODES CONVENTIONNELLES (MC)	51
2.4.1 Les méthodes indirectes (MI)	51
2.4.1.1 Méthodes à contre réaction de tension	52
2.4.1.1.1 Méthode à tension de référence fixe	52
2.4.1.1.2 Méthode à tension de référence en fonction de V_{co}	52
2.4.1.1.3 Méthode à tension de référence externe (cellule pilote)	53
2.4.1.2 Méthodes à contre réaction de courant	54
2.4.1.2.1 Méthode à courant de référence en fonction du courant du court-circuit I_{sc}	54
2.4.1.2.2 Méthode à courant de référence externe (cellule pilote)	54
2.4.2 Les méthodes directes	54
2.4.2.1 Méthodes à contre réaction de puissance	54
2.4.2.1.1 Méthode basée sur le contrôleur Perturb and Observe (P&O_MPPT)	55
2.4.2.1.2 Méthode basée sur le contrôleur d'incrémental d'inductance (IncCond)	57
2.5 LES MÉTHODES SOFT COMPUTING (MSC)	58
2.5.1 Les méthodes basées sur l'intelligence artificielle (MAI)	59
2.5.1.1 Contrôleur MPPT à base de réseaux de neurones artificiels (ANN_MPPT)	60
2.5.1.2 Contrôleur MPPT à base de logique floue (FLC_MPPT)	61
2.5.1.3 Contrôleur MPPT à base de méthodes hybrides	63
2.5.1.3.1 Méthodes basées sur les systèmes neuro-flous	63
2.5.1.3.2 Méthodes basées sur la logique floue et les algorithmes génétiques	64
2.5.2 Les méthodes bio inspirées (MBI)	67
2.5.2.1 Concepts de base du bio inspiré	67
2.5.2.2 Méthodes basées sur les algorithmes évolutionnaires (MAE)	70
2.5.2.2.1 Contrôleur MPPT à base des algorithmes génétiques (GA_MPPT)	71
2.5.2.2.2 Contrôleur MPPT à base de l'algorithme à évolution différentielle (DE_MPPT)	72
2.5.2.3 Méthodes basées sur l'intelligence en Essaim	73
2.5.2.3.1 Contrôleur MPPT à base de l'algorithme d'optimisation par Essaim de particules (PSO_MPPT)	75

TABLE DES MATIÈRES

2.5.2.3.2	Contrôleur MPPT à base de l'algorithme de colonies d'abeilles (ABC_MPPT)	77
2.5.2.3.3	Contrôleur MPPT à base de l'algorithme de luciole (FA_MPPT)	78
2.5.2.3.4	Contrôleur MPPT à base de l'algorithme d'optimisation de meute de loups (GWO_MPPT)	80
2.5.2.3.5	Contrôleur MPPT à base de l'algorithme de pollinisation des fleurs (FPA_MPPT)	82
2.5.2.3.6	Contrôleur MPPT à base de l'algorithme d'optimisation de ver luisant (GWSO_MPPT)	84
2.5.2.3.7	Contrôleur MPPT à base de l'algorithme d'optimisation de l'enseignement-apprentissage (TLBOA_MPPT)	85
2.5.2.3.8	Contrôleur MPPT à base de l'algorithme de recherche coucou (CS_MPPT)	86
2.5.2.3.9	Contrôleur MPPT à base d'algorithme de saut de grenouille (SFLA_MPPT)	88
2.5.2.3.10	Contrôleur MPPT à base de l'algorithme d'optimisation de colonie de fourmis (ACO_MPPT)	89
2.5.2.3.11	Contrôleur MPPT à base de l'algorithme d'optimisation de chauve-souris (BSO_MPPT)	91
2.6	DISCUSSION	93
2.7	CONCLUSION	97

PARTIE II : APPROCHE SOFTWARE POUR L'IMPLÉMENTATION DES CONTRÔLEURS MPPT APPLIQUÉE AUX ALGORITHMES ACO_NPU ET BSO

CHAPITRE 3	LES MÉTHODES BIO INSPIRÉES POUR LA RECHERCHE DU POINT DE PUISSANCE MAXIMALE	100
3.1	INTRODUCTION	100
3.2	APPLICATION DES ALGORITHMES ACO_NPU ET BSO POUR LA RECHERCHE DU POINT DE PUISSANCE MAXIMALE	100
3.2.1	Conception du contrôleur ACO_NPU_MPPT	101
3.2.1.1	<i>Principe de base de l'algorithme ACO : de l'inspiration au modèle mathématique</i>	<i>101</i>
3.2.1.2	<i>Description de l'algorithme ACO_NPU</i>	<i>103</i>
3.2.1.3	<i>Application de l'algorithme ACO_NPU pour la commande MPPT.....</i>	<i>104</i>
3.2.1.3.1	Recherche du point de puissance maximale PPM	106
3.2.1.3.2	Surveillance des changements climatiques	108

TABLE DES MATIÈRES

3.2.2 Conception du contrôleur BSO_MPPT	108
3.2.2.1 Principe de base de l'algorithme BSO : de l'inspiration au modèle mathématique	108
3.2.2.2 Description de l'algorithme BSO	109
3.2.2.3 Application de l'algorithme BSO pour la commande MPPT.....	111
3.2.2.3.1 Recherche du point de puissance maximale PPM	112
3.3 CONCLUSION	116
CHAPITRE 4 : MISE EN ŒUVRE ET SIMULATION DES CONTRÔLEURS MPPT : ÉTUDE COMPARATIVE DES MÉTHODES CONVENTIONNELLES, INTELLIGENTES ET BIO INSPIRÉES ...	118
4.1 INTRODUCTION	118
4.2 MÉTHODOLOGIE DE SIMULATION DES DIFFERENTS CONTRÔLEURS MPPT	119
4.2.1 Modélisation du système photovoltaïque	119
4.2.2 Paramétrage des algorithmes ACO_NPU et BSO	120
4.2.2.1 Paramétrage de l'algorithme ACO_NPU	121
4.2.2.2 Paramétrage de l'algorithme BSO	122
4.2.3 Simulation des différents contrôleurs	123
4.2.3.1 Etude du critère de convergence	124
4.2.3.2 Etude des critères de stabilité, de précision et de robustesse	128
4.2.3.2.1 Augmentation rapide de l'éclairement	128
4.2.3.2.2 Augmentation rapide de l'éclairement avec un temps transitoire	131
4.2.3.2.3 Diminution rapide de l'éclairement	133
4.2.3.2.4 Diminution rapide de l'éclairement avec un temps transitoire ...	136
4.2.3.2.5 Diminution rapide de la température	138
4.2.3.2.6 Effet de l'ombrage partiel	140
4.3 SYNTHÈSE DES DIFFERENTS CONTRÔLEURS MPPT	143
4.4 CONCLUSION	144
CHAPITRE 5 : PROPOSITION D'UNE NOUVELLE PLATEFORME POUR LE PROTOTYPAGE RAPIDE DES CONTRÔLEURS MPPT	146
5.1 INTRODUCTION	146
5.2 ÉTAT DE L'ART SUR LES PLATEFORMES POUR LES EnR	146
5.3 PRÉSENTATION DE LA PLATEFORME PVSP	148

TABLE DES MATIÈRES

5.4 ARCHITECTURE DE LA PLATEFORME PVSP	150
5.4.1 Le module "GPV"	150
5.4.2 Le module "Contrôleur MPPT"	150
5.4.3 Le module "Algorithme MPPT"	151
5.4.4 Le module "Afficheur"	151
5.5 DÉVELOPPEMENT DE LA PLATEFORME PVSP	151
5.5.1 L'onglet "Simulation"	153
5.5.2 L'onglet "Comparaison"	154
5.6 FONCTIONNALITÉS DE LA PLATEFORME PVSP	155
5.6.1 Sélection du modèle du panneau et élaboration de l'architecture du générateur photovoltaïque	155
5.6.2 Spécification des conditions atmosphériques	156
5.6.3 Sélection du contrôleur MPPT	157
5.6.4 Simulation du contrôleur MPPT	158
5.6.5 Comparaison des résultats de simulations	159
5.7 CONCLUSION	163
 PARTIE III : APPROCHE HARDWARE POUR L'IMPLÉMENTATION DES CONTRÔLEURS MPPT APPLIQUÉE AUX ALGORITHMES ACO_NPU ET BSO	
CHAPITRE 6 : ÉTAT DE L'ART SUR L'IMPLÉMENTATION DES CONTRÔLEURS MPPT SUR CIRCUIT FPGA	166
6.1 INTRODUCTION	166
6.2 LES CIRCUITS FIELD PROGRAMMABLES GATE ARRAY FPGA	167
6.2.1 Historique	167
6.2.2 Les principales caractéristiques des circuits FPGA	168
6.2.2.1 <i>Les performances</i>	168
6.2.2.2 <i>Le temps de mise sur le marché</i>	168
6.2.2.3 <i>Le coût</i>	169
6.2.2.4 <i>La fiabilité</i>	169
6.2.2.5 <i>La maintenance à long terme</i>	169
6.2.3 Le marché des circuits FPGA	170

TABLE DES MATIÈRES

6.2.4 Architecture générale des circuits FPGA de XILINX	171
6.2.4.1 Structure des Blocs Logiques (CLB)	172
6.2.4.2 Les ressources d'interconnexions	173
6.2.4.3 Les blocs d'Entrée/Sortie (E/S).....	174
6.2.5 Evolution des circuits FPGA de XILINX	174
6.3 IMPLÉMENTATION DES CONTRÔLEURS MPPT SUR CIRCUITS FPGA : ÉTAT DE L'ART	178
6.4 CONCLUSION.....	183
CHAPITRE 7 : PROTOTYPAGE RAPIDE DE CONTRÔLEUR BIO INSPIRÉ SUR CIRCUIT FPGA	185
7.1 INTRODUCTION	185
7.2 MÉTHODOLOGIE DE CONCEPTION HARDWARE : DU DOMAINE ALGORITHMIQUE JUSQU'À LA GÉNÉRATION D'IP	185
7.2.1 Evolution des méthodologies de conception	185
7.2.1.1 L'approche de conception classique RTL et ses limites	189
7.2.1.2 L'approche de conception HLS et ses avantages	192
7.3 PRÉSENTATION DE LA NOUVELLE MÉTHODOLOGIE DE CONCEPTION..	197
7.3.1 L'outil Vivado HLS.....	199
7.3.2 L'outil Vivado ISE Design Suite	203
7.4 APPLICATION DE L'APPROCHE HLS POUR LE PROTOTYPAGE RAPIDE DES CONTRÔLEURS BIO INSPIRÉS <i>ACO_NPU_MPPT</i> ET <i>BSO_MPPT</i>	204
7.4.1 Génération des IPs <i>ACO_NPU_MPPT</i> et <i>BSO_MPPT</i>	204
7.4.1.1 Génération de l'IP <i>ACO_NPU_MPPT</i>	204
7.4.1.1.1 Développement et vérification du code C pour le contrôleur <i>ACO_NPU_MPPT</i>	205
7.4.1.1.2 Application des directives d'optimisation pour le contrôleur <i>ACO_NPU_MPPT</i>	207
7.4.1.1.3 Synthèse du code C du contrôleur <i>ACO_NPU_MPPT</i>	208
7.4.1.1.4 Création de l'IP <i>ACO_NPU_MPPT</i> : Exportation au niveau RTL	211
7.4.1.2 Génération de l'IP <i>BSO_MPPT</i>	213
7.4.1.2.1 Développement et vérification du code C pour le contrôleur <i>BSO_MPPT</i>	213
7.4.1.2.2 Application des directives d'optimisation pour le contrôleur <i>BSO_MPPT</i>	214

TABLE DES MATIÈRES

7.4.1.2.3 Synthèse du code C du contrôleur BSO_MPPT	215
7.4.1.2.4 Création de l'IP BSO_MPPT : Exportation au niveau RTL	217
7.4.2 Implémentation des IPs ACO_NPU_MPPT et BSO_MPPT sur FPGA	220
7.4.2.1 <i>Implémentation de l'IP ACO_NPU_MPPT</i>	220
7.4.2.1.1 Simulation de l'IP_ACO_NPU_MPPT	221
7.4.2.1.2 Synthèse de l'IP_ACO_NPU_MPPT	222
7.4.2.1.3 Implémentation de l'IP_ACO_NPU_MPPT	224
7.4.2.2 <i>Implémentation de l'IP BSO_MPPT</i>	225
7.4.2.2.1 Simulation de l'IP_BSO_MPPT	225
7.4.2.2.2 Synthèse de l'IP_BSO_MPPT	226
7.4.2.2.3 Implémentation de l'IP_BSO_MPPT	228
7.4.3 Discussion	229
7.5 PROPOSITION D'UNE PLATEFORME HARDWARE POUR LE PROTOTYPAGE RAPIDE DES CONTRÔLEURS MPPT	232
7.5.1 Présentation de la plateforme	232
7.5.2 Architecture de la plateforme	232
7.6 VERS UNE PLATEFORME GLOBALE SOFTWARE/HARDWARE POUR LE PROTOTYPAGE RAPIDE DE CONTROLEURS IP_MPPT	234
7.6.1 Présentation de la plateforme globale Software/Hardware.....	235
7.6.2 Fonctionnement de la plateforme globale Software/Hardware	236
7.7 CONCLUSION	238
CONCLUSION GÉNÉRALE	241
BIBLIOGRAPHIE	245

Tableau.2.1	Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application des réseaux de neurones artificiels pour le développement de contrôleurs MPPT	61
Tableau.2.2	Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de la logique floue pour le développement de contrôleurs MPPT	63
Tableau.2.3	Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de méthodes hybride pour le développement de contrôleurs MPPT.....	65
Tableau.2.4	Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application des Algorithmes Génétiques pour le développement de contrôleurs MPPT.....	72
Tableau.2.5	Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application des Algorithmes à évolution différentielle pour le développement de contrôleurs MPPT.....	73
Tableau.2.6	Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de l'algorithme d'optimisation par essaim de particules pour le développement de contrôleurs MPPT.....	76
Tableau.2.7	Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de l'algorithme d'optimisation par colonies d'abeilles pour le développement de contrôleurs MPPT	78
Tableau.2.8	Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de l'algorithme d'optimisation de lucioles pour le développement de contrôleurs MPPT.....	80
Tableau.2.9	Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de l'algorithme d'optimisation de la meute de loup pour le développement de contrôleurs MPPT.....	82
Tableau.2.10	Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de l'algorithme de pollinisation de fleurs pour le développement de contrôleurs MPPT.....	83
Tableau.2.11	Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de l'algorithme d'optimisation de ver luisant pour le développement de contrôleurs MPPT.....	84
Tableau.2.12	Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de l'algorithme d'optimisation basée sur l'enseignement-apprentissage pour le développement de contrôleurs MPPT.....	86
Tableau.2.13	Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de l'algorithme d'optimisation basée sur la recherche coucou pour le développement de contrôleurs MPPT.....	87
Tableau.2.14	Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de l'algorithme d'optimisation basée sur le saut de grenouille pour le développement de contrôleurs MPPT.....	89
Tableau.2.15	Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de l'algorithme d'optimisation basée sur la de colonie de fourmis pour le développement de contrôleurs MPPT.....	90
Tableau.2.16	Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de l'algorithme d'optimisation basée sur la chauve-souris pour le développement de contrôleurs MPPT.....	92
Tableau.2.17	Critères majeurs relatifs aux différents contrôleurs MPPT.....	94
Tableau.4.1	Résultats de simulation pour différentes valeurs des paramètres.....	121
Tableau.4.2	Résultats de simulation pour différentes valeurs des paramètres.....	122
Tableau.4.3	Différents scénarios d'ombrage.....	141
Tableau.4.4	Caractéristiques des différents contrôleurs MPPT.....	143
Tableau.6.1	Evolution des circuits FPGA de Xilinx.....	175

Tableau.6.2	Synthèse des travaux relatifs à l'implémentation des MPPT sur circuit FPGA.....	179
Tableau.7.1	Estimation des ressources du design.....	207
Tableau.7.2	Estimation des ressources du design après application des directives d'optimisation.....	209
Tableau.7.3	Détails des instances du design.....	209
Tableau.7.4	Mémoire BRAM instancié dans le design.....	209
Tableau.7.5	Registres instancié dans le design.....	209
Tableau.7.6	Estimation des ressources du design sans directives d'optimisation.....	215
Tableau.7.7	Détails des instances du design.....	216
Tableau.7.8	Mémoire <i>BRAM</i> instanciées dans le design.....	216
Tableau.7.9	Registre instanciées dans le design.....	216
Tableau.7.10	Taux d'utilisation des ressources du circuit pour chaque fonction constituant l'architecture.....	224
Tableau.7.11	Ressources utilisées pour chaque IP.....	229

Figure.1.1 Constitution et principe d'une cellule photovoltaïque	31
Figure.1.2 Caractéristique I(V) d'une cellule photovoltaïque.	32
Figure.1.3 Caractéristique P(V) d'une cellule photovoltaïque.....	32
Figure.1.4 Circuit équivalent d'une cellule photovoltaïque à deux diodes.....	33
Figure.1.5 Caractéristique résultante d'un groupement de <i>ns</i> cellules en série.	34
Figure.1.6 Caractéristique résultante d'un groupement de <i>np</i> cellules en parallèle.....	34
Figure.1.7 Influence de l'éclairement sur les courbes	35
Figure.1.8 Influence de la température sur les courbes	37
Figure.1.9 Caractéristique P-V pour différents scénarios d'ombrage	38
Figure.1.10 Connexion directe GPV-Charge	39
Figure.1.11 Connexion d'un GPV à une charge à travers un étage d'adaptation	40
Figure.1.12 Chaîne de conversion photovoltaïque avec convertisseur statique contrôlé par une commande MPPT	41
Figure.2.1 Classification des critères de sélection des contrôleurs MPPT	45
Figure.2.2 Classification des méthodes MPPT selon [22].....	47
Figure.2.3 Classification des méthodes MPPT selon [23]	48
Figure.2.4 Classification des méthodes MPPT selon [26]	49
Figure.2.5 Classification des méthodes MPPT selon [28, 29, 31].....	50
Figure.2.6 Classification des méthodes MPPT proposée.....	51
Figure.2.7 Système de contrôle MPPT à rétroaction de tension avec tension de référence fixe....	52
Figure.2.8 Système de contrôle MPPT à rétroaction de tension avec tension de référence contrôlable.....	53
Figure.2.9 Système de contrôle MPPT à rétroaction de tension avec une cellule PV de référence.....	53
Figure.2.10 Mise en œuvre de la méthode MPPT basée sur la fraction du courant de court-circuit	54
Figure.2.11 Schéma bloc du système de contrôle avec rétroaction de puissance.....	55
Figure.2.12 Organigramme du contrôleur P&O.....	56
Figure.2.13 Organigramme du contrôleur Inc Cnd.....	57
Figure.2.14 Architecture d'un réseau de neurones.....	60
Figure.2.15 Mise en œuvre du contrôleur MPPT basée sur les réseaux de neurones.....	61
Figure.2.16 Structure de base du contrôleur MPPT à base logique floue.....	62
Figure.2.17 Mise en œuvre du contrôleur MPPT basée sur la logique floue.....	62
Figure.2.18 Architecture du réseau ANFIS.....	64
Figure.2.19 Mise en œuvre du contrôleur MPPT à base de Neuro Floue ANFIS sur la logique floue.....	64
Figure.2.20 Mise en œuvre du contrôleur MPPT à base de la méthode floue optimisé par les Algorithmes Génétiques.....	65
Figure.2.21 Caractéristiques du bio inspiré.....	68
Figure.2.22 Etapes d'élaboration des algorithmes bio inspirés.....	69
Figure.2.23 Principaux groupes du domaine Bio Inspiré.....	70
Figure.2.24 Structure d'un algorithme évolutionnaire.....	70
Figure.2.25 Organigramme du contrôleur GA.....	71
Figure.2.26 Organigramme du contrôleur DE.....	73
Figure.2.27 Essaim d'oiseaux et banc de poissons.....	75
Figure.2.28 Organigramme du contrôleur PSO.....	76
Figure.2.29 Organigramme du contrôleur ABC.....	77
Figure.2.30 Essaim de lucioles.....	79
Figure.2.31 Organigramme du contrôleur FA.....	79
Figure.2.32 Hiérarchie au sein d'une meute de loups.....	80
Figure.2.33 Différentes étapes recherche de proie.....	80
Figure.2.34 Organigramme du contrôleur GWO.....	81
Figure.2.35 Processus de pollinisation.....	82
Figure.2.36 Organigramme du contrôleur FPA.....	83
Figure.2.37 Organigramme du contrôleur GSO.....	84

Figure.2.38 Organigramme du contrôleur TLBO.....	85
Figure.2.39 Oiseaux parasites de nids « Coucou ».....	86
Figure.2.40 Organigramme du contrôleur CS.....	87
Figure.2.41 Organigramme du contrôleur SFLA.....	88
Figure.2.42 Déplacement des fourmis lors de la recherche de nourriture.....	89
Figure.2.43 Organigramme du contrôleur ACO.....	90
Figure.2.44 Déplacement de Chauve-souris lors de la recherche de nourriture.....	91
Figure.2.45 Organigramme du contrôleur BSO.....	92
Figure.3.1 (a) Déplacement aléatoire de fourmis	101
Figure.3.1 (b) Déplacement guidé par la phéromone.....	101
Figure.3.2. Comportement des fourmis lors de la recherche de nourriture.....	102
Figure.3.3 Organigramme du contrôleur ACO_NPU MPPT proposé.....	105
Figure.3.4 Phase de recherche du PPM avec l’algorithme ACO_NPU.....	106
Figure.3.5 Phase d’initialisation.....	106
Figure.3.6 (a) Initialisation et distribution fourmis selon fonction objective	107
Figure.3.6 (b) les fourmis se déplacent vers un espace de recherché restreint.....	107
Figure.3.6 (c) les fourmis convergent vers le point optimum, i.e. the MPP.....	107
Figure.3.7 Processus d’exécution de l’algorithme proposé ACO_NPU MPPT.....	108
Figure.3.8 Echolocalisation des chauves-souris.....	109
Figure.3.9 Comportement des chauves-souris dans un espace de recherche.....	109
Figure.3.10 Organigramme du contrôleur BSO MPPT proposé.....	112
Figure.3.11 Phase de la recherche du PPM avec de l’algorithme BSO.....	113
Figure.3.12 Phase d’initialisation.....	113
Figure.3.13 Phase de génération de nouvelle solution.....	114
Figure.3.14 Phase d’identification de la meilleure solution.....	115
Figure.3.15 (a) Initialisation et distribution de la population	116
Figure.3.15 (b) génération de nouvelle solution et direction vers la meilleurs	116
Figure.3.15 (c) chauve-souris en direction du PPM.....	116
Figure.4.1 Schéma bloc pour la simulation du système PV.....	119
Figure.4.2 Simulation pour différentes valeurs de paramètres du contrôleur. ACO_NPU_MPPT	121
Figure.4.3 Simulation pour différentes valeurs de paramètres du contrôleur BSO_MPPT.....	122
Figure.4.4 Evolution de la puissance de sortie du contrôleur ACO_NPU_MPPT sous conditions standards (S=1000W/m ² , T=25°C)	124
Figure.4.5 Evolution de la puissance de sortie du contrôleur BSO_MPPT sous conditions standards (S=1000W/m ² , T=25°C)	125
Figure.4.6 Différentes étapes du contrôleur ACO_NPU_MPPT pour atteindre le PPM.....	126
Figure.4.7 Différentes étapes du contrôleur BSO_MPPT pour atteindre le PPM.....	126
Figure.4.8 Evolution de la puissance de sortie des contrôleurs BSO, ACO_NPU, FL_GA, ANFIS, ANN, FLC and P&O MPPT sous conditions standards.....	127
Figure.4.9 Temps de réponse des contrôleurs BSO, ACO_NPU, ACO, FL_GA, ANFIS, ANN, FLC et P&O MPPT.....	127
Figure.4.10 Evolution de la puissance du contrôleur BSO et ACO_NPU MPPT pour une augmentation rapide de l’éclairement de 500 à 1000W/m ² , 25°C.....	129
Figure.4.11 Evolution de la puissance du contrôleur ACO_NPU_MPPT, BSO_MPPT, GA_MPPT, ANFIS_MPPT, FL_MPPT, ANN_MPPT et P&O MPPT pour une augmentation rapide de l’éclairement de 500 à 1000W/m ² , 25°C.....	130
Figure.4.12 (a)-(b). Evolution de la puissance du contrôleur ACO_NPU et BSO MPPT pour une augmentation rapide de l’éclairement de 500 à 1000W/m ² , 25°C, avec une transition de 5s.....	131
Figure.4.13 Evolution de la puissance du contrôleur ACO_NPU_MPPT, BSO_MPPT, GA_MPPT, ANFIS_MPPT, FL_MPPT, ANN_MPPT et P&O MPPT pour une augmentation rapide de l’éclairement de 500 à 1000W/m ² , 25°C, avec une transition de 5s.....	132

Figure.4.14 Evolution de la puissance du contrôleur ACO_NPU et BSO MPPT pour une diminution rapide de l'éclairement de 1000 à 500W/m ² , 25°C.....	134
Figure.4.15 Evolution de la puissance du contrôleur ACO_NPU_MPPT, BSO_MPPT, GA_MPPT, ANFIS_MPPT, FL_MPPT, ANN_MPPT et P&O MPPT pour une diminution rapide de l'éclairement de 1000 à 500W/m ² , 25°C.....	135
Figure.4.16 Evolution de la puissance du contrôleur ACO_NPU et BSO MPPT pour une diminution rapide de l'éclairement de 1000 à 500W/m ² , 25°C, avec une transition de 5s.....	136
Figure.4.17 Evolution de la puissance du contrôleur ACO_NPU_MPPT, BSO_MPPT, GA_MPPT, ANFIS_MPPT, FL_MPPT, ANN_MPPT et P&O MPPT pour une diminution rapide de l'éclairement de 1000 à 500W/m ² , 25°C, avec une transition de 5s.....	137
Figure.4.18 Evolution de la puissance du contrôleur ACO_NPU et BSO MPPT pour une diminution rapide de la température de 50 à 25°C, avec une transition de 5s.....	139
Figure.4.19 Evolution de la puissance du contrôleur ACO_NPU_MPPT, BSO_MPPT, GA_MPPT, ANFIS_MPPT, FL_MPPT, ANN_MPPT et P&O MPPT pour une diminution rapide de la température de 50 à 25°C, avec une transition de 5s.....	140
Figure.4.20 Caractéristique P-V pour différents pattern d'ombrage.....	141
Figure.4.21 The extracted power during the selected shading patterns with the corresponding P_V characteristic curve, (a) SP1, (b) SP2, (c) SP3.....	142
Figure.5.1 Structure de la plateforme PVSP.....	149
Figure.5.2 Les modules constituant le système PV globale de la plateforme.....	150
Figure.5.3 Structure interne de la plateforme.....	152
Figure.5.4 Interface principale de la plateforme PVSP.....	152
Figure.5.5 Interface principale de PVSP avec activation de l'onglet <i>Simulation</i>	153
Figure.5.6 Interface principale de PVSP avec activation de l'onglet <i>Comparaison</i>	154
Figure.5.7 Sélection du modèle du panneau PV, Dimensionnement du générateur PV.....	155
Figure.5.8 Configuration des conditions climatiques.....	156
Figure.5.9 Configuration des conditions climatiques : variation stable en Température et Irradiation.....	156
Figure.5.10 Configuration des conditions climatiques : variation rapide en Température et Irradiation.....	156
Figure.5.11 Configuration des conditions climatiques : variation spécifique pour chaque panneau (a) en Température, (b) en Irradiation.....	157
Figure.5.12 Configuration des conditions climatiques : ombrage partiel déclenché avec temporisateur.....	157
Figure.5.13 Sélection et paramétrage des contrôleurs, (a) P&O, (b) PSO.....	158
Figure.5.14 Lancement de simulation.....	158
Figure.5.15 Trace de l'évolution I-V et P-V et visualisation du PPM, (a) condition standard, (b) sous ombrage partiel.....	159
Figure.5.16 Configuration de la plateforme pour la simulation du contrôleur P&O, (a) condition standard, (b) sous ombrage partiel.....	160
Figure.5.17 Caractéristique P-V et I-V sous condition standard.....	161
Figure.5.18 Trace de l'évolution de la puissance dans le temps sous condition standard.....	161
Figure.5.19 Trace d'exécution du contrôleur P&O.....	161
Figure.5.20 Caractéristique P-V sous condition d'ombrage partiel.....	161
Figure.5.21 Trace de l'évolution de la puissance dans le temps sous ombrage partiel.....	162
Figure.5.22 Trace d'exécution du contrôleur P&O sous ombrage partiel.....	162
Figure.5.23 (a)-(b) Caractéristique P-V sous condition d'ombrage partiel.....	163
Figure.6.1. Les parts de marché des circuits FPGA pour chaque firme.....	171
Figure.6.2. Architecture générale d'un circuit FPGA.....	171

Figure.6.3. Architecture interne d'un CLB.....	172
Figure.6.4. Ressources d'interconnexion.....	173
Figure.6.5. Vue transversale de l'architecture du circuit <i>Virtex7</i>	1476
Figure.6.6. Architecture en colonne des circuits FPGA <i>Série7</i> de Xilinx.....	166
Figure.6.7. Architecture interne d'un CLB du circuit FPGA <i>Virtex Série7</i> de Xilinx.....	177
Figure.6.8. Architecture interne du <i>DSP48E1</i> du circuit FPGA <i>Virtex Série7</i> de Xilinx.....	178
Figure.7.1 Diagramme de Gajski et Kuhn : domaine et niveaux d'abstraction	186
Figure.7.2. Diagramme de Gajski et Kuhn : (a) niveaux d'abstraction ; (b) principaux composants aux différents niveaux d'abstraction.....	187
Figure.7.3. Les approches de conception.....	189
Figure.7.4. Flot de conception classique.....	190
Figure.7.5. Flot de conception HLS.....	193
Figure.7.6 Nouvelles approches de conception.....	195
Figure.7.7. Nouvelle méthodologie de conception.....	197
Figure.7.8. Flot de conception FPGA basé VIVADO HLS / VIVADO Design Suite.....	198
Figure.7.9. Flot de conception de l'outil <i>VIVADO HLS</i>	199
Figure.7.10. Optimisation « <i>Pipelining des boucles</i> ».....	202
Figure.7.11. Optimisation « <i>Pipelining des fonctions</i> ».....	202
Figure.7.12. Optimisation « <i>Dataflow Pipelining des fonctions</i> ».....	202
Figure.7.13. Flot de conception de l'outil <i>VIVADO ISE/ Intégrateur d'IP</i>	203
Figure.7.14. Sélection de la carte de prototypage/ circuit FPGA.....	204
Figure.7.15. Les structures de données utilisées.....	205
Figure.7.16. Les fonctions utilisées.....	206
Figure.7.17. Validation du code <i>C</i> pour le contrôleur « <i>ACO_NPU_MPPT</i> ».....	206
Figure.7.18. Directives d'optimisations appliquées au code <i>C</i> du contrôleur <i>ACO_NPU_MPPT</i> . <i>BSO_MPPT</i>	208
Figure.7.19. Analyse d'exécution des performances pour le contrôleur <i>ACO_NPU_MPPT</i>	210
Figure.7.20. Ressources allouées à la fonction « <i>aco_sort</i> ».....	211
Figure.7.21. (a) Sélection du mode de génération et du langage <i>HDL</i>	212
Figure.7.21. (b) Création du bloc <i>IP</i> pour le contrôleur <i>ACO_NPU_MPPT</i>	212
Figure.7.21. (c) Extrait du code <i>VHDL</i> généré pour la fonction « <i>aco_search</i> ».....	212
Figure.7.22. La structure de données utilisées pour le contrôleur <i>BSO_MPPT</i>	213
Figure.7.23. Les fonctions utilisées pour le contrôleur <i>BSO_MPPT</i>	213
Figure.7.24. Validation du code <i>C</i> pour le contrôleur « <i>BSO_MPPT</i> ».....	214
Figure.7.25. Directives d'optimisations appliquées au code <i>C</i> du contrôleur <i>BSO_MPPT</i>	215
Figure.7.26. Analyse temporelle d'exécution du code pour le contrôleur <i>BSO_MPPT</i>	216
Figure.7.27. Ressources allouées pour la fonction« <i>comparison_solution</i> ».....	217
Figure.7.28. (a) Création du bloc <i>IP</i> pour le contrôleur <i>BSO_MPPT</i> généré	218
Figure.7.28. (b) Code <i>VHDL</i> généré pour la fonction « <i>main_BSO_compaison_solution</i> ».....	218
Figure.7.29. (a) Validation du code <i>C</i> pour le « <i>GPV</i> ».....	219
Figure.7.29. (b) Création et génération du bloc <i>IP_GPV</i>	219
Figure.7.30. Intégration de l' <i>IP</i> « <i>ACO_MPPT</i> » dans <i>IP Catalog</i> la bibliothèque de l'outil Vivado.....	220
Figure.7.31. Architecture globale <i>IP_ACO_NPU_MPPT</i> et <i>IP_GPV</i> Wrappée.....	220
Figure.7.32. Simulation Comportementale l' <i>IP_ACO_NPU_MPPT</i> (sans ombrage).....	221
Figure.7.33. Simulation Comportementale de l' <i>IP_ACO_NPU_MPPT</i> (avec ombrage).....	221
Figure.7.34. Architecture globale au niveau <i>RTL</i> après étape de synthèse.....	222
Figure.7.35. Schéma <i>RTL</i> des différents blocs constituant l'architecture.....	223
Figure.7.36. Détail de l'architecture de la fonction « <i>aco_sort</i> » au niveau <i>RTL</i>	223
Figure.7.37. Implémentation de l'architecture sur circuit FPGA.....	224
Figure.7.38. Intégration de l' <i>IP</i> « <i>BSO_MPPT</i> » dans <i>IP Catalog</i> la bibliothèque de l'outil Vivado.....	225
Figure.7.39. Architecture globale <i>IP_BSO_MPPT</i> et <i>IP_GPV</i> Wrappée.....	225
Figure.7.40. Simulation Comportementale l' <i>IP_BSO_MPPT</i>	226

Figure.7.41. Architecture globale au niveau RTL après étape de synthèse.....	226
Figure.7.42. Schéma RTL des différents blocs constituant l'architecture.....	227
Figure.7.43. Détail de l'architecture de la fonction « <i>Best_solution</i> » au niveau <i>RTL</i>	227
Figure.7.44. Implémentation de l'architecture du système PV sur circuit FPGA.....	228
Figure.7.45. Synthèse des ressources utilisées pour chaque fonction du design.....	228
Figure.7.46. (a) Taux des ressources utilisées pour l' <i>IP_ACO_NPU_MPPT</i>	229
Figure.7.46. (b) Répartition des principales fonctions en termes de LUTs de l'IP ACO_NPU_MPPT.....	229
Figure.7.47. (a) Taux des ressources utilisées pour l' <i>IP_BSO_MPPT</i>	230
Figure.7.47. (b) Répartition des principales fonctions en termes de LUTs de l'IP <i>BSO_MPPT</i> ...	230
Figure.7.48. (a) Structure des directories <i>IP_BSO_MPPT</i>	231
Figure.7.48. (b) Structure des directories <i>IP_ACO_NPU_MPPT</i>	231
Figure.7.49. Architecture de la plateforme hardware.....	233
Figure.7.50. Architecture de la plateforme globale Software/Hardware.....	235

Liste des abréviations

ABC_MPPT	Artificial Bee Colony MPPT
ACO_MPPT	Ant Colony Optimization Algorithm MPPT
ACO_NPU_MPPT	Ant Colony Optimization with New Pheromone Update Colony Optimization MPPT
ANN_MPPT	Artificial Neural Network MPPT
BSO_MPPT	Bat Swarm Optimization Algorithm MPPT
CS_MPPT	Cuckoo Search Algorithm MPPT
CMPPT	Centralied Maximum Power Point Tracking
DC	Direct Current (Convertisseur DC-DC)
DE_MPPT	Differential Evolution MPPT
DMPPT	Distributed Maximum Power Point Tracking
FA_MPPT	Firefly Algorithm MPPT
FLC_MPPT	Fuzzy Logic Controller MPPT
FPA_MPPT	Flower Pollination Algorithm MPPT
FPGA	Field Programmable Gate Array
GA_MPPT	Genetic Algorithm MPPT
GPV	Générateur Photovoltaïque
GWO_MPPT	Gray Wolf Optimization MPPT
GWSO_MPPT	Glow Worm Swarm Optimization MPPT
HDL	Hardware Description Language
HLS	High Level Synthesis
IncCnd_MPPT	Incremental Conductance MPPT
IP	Intellectual Property
MAE	Méthodes d'Algorithmes Evolutionnaires
MBD	Model Based Design
MBI	Méthodes Bio Inspirées
MC	Méthodes Conventionnelles
MI	Méthodes Intelligentes
MIA	Méthodes d'Intelligence Artificielles
MIE	Méthodes d'intelligence en Essaim
MPP	Maximum Power Point
MPPT	Maximum Power Point Tracking
MSC	Méthodes Soft Computing
P&O_MPPT	Perturb and Observe MPPT

Liste des abréviations

PPM	Point de Puissance Maximal
PS	Partial Shadding
PSO_MPPT	Particle Swarm Optimization MPPT
PV	Photovoltaïque
PVSP	PhotoVoltaic Simulation Platform
RTL	Register Transfer Level
SFLA_MPPT	Shuffled Frog Leap Algorithm MPPT
TLBOA_MPPT	Teaching Learning Based Optimization Algorithm MPPT

INTRODUCTION GÉNÉRALE

Les besoins mondiaux en énergie de l'industrie moderne n'ont cessé de croître ces dernières années. Les sources conventionnelles d'énergie, principalement fossiles comme le charbon et les hydrocarbures, laissent progressivement place aux énergies propres ou renouvelables dont les coûts de production diminuent progressivement.

La production, le transport, le stockage, la transformation et la consommation des énergies conventionnelles font courir à l'environnement naturel dans lequel évolue l'humanité toute entière, des dangers. En effet, ces sources d'énergie sont responsables directement de grandes pollutions des sols, des nappes phréatiques, des océans, de l'atmosphère, des changements climatiques liés à l'augmentation de l'effet de serre...etc. Les exemples de catastrophes qui en témoignent n'en sont que nombreux.

Par ailleurs, les considérations stratégiques des grandes puissances générant continuellement des craintes de dépendance aux hydrocarbures, particulièrement celles qui n'en produisent pas, ou peu, d'une part, ainsi que la quête morale des sociétés modernes de sources d'énergie moins polluantes et respectueuses de la nature, a conduit ces pays à investir dans les énergies renouvelables telles que l'énergie solaire, éolienne, géothermique...etc. Bien qu'encore modeste dans leur contribution au bilan énergétique, il n'en demeure pas moins que, année après année, ces sources d'énergie consolident leur effort dans le choix énergétique car elles sont non seulement inépuisables, mais promettent un avenir au développement durable et une préservation de l'environnement dans lequel évolue l'humanité.

Certaines de ces énergies ont connu un développement plus ou moins significatif alors que d'autres continuent à évoluer dans le domaine de la recherche et du développement. Ce qui est certain, c'est que les progrès et les prouesses technologiques réalisés à ce jour, tant dans le domaine de la recherche que celui du monde économique, promettent des avancées significatives pour les prochaines décades.

Parmi les sources dites propres, l'énergie solaire apparaît comme étant une source d'énergie alternative pouvant répondre aux besoins des applications exprimées dans des milieux hostiles comme les zones désertiques ou enclavées, des dispositifs comme des satellites ou sondes, ...etc.

En effet, la technologie de conversion d'énergie solaire par effet photovoltaïque a montré un potentiel important comme source d'énergie. Cela explique, en partie, un développement tangible de l'énergie solaire au niveau résidentiel et industriel.

L'énergie solaire permet d'exploiter la lumière apportée par les rayons du soleil pour produire de l'énergie sous forme d'électricité ou de chaleur.

La conversion de l'énergie solaire en énergie électrique repose sur la découverte de Becquerel sur l'effet photovoltaïque qui permet la conversion directe du rayonnement solaire en électricité par action des photons. L'énergie photovoltaïque est de nos jours l'une des sources d'énergie la plus développée. Plusieurs décennies de recherche ont permis d'atteindre une maturité en matière de rendement des modules et des systèmes photovoltaïques.

Dans ce contexte, l'énergie, que pourrait produire les systèmes photovoltaïques, apparaît comme étant une source d'énergie importante couvrant d'année en année une proportion non négligeable des besoins énergétiques de l'humanité.

De nos jours, la filière photovoltaïque a atteint un stade de maturité et de crédibilité économique avérée. Il n'en demeure pas moins, qu'il subsiste encore aujourd'hui certains aspects à améliorer. Seulement, sur le plan de la recherche, la conception, l'optimisation et la réalisation des systèmes photovoltaïques posent toujours des difficultés techniques qui restent d'actualité. Les prouesses d'aujourd'hui résident dans la capacité de donner des réponses à ces problèmes techniques pour amener ces systèmes à un degré de maturité suffisant pour en faire des produits industriels viables à part entière.

Pour les problèmes à solutionner, nous évoquerons ceux relatif à l'association de plusieurs cellules photovoltaïques donnant lieu à un générateur PV. La recherche sans cesse du point de puissance maximale (PPM) dépend de plusieurs paramètres : d'abord, la position de ce point qui n'est pas fixe mais se déplace en fonction de l'ensoleillement, de l'ombrage ainsi que de la température. A cause du coût relativement onéreux de ce genre de systèmes, il est primordial d'extraire le maximum de puissance du générateur PV moyennant des mécanismes de tracking (*Maximum Power Point* MPPT) de manière à maintenir en permanence la puissance générée à son niveau maximal.

Depuis l'apparition de la première chaîne de conversion PV dotée d'un contrôleur MPPT en 1968 [1], la littérature n'a cessé de proposer un éventail de solutions pour les méthodes de contrôle spécialement dédiées à l'optimisation de la puissance produite par le champ PV.

La conception des contrôleurs était d'abord basée sur les méthodes conventionnelles (MC) : Perturb & Observe (P&O), Incremental Conductance (IncCond) et Hill Climbing dont l'avantage réside dans leur simplicité. Plus récemment, nous avons assisté à l'émergence de nouveaux types de contrôleurs basés sur l'utilisation des méthodes d'intelligence artificielle (MIA), comme les Réseaux de Neurones, les Algorithmes Génétiques ou la Logique Floue.

Comparé aux contrôleurs conventionnels, ceux issus des méthodes d'intelligence artificielle offrent d'importants avantages en raison, notamment, de leur capacité à surmonter les limites des méthodes conventionnelles liées aux variations atmosphériques et des oscillations autour du PPM. Néanmoins, malgré ces mérites, il n'en demeure pas moins que ces dernières, présentent l'inconvénient majeur du temps de calcul très long.

Pour remédier à ce problème, de nouvelles méthodes intelligentes ont été développées récemment. Celles-ci sont beaucoup plus efficaces que les méthodes d'intelligence artificielles traditionnelles.

Afin de les distinguer de celles-ci, on les définit comme étant des méthodes *Bio inspirées*. Ces méthodes ont été appliquées dans de nombreux domaines en raison de leur aptitude à optimiser la capacité de recherche sans fournir trop de calculs mathématiques.

Au vu des avantages tirés de ces méthodes, ces dernières, récemment introduites dans le domaine des énergies renouvelables, plus particulièrement les systèmes PV, suscitent un vif intérêt de la part de la communauté scientifique et présentent un avenir prometteur qui demeure relativement vierge au vu du nombre de travaux existants.

C'est dans cette optique que nous proposons dans le cadre de notre thèse d'exploiter les performances des algorithmes bio inspirés pour traiter le problème d'extraction et de poursuite de la puissance maximale (MPPT) d'un générateur PV d'une façon précise vu que ces méthodes bio inspirées sont prédisposées pour traiter et prendre en charge le problème d'ombrage partiel et donc d'atteindre le GMPP.

Le choix final s'est porté sur les algorithmes de colonies de fourmis plus communément connus sous le nom de « Ant Colony Optimisation » (ACO) et de la Chauve-Souris ou « Bat Swarm Optimisation » (BSO). Une fois l'étape de conception des deux contrôleurs franchie et toujours dans le contexte d'amélioration des performances des systèmes PV, nous nous sommes intéressés à l'implémentation hardware de ces derniers.

Les progrès liés aux techniques basées sur l'intelligence bio inspirée embarquées sur circuit programmable (*Field Programmable Gate Array*) FPGA a permis de surmonter de réels difficultés d'ingénierie. Cependant, l'application des circuits FPGAs au domaine du solaire PV demeure relativement limitée.

Habituellement, les contrôleurs MPPT sont implémentés sur de microcontrôleurs et des DSPs, bien que ces derniers offrent des performances acceptables, il n'en demeure pas moins qu'ils ne fournissent pas un degré de flexibilité aussi élevé que ceux des circuits FPGA. De ce fait, l'implémentation sur circuit FPGA des contrôleurs Bio inspirés joue un rôle important dans le contrôle et la gestion de puissance des systèmes PV.

Pour l'implémentation hardware des deux contrôleurs bio inspirés, nous avons adopté une nouvelle approche de conception basée sur la synthèse de haut niveau (*High Level Synthesis*, HLS). En effet, durant les dernières années, les évolutions technologiques dans les langages de synthèse de haut niveau (HLS) ont permis de remplacer la conception au niveau RTL pour la génération de certains circuits FPGA. L'introduction d'un outil HLS dans le flot de conception permet d'automatiser une partie de la génération de la description RTL. Ceci réduit considérablement le temps de développement en automatisant la génération de la description matérielle du circuit.

De plus, toujours dans le cadre de nos travaux de recherches, et dans un souci de faciliter le prototypage rapide des contrôleurs MPPT du point de vue Software et Hardware, nous avons proposé deux plateformes distinctes :

La première proposition intitulé *Photovoltaic Simulator Platform PVSP* est dédiée pour le développement et l'évaluation des performances à travers des tests et simulations des contrôleurs MPPT. Elle constitue un outil utile pour la communauté scientifique dont les travaux de recherche portent sur le développement et tests de nouveaux contrôleurs du fait qu'à ce jour, il n'existe pas de plateforme destinée à l'évaluation des performances des

contrôleurs MPPT. De plus, l'outil peut être perçu comme outil pédagogique innovant qui pourrait être destiné à l'enseignement des connaissances de base pour l'optimisation de puissance dans les systèmes et applications PV.

La seconde, quant à elle, est plutôt destinée au prototypage rapide de contrôleurs MPPT en vue d'une implémentation hardware de ces derniers. Elle consiste à la création et la génération de blocs *IP_MPPT*, en partant des spécifications à un haut niveau et en adoptant une nouvelle approche de conception basée sur la synthèse de haut niveau (HLS). Ces derniers seront stockés dans une bibliothèque en vue d'une réutilisation pour une intégration éventuelle dans un design (Design Reuse), facilitant ainsi le développement hardware des systèmes photovoltaïques.

Le présent travail de thèse est subdivisé en trois parties :

- La première partie, intitulée les *Techniques de recherche du point de puissance maximale* est subdivisée en deux chapitres :
 - Dans le premier chapitre, nous présenterons les concepts de base des systèmes PV ;
 - Le deuxième chapitre, quant à lui, portera sur les méthodes de recherche du point de puissance maximale MPPT, où un état de l'art sera donné, suivi de la proposition d'une nouvelle approche pour la classification de ces méthodes ;
- La deuxième partie, intitulée *Approche software pour l'implémentation des contrôleurs MPPT appliquée aux algorithmes ACO_NPU et BSO* est subdivisée en trois chapitres :
 - Dans le chapitre 3, une étude théorique, où les notions de base des méthodes bio inspirées appliquées à la commande MPPT seront étudiées. Le chapitre abordera le développement des contrôleurs MPPT (ACO_NPU et BSO) basés sur ces méthodes ;
 - Le chapitre 4, quant à lui, sera consacré aux tests et simulations des contrôleurs développés ;
 - Le chapitre 5 introduira la nouvelle plateforme software proposée pour le développement et le prototypage des contrôleurs MPPT ;
- La troisième partie, intitulée *Approche hardware pour l'implémentation des contrôleurs MPPT appliquée aux algorithmes ACO_NPU et BSO*, est subdivisée en deux chapitres :
 - Le chapitre 6 introduira les circuits programmables, les approches de conception sur circuit FPGA ainsi qu'un état de l'art sur l'implémentation des contrôleurs MPPT sur circuit FPGA ;

- Le chapitre 7 abordera l'approche de synthèse haut niveau, suivi d'une implémentation hardware des deux contrôleurs bio inspirés par ladite méthode sur circuit FPGA serie 7 de la carte de prototypage *Virtex 707 de Xilinx*.

Nous présenterons ainsi la plateforme hardware qui pourra servir pour le développement et le prototypage de tout type de contrôleur IP_MPPT, qu'il soit issu des méthodes conventionnelles, intelligentes ou bio inspirées.

Nous terminerons par présenter une vue globale de la nouvelle plateforme qui englobe à la fois la partie software et la partie hardware qui sera aisément exploitée pour le prototypage rapide d'IP_MPPT dédiés aux multiples applications liées aux systèmes photovoltaïques.

Nous terminerons notre travail par une conclusion générale dans laquelle nous résumerons les différents résultats obtenus et nous donnerons quelques perspectives qui pourraient être prises en compte pour la continuité de ce travail.

PARTIE I

**LES TECHNIQUES DE RECHERCHE
DU POINT DE PUISSANCE MAXIMALE**

Chapitre 1

**CONCEPTS DE BASE
DES SYSTÈMES PHOTOVOLTAÏQUES**

1.1 INTRODUCTION

Au cours de ce chapitre, les bases théoriques et pratiques nécessaires à la compréhension de notre sujet de recherche, seront présentées.

Ainsi, nous introduirons les concepts de base liés au fonctionnement d'un système photovoltaïque comme suit :

- L'effet photovoltaïque (ou la conversion photon-électron) et la modélisation mathématique de cette conversion ainsi que les propriétés de cette source d'énergie qui en découlent en fonction des conditions atmosphérique (éclairage et température) et de l'influence de l'ombrage sur la production de puissance ;
- La problématique du transfert de la puissance maximale en schématisant le plus possible une chaîne de conversion élémentaire composée d'un convertisseur statique qui joue le rôle d'un adaptateur de charge lors d'une connexion indirecte ;
- Le principe général de fonctionnement d'une commande MPPT.

1.2 L'EFFET PHOTOVOLTAÏQUE

1.2.1 Principe d'une cellule photovoltaïque

L'effet photovoltaïque a été mis en évidence pour la première fois par E. Becquerel en 1839 : il découvrit que certains matériaux délivraient une petite quantité d'électricité quand ils étaient exposés à la lumière. Albert Einstein expliqua le phénomène photoélectrique en 1912, mais il fallut attendre le début des années 50 pour sa mise en application pratique dans la réalisation d'une cellule photovoltaïque en silicium d'un rendement de 4,5% [2].

La première cellule photovoltaïque a été développée aux Etats-Unis en 1954 par les chercheurs des laboratoires Bell, qui ont découvert que la photosensibilité du silicium pouvait être augmentée en dopant le semi-conducteur par des "impuretés" appropriées.

Une cellule photovoltaïque est définie comme étant un capteur composé d'un matériau semi-conducteur qui va transformer l'énergie de la lumière absorbée en énergie électrique. Le principe de fonctionnement est basé sur les propriétés d'absorption des rayons de la lumière par les matériaux semi-conducteurs. Le choix des matériaux utilisés pour la construction des cellules photovoltaïques se fait en fonction des propriétés physiques des électrons qui sont susceptibles d'être libérés de leurs atomes lorsqu'ils sont excités par des photons provenant du spectre solaire.

Une fois libérés, ces électrons forment un courant électrique continu, en se déplaçant dans le matériau, ce qui génère une force électromotrice aux bornes de la cellule. Ce phénomène physique est appelé « effet photovoltaïque ». La figure.1.1 illustre la constitution et le principe d'une cellule photovoltaïque en silicium.

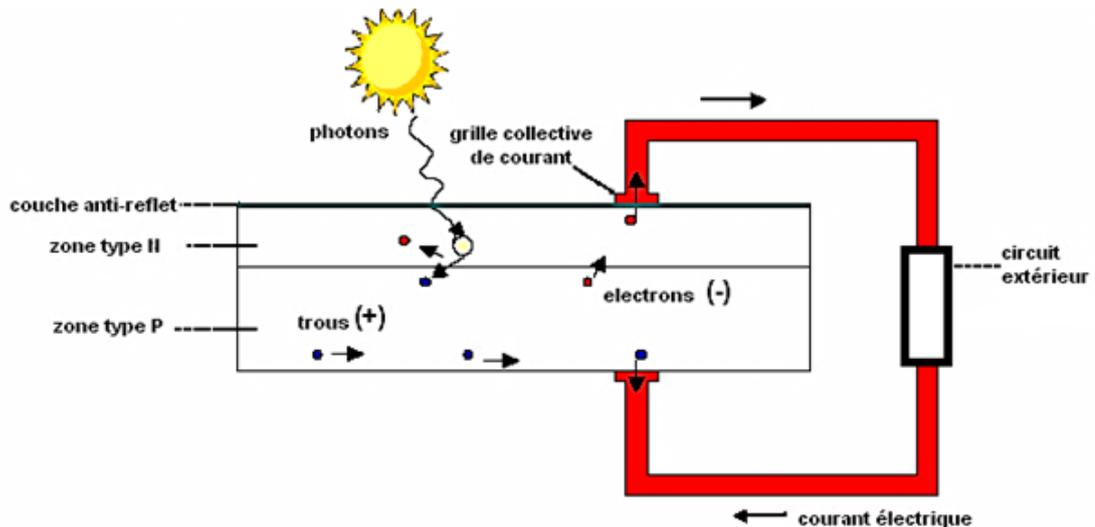


Figure.1.1 Constitution et principe d'une cellule photovoltaïque.

Comparable à une diode utilisée classiquement en électronique, une cellule photovoltaïque peut être réalisée à partir de deux couches de silicium, une dopée au bore (P) et l'autre dopée au phosphore (N) (Figure.1.1.). Entre les deux zones se développent une jonction PN avec une barrière de potentiel. La zone N est couverte par une grille métallique qui sert de cathode (contact avant) et surtout de collecteurs d'électrons, tandis qu'une couche métallique (contact arrière) recouvre l'autre face du cristal et joue le rôle d'anode. Lorsque les photons sont absorbés par le semi-conducteur, ils transmettent leur énergie aux électrons par collision. Si l'énergie transmise est supérieure à celle associée à la bande interdite (E_g) du semi-conducteur, des paires électrons-trous sont alors créés dans cette zone par arrachement des électrons.

Sous l'effet d'un champ électrique qui règne dans cette zone, ces porteurs libres sont drainés vers les contacts métalliques des régions P et N. Il en résulte alors un courant électrique dans la cellule PV et une différence de potentiel (de 0.6 à 0.8 Volt) supportée entre les électrodes métalliques de la cellule, communément nommée tension de circuit ouvert (V_{co}), fortement dépendante de la température.

Le courant maximal I_{max} se produit lorsque les bornes de la cellule sont court-circuitées. On parle alors de courant de court-circuit (I_{cc}) dépendant fortement du niveau d'éclairement (E).

1.2.2 Caractéristique électrique d'une cellule photovoltaïque

Une cellule photovoltaïque possède une caractéristique $I(V)$ non linéaire. La tension en circuit ouvert (V_{oc}) d'une cellule PV est comprise entre 0,3V et 0,7V selon le matériau semi-conducteur utilisé, la température de la cellule et son état de vieillissement. Alors que son courant de court-circuit (I_{cc}) varie principalement selon le niveau d'irradiation solaire et selon les technologies et les tailles des cellules considérées.

La figure.1.2 illustre la caractéristique non linéaire $I(V)$ d'une cellule PV à base de silicium pour un éclairement et une température donnée, avec la présence d'un point de puissance maximal (PPM) caractérisé par sa tension et son courant optimum (V_{opt} et I_{opt}).

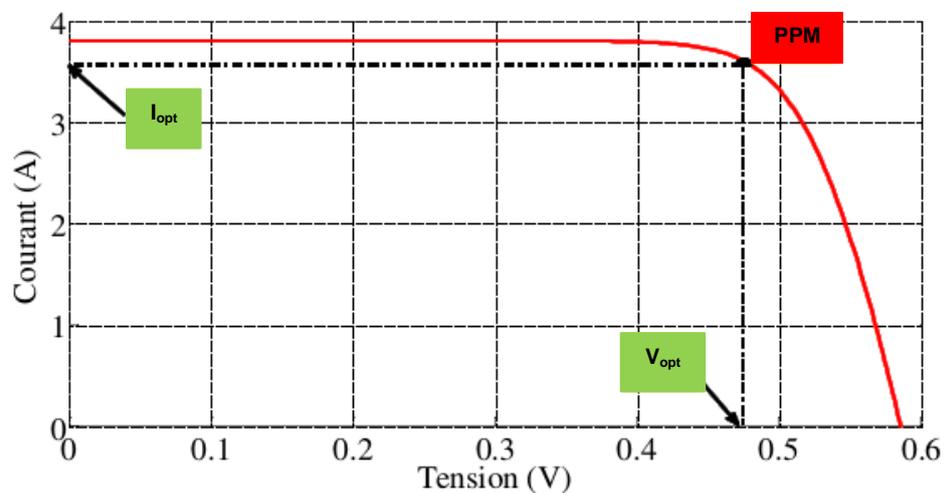


Figure.1.2 Caractéristique $I(V)$ d'une cellule photovoltaïque en Silicium.

Etant donné que la puissance de la cellule est le produit du courant par la tension, alors de même, on peut tracer la caractéristique $P-V$ de la cellule PV (Figure.1.3).

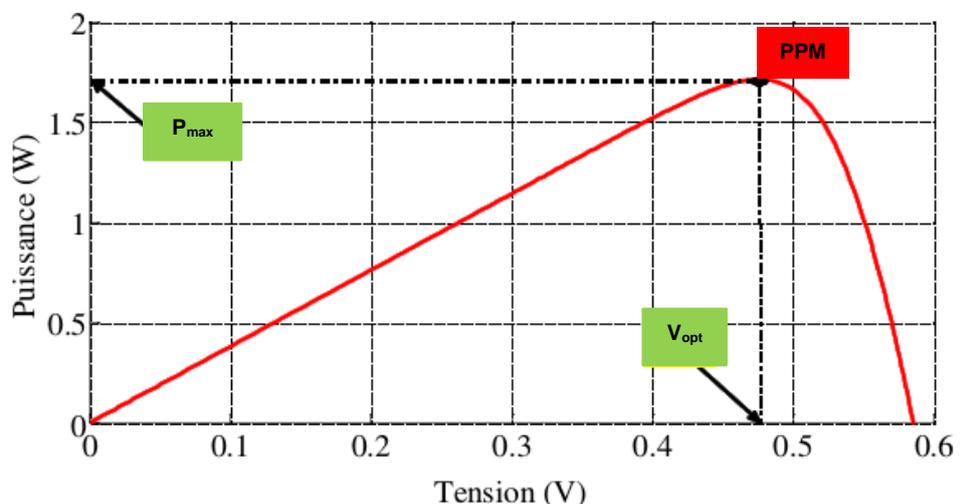


Figure.1.3 Caractéristique $P(V)$ d'une cellule photovoltaïque en Silicium.

1.2.3 Modélisation d'une cellule photovoltaïque

De nombreux modèles mathématiques de cellule photovoltaïque sont proposés dans la littérature [3,4]. Ceux-ci ont été développés dans le but d'obtenir les caractéristique courant-tension servant à l'analyse et l'évaluation des performances des systèmes photovoltaïques.

La figure.1.4 propose le modèle électrique d'une cellule photovoltaïque à deux diodes qui est largement utilisé dans la littérature [5,6].

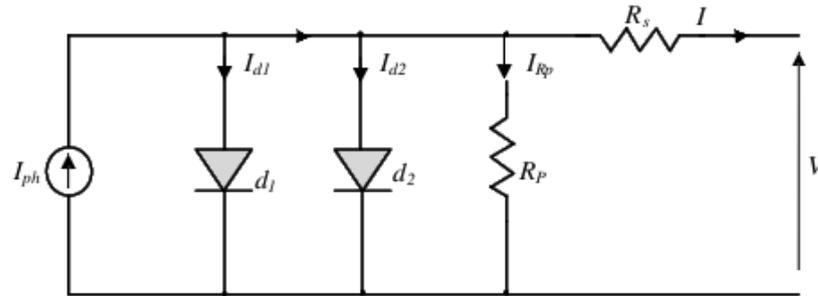


Figure.1.4 Circuit équivalent d'une cellule photovoltaïque à deux diodes.

Le modèle est constitué de deux diodes de courants de saturation I_{d1} et I_{d2} montées en parallèle, d'une source de courant qui va générer le courant de court-circuit de la cellule dépendant de l'éclairement solaire. La résistance série R_s tient compte de la résistivité du matériau et du contact semi-conducteur-métal. Sa valeur peut être déterminée par l'inverse de la pente de la caractéristique $I(V)$ pour la tension à circuit ouvert V_{co} . La résistance parallèle R_p traduit la présence d'un courant de fuite dans la jonction.

L'expression mathématique de la caractéristique courant-tension (I-V) d'une cellule solaire en se référant au schéma équivalent de la figure 4 est donnée par [7]:

$$I = I_{ph} - I_{d1} \left[e^{\frac{q(V+IR_s)}{n_1 kT}} - 1 \right] - I_{d2} \left[e^{\frac{q(V+IR_s)}{n_2 kT}} - 1 \right] - \frac{V+IR_s}{R_p} \quad (1)$$

$$I_{ph}(T) = I_{ph} |_{(T=298 K)} [1 + (T - 298. K)(5. 10^{-4})] \quad (2)$$

$$I_{d1} = K_1 T^3 e^{-\frac{E_g}{kT}} \quad (3)$$

$$I_{d2} = K_2 T^{5/2} e^{-\frac{E_g}{kT}} \quad (4)$$

1.3 LE GÉNÉRATEUR PHOTOVOLTAÏQUE

1.3.1 Structure d'un générateur photovoltaïque

Dans des conditions d'ensoleillement standard (1000W/m^2 ; 25°C), la puissance maximale délivrée par une cellule silicium de 150 cm^2 est d'environ 2.3 W sous une tension de 0.5V . Cette faible puissance est insuffisante pour la plupart des applications domestiques ou industrielles. Pour produire davantage de puissance, plusieurs cellules doivent être assemblées afin de créer un module photovoltaïque complet. Ainsi, un générateur photovoltaïque est donc constitué d'un réseau série/parallèle de plusieurs modules photovoltaïques, regroupés en panneaux, constitués de modules identiques.

Dans un groupement de n_s cellules en série, la caractéristique résultante du groupement est obtenue par addition des tensions élémentaires de chaque cellule, alors que le courant traversant les cellules reste le même.

La figure.1.5 illustre la caractéristique résultante,

avec $I_{scc} = I_{cc}$ et $V_{sco} = n_s \cdot V_{co}$.

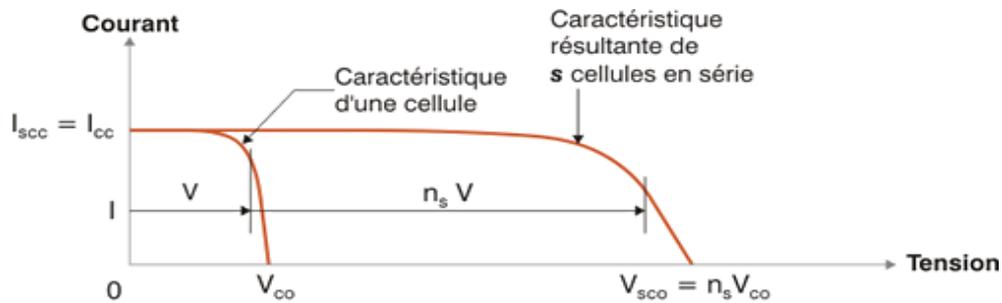


Figure.1.5 Caractéristique résultante d'un groupement de n_s cellules en série.

Dans le cas d'un groupement de n_p cellules en parallèle, les cellules sont soumises à la même tension et la caractéristique résultante du groupement est obtenue par l'addition des courants.

La figure.1.6 illustre la caractéristique résultante avec $I_{pcc} = n_p \cdot I_{cc}$ et $V_{pco} = V_{co}$.

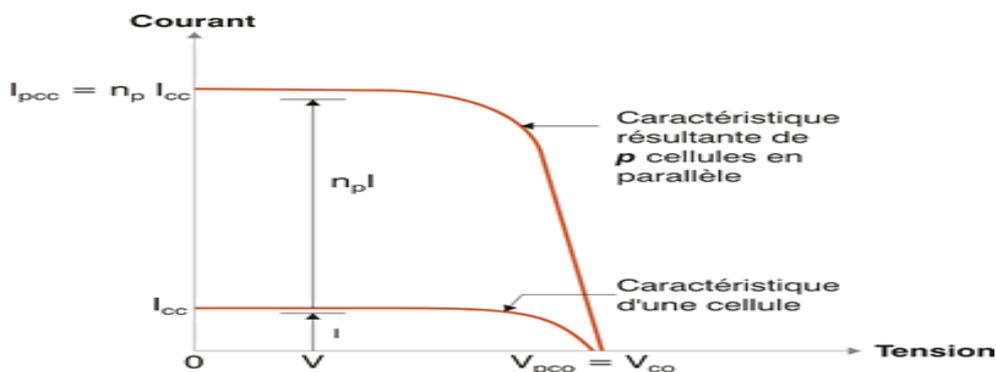


Figure.1.6 Caractéristique résultante d'un groupement de n_p cellules en parallèle.

1.3.2 Caractéristique Courant-Tension et Puissance-Tension

1.3.2.1 Influence de l'éclairement sur les courbes I-V et P-V

La caractéristique I-V d'une cellule PV est directement dépendante de l'éclairement et de la température ce qui implique que la puissance délivrée par un module PV en dépend également.

La figure.1.7 illustre les caractéristiques I-V et P-V pour différents niveaux d'éclairements à une température donnée de 25C°.

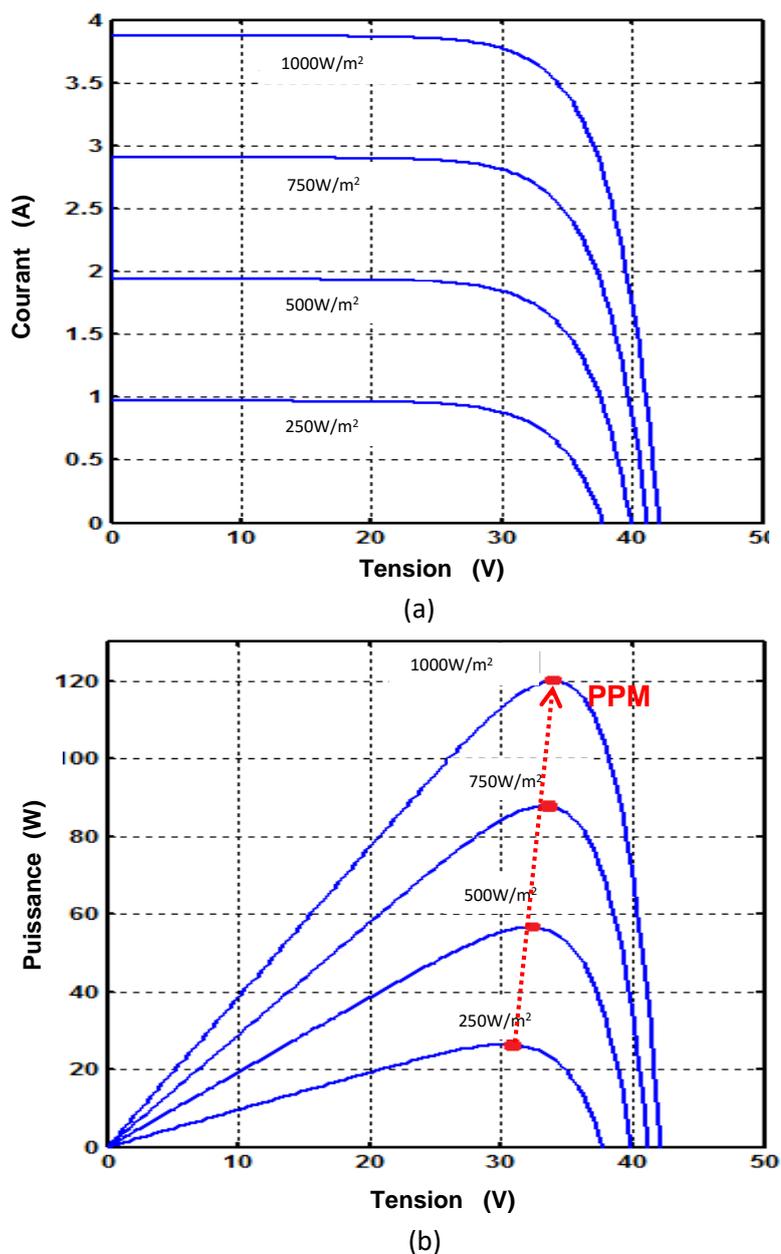


Figure.1.7 Influence de l'éclairement sur les courbes : (a) Courant-Tension (I-V)
(b) Puissance-Tension (P-V).

Il est clair que la valeur du courant de court-circuit est directement proportionnelle à l'intensité du rayonnement. Par contre, la tension en circuit ouvert ne varie pas dans les mêmes proportions, mais reste quasiment identique même à faible éclairement.

Ceci implique que :

- La puissance maximale (PM) de la cellule est proportionnelle à l'éclairement ;
- Les points de puissance maximale se situent quasiment à la même tension.

1.3.2.2 Influence de la température sur les courbes I-V et P-V

La température a également une influence sur la caractéristique courant/tension. Nous remarquons que la température a une influence négligeable sur la valeur du courant de court-circuit. En revanche, la tension en circuit ouvert baisse considérablement avec l'augmentation de la température, par conséquent la puissance extractible diminue.

De ce fait, il est important de prendre en considération la variation de la température du site lors du dimensionnement de l'installation. Ceci est dû au fait que la puissance du module constitué de cellules en Silicium mono ou polycristallin diminue environ de 0,5% par chaque degré d'augmentation de la température de la cellule au-dessus de 25°C.

La figure.1.8 illustre les caractéristiques I-V et P-V pour différentes valeurs de la Température pour un éclairement donné de 1000W/m².

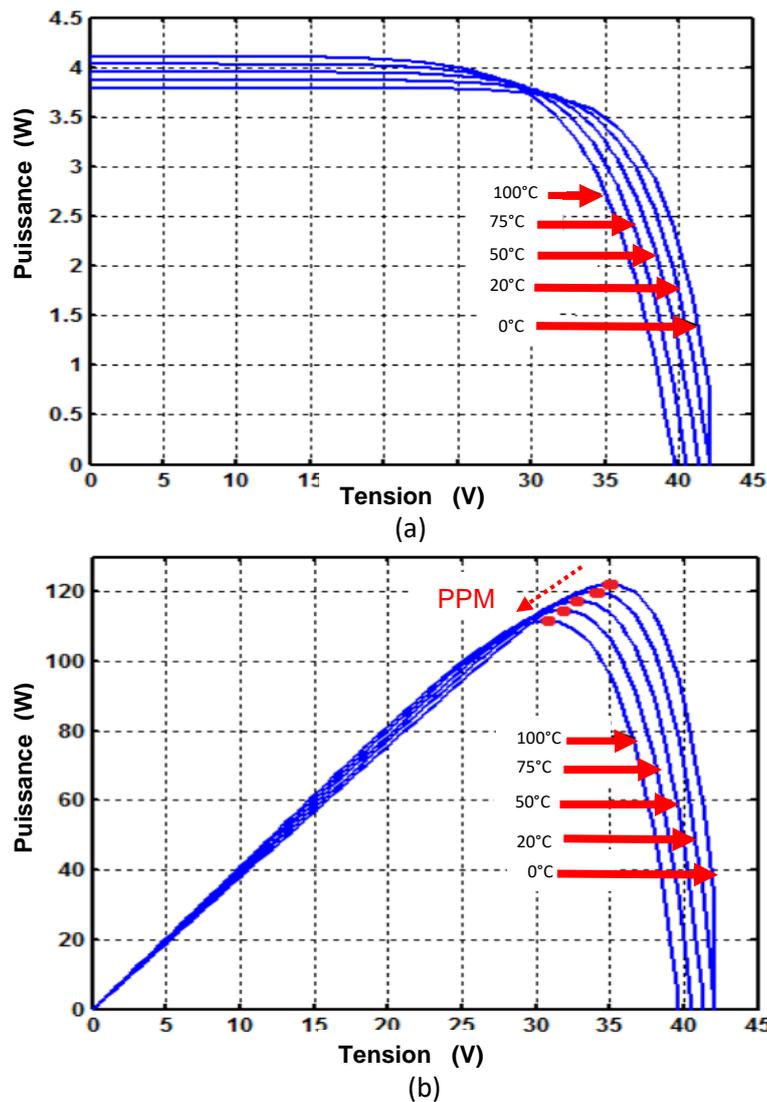


Figure.1.8 Influence de la température sur les courbes : (a) Courant-Tension (I-V)
(b) Puissance-Tension (P-V).

1.3.2.3 Influence d'un éclairage non homogène : Ombrage partiel

Lorsque les cellules d'un module PV sont éclairées de manière non homogène causé par l'ombrage dû à la présence à proximité d'arbres, d'immeubles, ou au passage de nuages ou d'oiseaux, ou dans le cas de vieillissement des cellules ou des erreurs de fabrication ; le module photovoltaïque réagit de manière très sensible à ces phénomènes.

En effet, l'ombrage altère grandement la production globale d'un système photovoltaïque [8-12]. Dans ce cas, l'allure des courbes caractéristiques I-V et P-V change : elle présente des maxima locaux. La recherche du Point de Puissance Maximale (PPM) devient plus compliquée. De plus, le rendement baisse fortement au-delà d'un certain pourcentage de la surface ombragée [13].

A titre d'illustration, on considère un générateur PV composé de quatre modules type MSX 120. Chacun est capable de fournir une puissance crête de 120 W dans des conditions standard de test STC (25 °C, 1000 W/m²). On fixe la température à la valeur de référence et on met les quatre modules à différents éclairagements selon trois scénarios d'ombrage (SP) :

- SP1 : Le premier est soumis à 800 W/m², le second à 500 W/m², le troisième à 300 W/m² et le dernier à 100 W/m².
- SP2 : Le premier est soumis à 800 W/m², le second à 200 W/m², le troisième à 200 W/m² et le dernier à 800 W/m².
- SP3 : Le premier est soumis à 1000 W/m², le second à 800 W/m², le troisième à 700 W/m² et le dernier à 600 W/m².

L'influence de l'ensoleillement non homogène sur les caractéristiques P-V est illustrée dans la figure.1.9. On constate l'existence de plusieurs maxima locaux et un maxima global selon le scénario considéré.

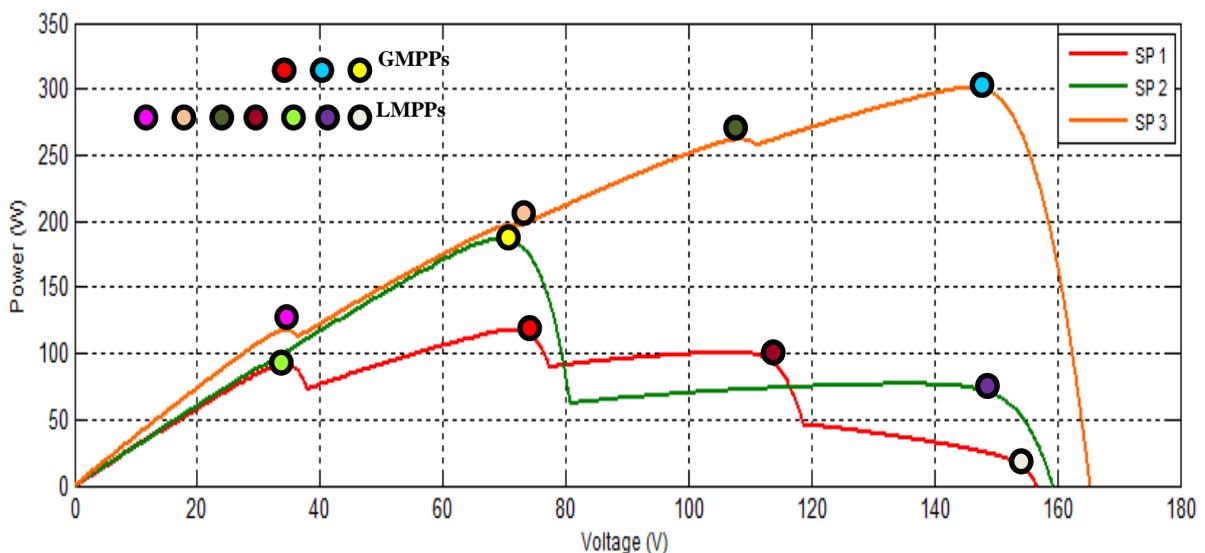


Figure.1.9 Caractéristique P-V pour différents scénarios d'ombrage.

1.4 CONNEXION GÉNÉRATEUR PHOTOVOLTAÏQUE-CHARGE

1.4.1 Connexion directe

Au début, l'exploitation du solaire photovoltaïque se faisait en connexion directe (Figure.1.10). De ce fait, le point de fonctionnement du générateur PV dépendait de l'impédance de la charge à laquelle il est connecté. En d'autres termes, il a une tension V_{opt} fixée par la tension nominale de l'application.

Ce choix se justifie principalement par la simplicité, la fiabilité et le faible coût. Dans cette configuration, la présence de la diode anti retour est indispensable pour empêcher la circulation d'un courant négatif vers les modules PV.

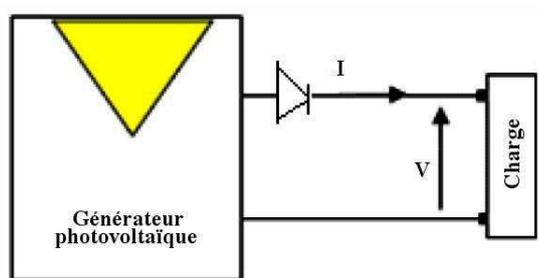


Figure.1.10 Connexion directe GPV-Charge.

Néanmoins, cette configuration ne permet pas de limitation et/ou de réglage de la tension de la charge. Ceci explique le fait qu'elle est restée restreinte à des applications nécessitant une tension fixe ou faiblement variable (cas des batteries de stockage dans les systèmes isolés).

Plus généralement, la puissance extraite d'un générateur PV connecté directement à une application est souvent très éloignée du PPM que peut délivrer le générateur [14,15].

1.4.2 Connexion indirecte via un étage d'adaptation

Afin de remédier à la problématique suscitée, il s'est avéré nécessaire de rajouter un étage d'adaptation, comme indiqué sur la figure.1.11. Ce dernier, garantit le transfert d'énergie et qu'il s'effectue dans des conditions de fonctionnement optimales pour le générateur PV et la charge.

Pour que l'ensemble fonctionne de manière idéale, diverses boucles de contrôle en entrée et en sortie de l'étage d'adaptation sont nécessaires :

- En entrée, elles garantissent l'extraction du PPM disponible aux bornes du générateur PV, à chaque instant ;
- En sortie, elles permettent un fonctionnement optimal de chaque application dans son mode le plus approprié.

Les techniques utilisées pour les boucles de contrôle en entrée consistent à associer à l'étage d'adaptation une commande appelée MPPT ou Maximum Power Point Tracking, qui effectue une recherche permanente du point de puissance maximale PPM.

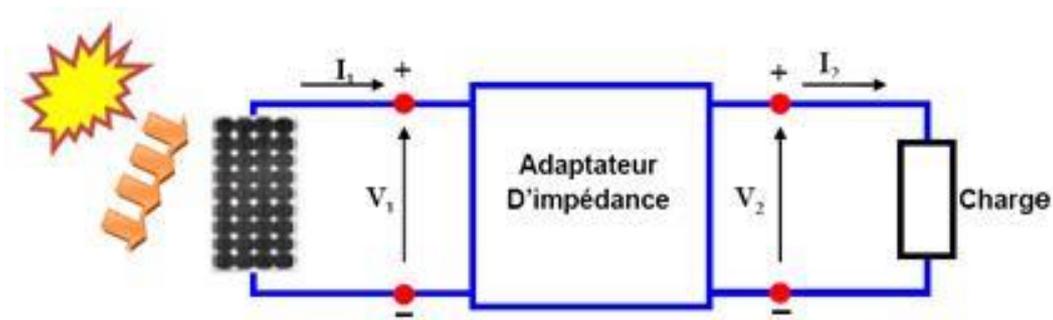


Figure.1.11 Connexion d'un GPV à une charge à travers un étage d'adaptation.

1.5 PRINCIPE DE RECHERCHE DU POINT DE PUISSANCE MAXIMALE

Afin de permettre à un système PV de fonctionner continuellement à des PPM, il est nécessaire de prévoir l'introduction de commandes spécifiques dits de recherche du point de puissance maximale PPM ou « Maximum Power Point Tracking – MPPT ». Leur principe est simple : il consiste à rechercher constamment le point de puissance maximale tout en gardant une adaptation acceptable entre le générateur et sa charge permettant ainsi de transférer tout le temps le maximum de puissance.

La figure.1.12 reprend le schéma de la figure 1.11 précédente dans laquelle il a été procédé à l'ajout d'une commande MPPT associée à un convertisseur statique permettant de faire une adaptation entre le générateur PV et la charge de telle sorte que la puissance générée corresponde à sa valeur maximale et qu'elle soit transférée directement à la charge.

Il est à noter que plusieurs types de convertisseurs DC/DC peuvent être utilisés : sans isolation galvanique comme Buck (dévolteur), Boost (survolteur), Buck-Boost (dévolteur - survolteur), Cuck ou Sepic ; ou avec isolation galvanique comme Flyback ou Forward.

La technique de contrôle introduite grâce au MPPT consiste à agir automatiquement sur le rapport cyclique de façon à amener le point de fonctionnement du générateur à sa valeur optimale de puissance quelque soient les conditions atmosphériques et/ou les variations brusques de la charge.

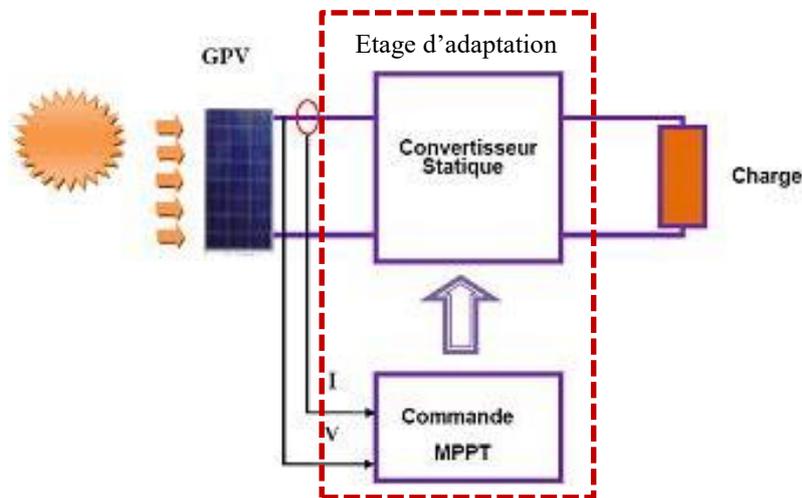


Figure.1.12 Chaîne de conversion photovoltaïque avec convertisseur statique contrôlée par une commande MPPT.

De ce fait, la commande MPPT fait varier le rapport cyclique du convertisseur (CS) de telle sorte que la puissance fournie par le GPV soit maximale à ses bornes. En général, il est basé sur la variation du rapport cyclique du CS jusqu'à se placer sur le PPM en fonction des évolutions des paramètres d'entrée du CS (I_{PV} et V_{PV}).

Dans le cadre de nos travaux dont l'un des objectifs porte sur l'optimisation de puissance dans une chaîne de conversion PV, nous avons opté, comme nous le verrons dans les chapitres suivants, pour une configuration réduite qui consiste en un générateur PV débitant sur une charge par l'intermédiaire d'un convertisseur comme l'illustre la figure.1.12.

1.6 CONCLUSION

Ce chapitre introductif a été consacré d'abord à la présentation, dans le détail, des concepts de base des systèmes photovoltaïques, en expliquant l'origine physique de l'effet photovoltaïque et l'interaction photon-électron. Ensuite, il a été procédé à la description des principaux modules qui composent ces systèmes ainsi que l'impact des conditions atmosphériques sur les caractéristiques du principal module en l'occurrence le générateur photovoltaïque et ceci en

fonction de l'éclairement et de la température qui agissent directement sur la puissance délivrée par celui-ci.

Par ailleurs, il a été démontré que la mise en œuvre d'une chaîne de conversion était plus que nécessaire pour exploiter au mieux l'énergie délivrée par le générateur photovoltaïque. Chaque élément de cette chaîne a un rôle bien défini pour le fonctionnement désiré selon le cahier des charges de l'utilisateur. Exploiter « au mieux » cette énergie délivrée reviendrait à rechercher continuellement le point de puissance maximale.

C'est ainsi que dans le prochain chapitre, un état de l'art sur les méthodes de recherche du point de puissance maximale sera présenté.

Chapitre 2

**LES MÉTHODES DE RECHERCHE DU
POINT DE PUISSANCE MAXIMALE MPPT :
ÉTAT DE L'ART**

2.1 INTRODUCTION

Au cours de ce chapitre, nous allons présenter un état de l'art des méthodes de recherche du point de puissance maximale MPPT à l'issue duquel, nous proposerons une nouvelle approche de classification des commandes MPPT. Nous allons discuter des principaux critères permettant la sélection d'un contrôleur. Aussi, les concepts de base des méthodes bio inspirées seront introduits.

A l'issue de ce chapitre, nous serons en mesure de choisir deux méthodes pour l'implémentation des contrôleurs MPPT à travers la revue de la classification des travaux existants dans la littérature et la sélection de deux algorithmes performants utilisés dans l'élaboration de contrôleurs MPPT.

2.2 LES MÉTHODES DE RECHERCHE DU POINT DE PUISSANCE MAXIMALE MPPT

C'est en janvier 1968, que *A.F. Boehringer* [1] a publié la première loi de commande MPPT adaptée à une source d'énergie renouvelable de type photovoltaïque. Depuis, plusieurs travaux de recherches traitant des contrôleurs MPPT ont été menés. Ces contrôleurs diffèrent selon les critères, par leur principe de fonctionnement, leur précision et leur rapidité de recherche, etc... Malheureusement, à ce jour, il n'existe toujours pas de standard qui puisse définir comment quantifier les performances d'un contrôleur MPPT face aux critères de sélection, ni comment classer ces derniers selon les méthodes utilisées. Le choix de l'une d'entre elles reste éprouvant puisque chacune présente des avantages et des inconvénients.

Dans le cadre de nos travaux de recherche, nous proposons une nouvelle approche de classification des contrôleurs MPPT selon les méthodes existantes, ainsi qu'une classification permettant de mettre en exergue les critères de performances qui nous semblent être essentiels pour la conception d'un contrôleur MPPT.

2.2.1 Critères de sélection des commandes MPPT

Le problème fondamental abordé par les techniques MPPT est de déterminer automatiquement la tension de sortie et le courant photovoltaïque pour lequel le générateur PV produit sa puissance de sortie maximale sous diverses conditions atmosphériques. Malgré le fait que ces techniques sont conçues pour les mêmes objectifs, ils diffèrent considérablement en termes de *complexité*, de *flexibilité*, du *nombre de capteurs nécessaires*, de *robustesse*, de *vitesse de convergence*, et de *l'efficacité*.

De plus, il est à retenir que la recherche du point de puissance maximale (PPM) peut parfois être une tâche difficile, surtout lorsque le système PV ne reçoit pas un rayonnement uniforme, comme dans le cas de l'ombrage partiel (Partial Shading). En effet, ce type de problèmes a été sérieusement pris en considération au cours de ces dernières années et continue de susciter un grand intérêt au sein de la communauté des chercheurs dans le domaine des énergies renouvelables, ainsi que le monde des industriels, car il représente le moyen le plus économique visant à améliorer l'efficacité de l'ensemble du système.

La Figure.2.1, ci-après illustre une sélection des principaux critères que les concepteurs doivent prendre en considération pour l'élaboration de contrôleurs MPPT.

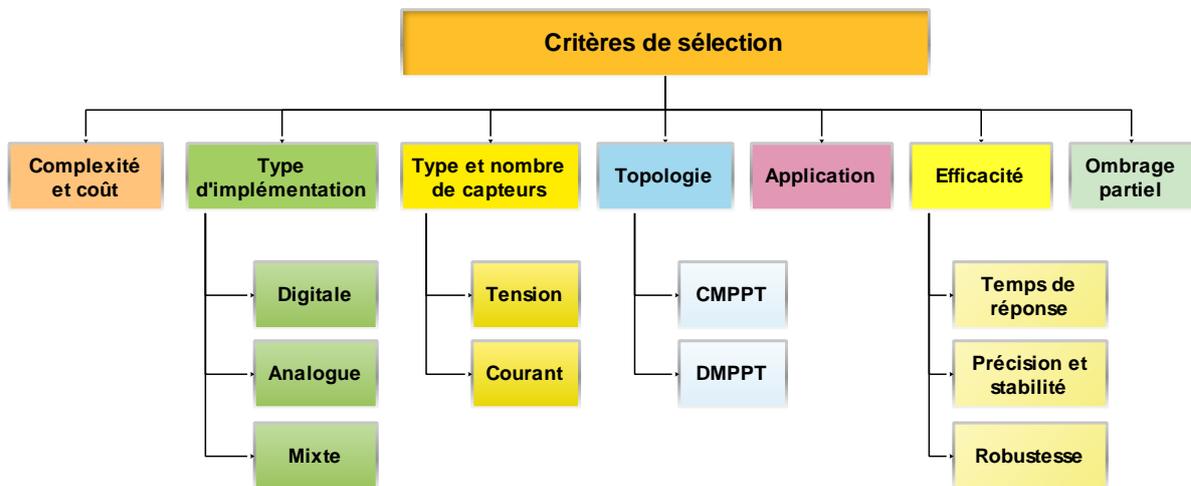


Figure.2.1 Classification des critères de sélection des contrôleurs MPPT.

Comme illustré dans la Figure.2.1, ces contrôleurs MPPT dépendent de :

- **La complexité et le coût :**

Le coût d'une commande MPPT dépend de sa complexité de mise en œuvre : approche analogique ou numérique, nécessitant ou pas un logiciel de programmation, type et nombre de capteurs utilisés. En règle générale, la notion de coût d'une commande MPPT revient toujours au type d'implémentation (analogique ou numérique) ; il est souvent cité dans la littérature que les implémentations analogiques sont moins coûteuses, vu que les implémentations numériques nécessitent des microcontrôleurs programmables.

Ainsi, l'élimination de certains capteurs peut diminuer substantiellement le coût d'une commande MPPT, en particulier les capteurs de courant.

- ***L'implémentation :***

Les commandes MPPT peuvent être implantées en analogique, en numérique ou de manière mixte (les deux en même temps). Cela a une influence directe sur la simplicité de l'implémentation de l'algorithme et donc, sur le coût de la commande MPPT et de son rendement.

- ***Le type et le nombre de capteurs nécessaires :***

Le nombre requis de capteurs dépend de la facilité d'utilisation. Il est plus facile, plus fiable et plus économique d'utiliser des capteurs de tension que des capteurs de courant, souvent plus coûteux et plus difficiles à mettre en œuvre.

- ***La topologie :***

Généralement, il existe deux configurations de contrôleurs MPPT : la configuration centralisée (CMPPT) et la configuration distribuée (DMPPT). La première signifie que l'ensemble du système photovoltaïque est contrôlé par une seule commande MPPT, alors que pour la seconde, chaque module photovoltaïque possède sa propre commande MPPT.

- ***L'application :***

La commande MPPT est adaptée pour chaque application en tenant compte des contraintes : par exemple, dans les stations orbitales des satellites, le coût et la complexité de l'algorithme ne sont pas aussi importants que la performance. Elle doit être en mesure de suivre en permanence le MPP.

- ***L'ombrage partiel - Partial Shadding (PS) :***

Les systèmes photovoltaïques sont extrêmement sensibles à l'ombrage partiel (PS). Si un module d'une chaîne est partiellement ombragé, les cellules ombragées du module vont absorber la puissance des cellules non ombragées. De ce fait, au lieu de contribuer à la génération de puissance de sortie, ces derniers agissent comme une charge.

- ***L'efficacité :***

C'est une caractéristique importante pour la sélection d'une commande MPPT. Elle dépend principalement du *temps de réponse*, de la *précision*, de la *stabilité* et de la *robustesse*.

2.2.2 Classification des méthodes MPPT

Depuis la parution de la première commande en 1968, divers travaux sur les contrôleurs MPPT ont été menés afin d'accroître continuellement l'efficacité des systèmes photovoltaïques. Vu le grand nombre de publications dans ce domaine, des travaux de synthèses relatifs aux différents contrôleurs MPPT ont été présentés dans [16-21].

Globalement, ces contrôleurs MPPT peuvent être regroupés sous deux grandes catégories : les *méthodes conventionnelles* (MC) et les *méthodes intelligentes* (MI) et ce, malgré le fait que plusieurs approches pour la classification des contrôleurs MPPT aient été proposées [22-25].

Une première proposition de classification des contrôleurs MPPT au niveau du laboratoire LDCCP (*Laboratoire des Dispositifs de Communications et de Conversion Photovoltaïque, ENP*) a été proposée par K.Sobaihi [22]. Comme l'illustre la figure.2.2, l'auteur distingue les catégories de contrôleurs selon trois méthodes : celles à contre réaction de tension, à contre réaction de courant et celles à contre réaction de puissance.

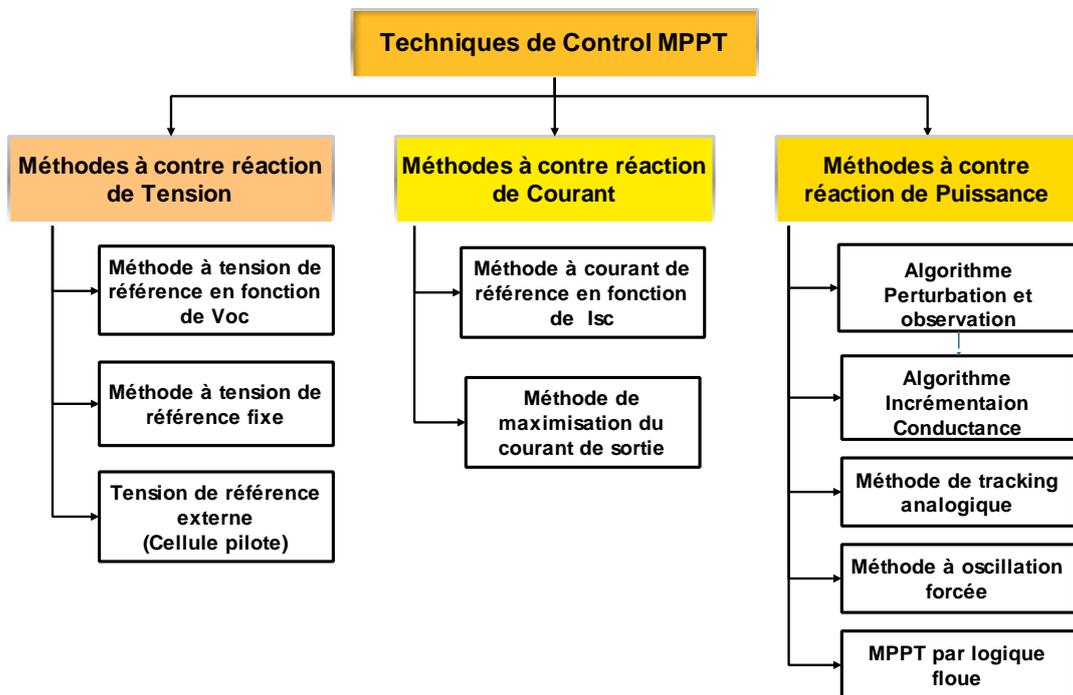


Figure.2.2 Classification des méthodes MPPT selon [22].

Pallavee Bhatnagar et al. [23], Priety et Y.Kumar [24] et A.Reza Reizu et Mohamed Hassan Moradi [25] ont proposé en 2013, une classification des contrôleurs MPPT ordonnés selon trois groupes distincts : les méthodes indirectes (*Offline*), les méthodes directe (*Online*) et les méthodes ayant recours à l'*intelligence artificielle*.

Les méthodes indirectes emploient habituellement les données techniques du panneau photovoltaïque pour estimer le PPM en incluant les informations préalables comme les courbes I(V) et P(V) à diverses conditions climatiques et les différents modèles mathématiques du panneau, ... etc.

Les méthodes directes emploient, en temps réel, les mesures de la tension et/ou du courant du panneau photovoltaïque afin de détecter le PPM. Ces méthodes ne tiennent pas compte des mesures de la température et de l'ensoleillement ni du modèle du panneau photovoltaïque.

Enfin, les méthodes qui ont recours à l'*intelligence artificielle* ne nécessitent pas un modèle mathématique et s'appliquent normalement aux systèmes non linéaires offrant ainsi une grande robustesse.

La figure.2.3 illustre la classification proposée par [23].

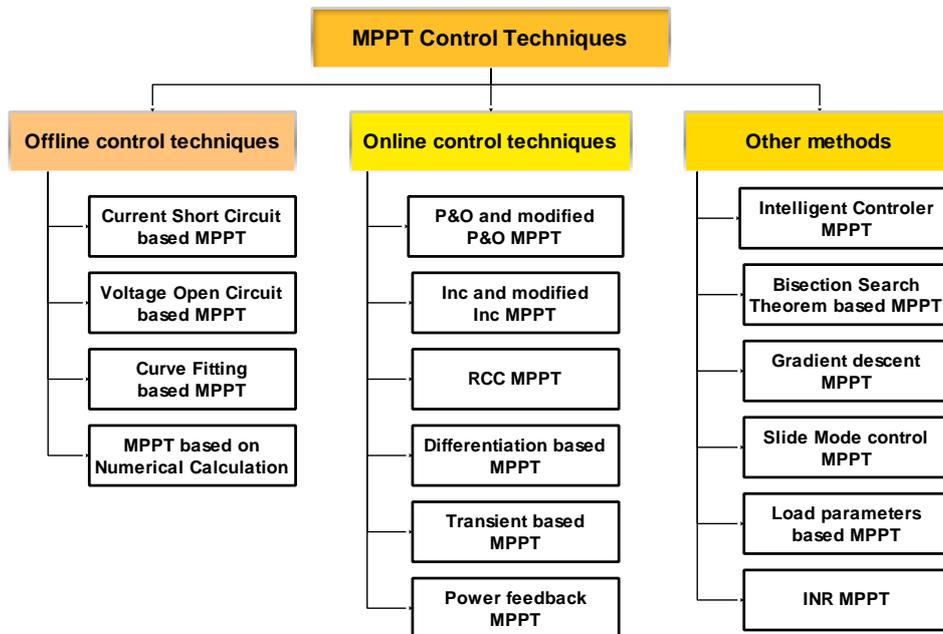


Figure.2.3 Classification des méthodes MPPT selon [23].

Comme l'illustre la figure.2.4, les auteurs A. Gupta et al.[26], ont proposé une autre approche de classification dans laquelle les contrôleurs MPPT sont catégorisés en trois principaux groupes sous d'autres appellations : les *méthodes conventionnelles* (MC), les *méthodes intelligentes* (MI) et les *méthodes hybrides* (MH).

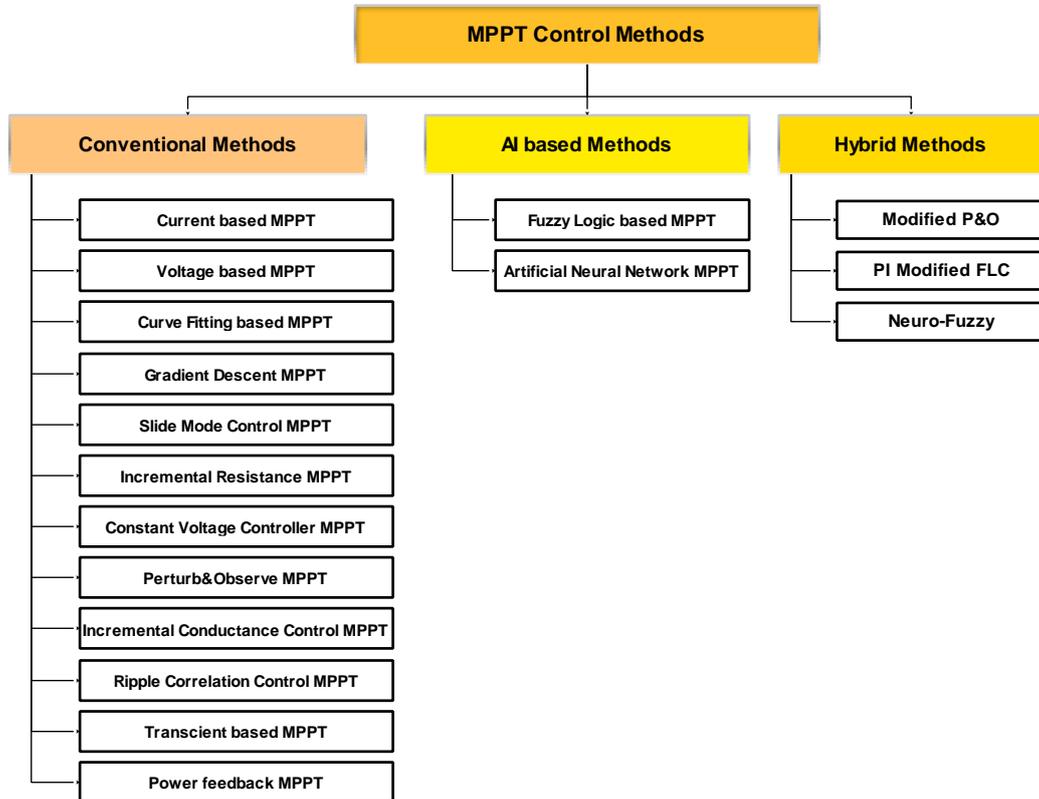


Figure.2.4 Classification des méthodes MPPT selon [26].

Par ailleurs, K.Ishaque et Z.Salam [27] ont proposé de classer les contrôleurs MPPT en deux grandes catégories : les *méthodes conventionnelles* (MC) et les *méthodes soft computing* (SC).

Alors que dans [28-31], S.Dileep, et al., M.Balamurugan, M.Sayedmahmoudian, Z.Salam et al., ont proposé une autre approche de classification des contrôleurs MPPT en tenant compte que des *méthodes intelligentes*. Ces dernières sont regroupées sous le nom de *méthodes soft computing* (SC).

La figure.2.5 illustre les classifications proposées par [28, 29,31].

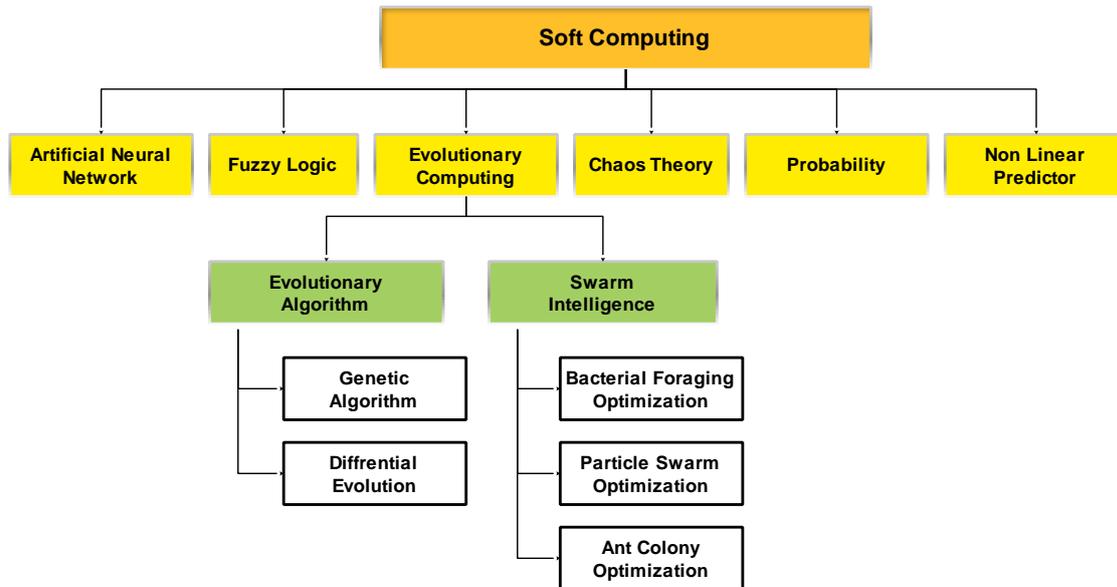


Figure.2.5 Classification des méthodes MPPT selon [28, 29,31].

Avec les différentes propositions existantes relatives à la classification des contrôleurs MPPT et l'évolution algorithmique lors de la dernière décennie, ayant permis le développement de nouveaux contrôleurs, il nous est apparu nécessaire de proposer une nouvelle approche de classification qui tiendrait compte de ces avancées technologiques importantes.

L'architecture proposée de cette nouvelle classification permettant de mettre en avant les contrôleurs MPPT, sera présentée dans la prochaine section.

2.3 PROPOSITION D'UNE NOUVELLE APPROCHE DE CLASSIFICATION DES CONTRÔEURS MMPT

Le grand nombre de nouveaux algorithmes introduits ces dernières années pour le contrôle MPPT a engendré une évolution drastique dans la gestion de la puissance des systèmes PV. A la base des différents travaux existant présentés en *section 2.2.2* relatifs à la classification de ces derniers, nous proposons, dans ce qui suit, une nouvelle approche de classification qui prend en considération l'évolution algorithmique appliquée aux contrôleurs MPPT.

Comme l'illustre la figure.2.6 la nouvelle classification que nous proposons va permettre d'avoir une vue globale des contrôleurs MPPT en partant des méthodes simples, plus communément appelées *méthodes conventionnelles* (MC) jusqu'à arriver aux *méthodes soft computing* (MSC) qui intègrent les *méthodes d'intelligence artificielle* (MIA) ainsi que les méthodes nouvellement introduites, connues sous le nom des *méthodes bio inspirées* (MBI).

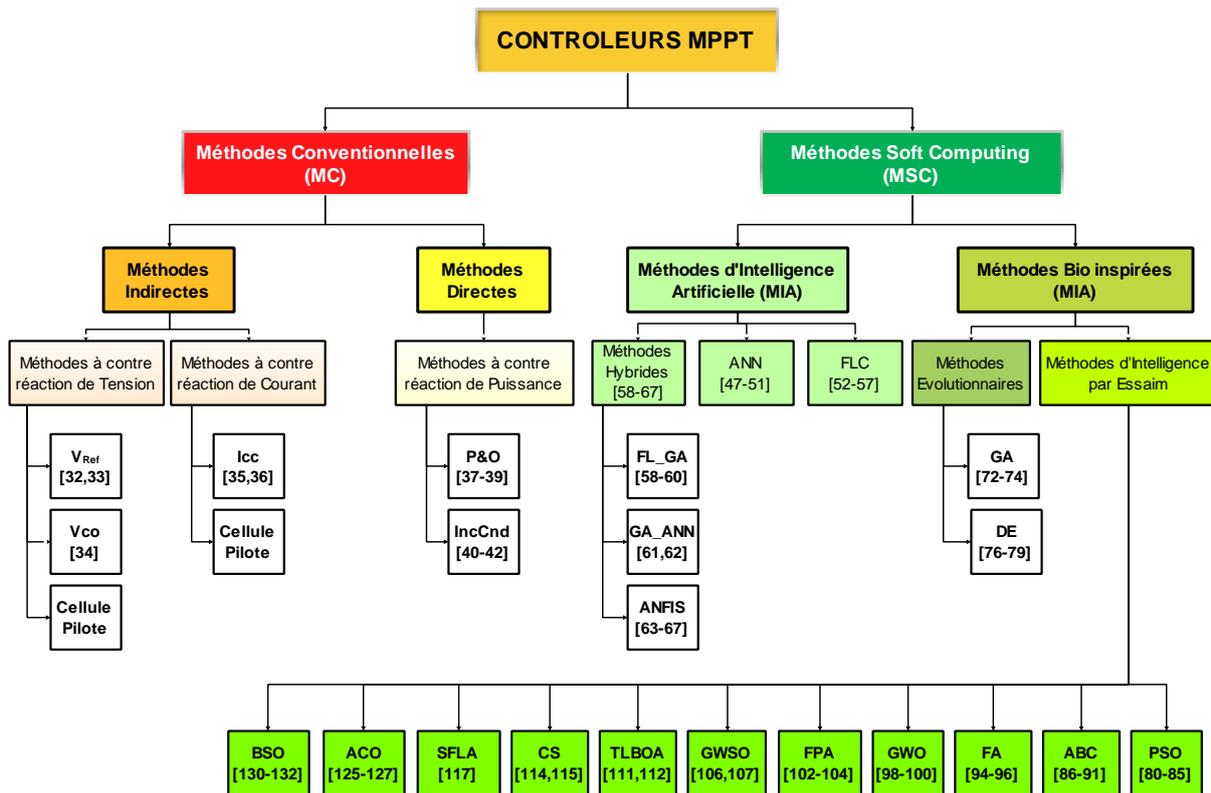


Figure.2.6 Classification des méthodes MPPT proposée.

Dans ce qui suit, nous allons présenter chacune de ses méthodes.

2.4 LES MÉTHODES CONVENTIONNELLES (MC)

Les premiers contrôleurs MPPT à être développés sont issus des méthodes conventionnelles. Ces derniers sont subdivisés en deux grandes catégories : les *méthodes indirectes* (MI) et les *méthodes directes* (MD).

Bien qu'à ce jour un grand nombre de publications scientifiques introduisent de nouveaux contrôleurs MPPT basés sur ces méthodes, dans ce qui suit, nous nous intéresserons aux contrôleurs les plus populaires et les plus largement utilisés de par leur simplicité d'implémentation.

2.4.1 Les méthodes indirectes (MI)

Les contrôleurs issus de ces méthodes sont basés sur l'utilisation des données incluant les paramètres et les courbes spécifiques du générateur PV avec différentes valeurs de la température et de l'irradiation solaire, ou bien sur les modèles mathématiques obtenus à partir

de données empiriques pour estimer la position probable du point de puissance maximale recherché. Parmi les techniques issues de cette catégorie, nous avons :

2.4.1.1 Méthodes à contre réaction de tension

Ces méthodes sont simples et peu coûteuses. Leur principe repose sur le contrôle de la tension de fonctionnement du panneau PV en la comparant avec une tension de référence. Cela génère une tension d'erreur qui fait varier le rapport cyclique de la PWM de commande afin d'annuler cette erreur.

Selon la nature de cette tension de référence, trois types de méthodes existent : les méthodes à *tension de référence fixe*, les méthodes à *tension de référence en fonction de V_{co}* et les méthodes à *tension de référence externe* (cellule pilote).

2.4.1.1.1 Méthode à tension de référence fixe

Le principe de la méthode consiste à comparer la tension de sortie du panneau et une tension de référence fixe prédéfinie tout en ajustant continuellement le rapport cyclique (Figure.2.7). Cette tension correspond à la tension moyenne de l'intervalle des points de puissances maximales relevée par des tests avec diverses conditions de température et d'éclairement en agissant sur les facteurs de pondération lors de la mise au point. Cela permet ainsi de générer le maximum de puissance [32,33].

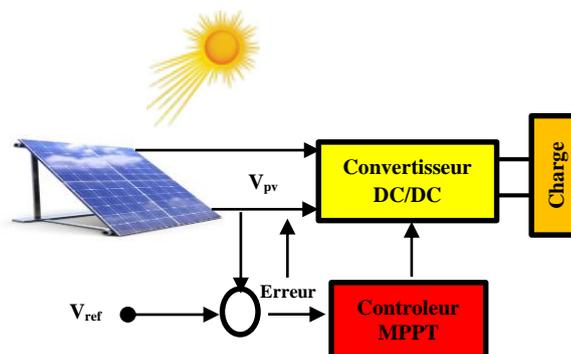


Figure.2.7 Système de contrôle MPPT à rétroaction de tension avec tension de référence fixe.

2.4.1.1.2 Méthode à tension de référence en fonction de V_{co}

Cette méthode de contrôle est différente de la précédente : en effet, pour un ensoleillement et une température donnés, la tension correspondante à la puissance maximale du panneau est

exprimée comme une fonction linéaire de la tension en circuit ouvert V_{co} du panneau (Figure.2.8).

Cette tension est prélevée régulièrement en débranchant le panneau pour une courte durée afin d'ajuster la tension de référence précédente à une certaine proportionnalité généralement égale à 0.7. En exploitant cette propriété, on peut traquer en permanence le PPM [34].

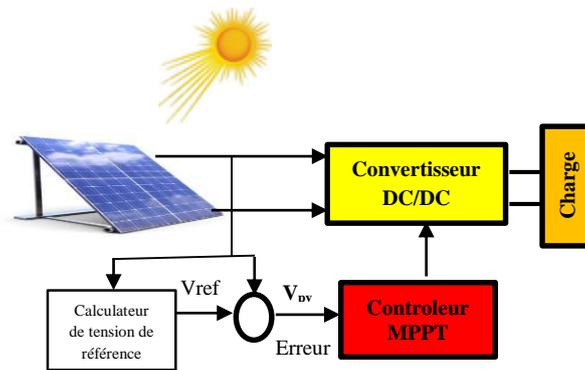


Figure.2.8 Système de contrôle MPPT à rétroaction de tension avec tension de référence contrôlable.

2.4.1.1.3 Méthode à tension de référence externe (cellule pilote)

Afin palier aux insuffisances de la méthode précédente (débranchant le panneau pour une courte durée), une cellule pilote est ajoutée au panneau solaire (il s'agit d'une cellule photovoltaïque simple, électriquement indépendante du reste de la rangée). La tension à circuit ouvert de cette cellule mesurée continuellement va fournir une information tacite de la tension en circuit ouvert de l'ensemble des panneaux solaires, en multipliant cette tension avec le nombre de cellules en série. Cette manière de faire permet d'éviter l'interruption du système (Figure.2.9).

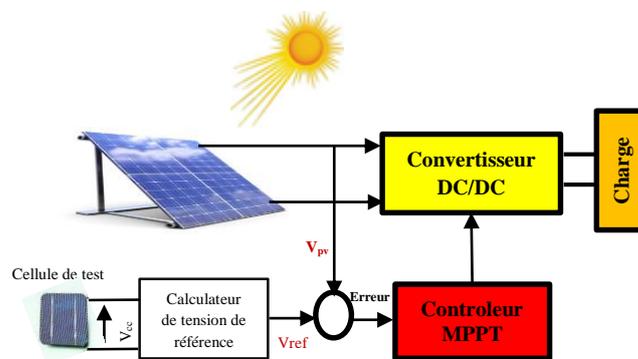


Figure.2.9 Système de contrôle MPPT à rétroaction de tension avec une cellule PV de référence.

2.4.1.2 Méthodes à contre réaction de courant

Par analogie, ces dernières sont subdivisées en méthodes de *référence en fonction du courant de court-circuit I_{sc}* et méthodes de *courant de référence externe* (cellule pilote).

2.4.1.2.1 Méthode à courant de référence en fonction du courant du court-circuit I_{sc}

Comme dans les méthodes à contre réaction de tension, le courant de court-circuit du panneau permet de récupérer la valeur du courant optimal I_{MPP} , correspondant au "PPM" du panneau. Le courant optimal est proportionnel au courant de court-circuit, généralement égale à 0.8. Comme le coefficient de proportionnalité est quasi constant en fonction des conditions d'ensoleillement et de température, le courant optimal est proportionnel au courant de court-circuit [35,36]. La figure.2.10 illustre la mise en œuvre d'un contrôleur MPPT à base du courant de court-circuit I_{sc} du panneau.

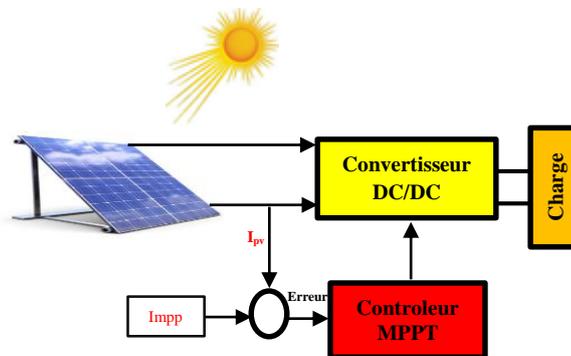


Figure.2.10 Mise en œuvre de la méthode MPPT basée sur la fraction du courant de court-circuit.

2.4.1.2.2 Méthode à courant de référence externe (cellule pilote)

L'utilisation d'une cellule pilote comme source d'information de courant de court-circuit de l'ensemble des panneaux n'est pas possible car court-circuiter en permanence cette dernière causera un échauffement supplémentaire. Ceci va altérer l'information générée par cette cellule et causer sa destruction rapide.

2.4.2 Les méthodes directes

2.4.2.1 Méthodes à contre réaction de puissance

Elles sont basées sur des algorithmes de recherche itérative visant à trouver le point de fonctionnement du panneau afin que la puissance générée soit maximale sans interruption du fonctionnement du système. Son principe de base est donc bâti autour de la mesure des deux composantes principales, le courant I_{PV} et la tension V_{PV} de sortie du panneau (Figure.2.11).

L'évaluation de la puissance instantanée de ce dernier est calculée par une simple multiplication des deux composantes $P_{PV} = V_{PV} * I_{PV}$ par un algorithme de recherche itératif qui va trouver et rectifier en temps réel la position du PPM, i.e. la puissance maximale, en effectuant la recherche dans un sens. L'algorithme maintient ce sens si la puissance de sortie augmente constamment et dans le cas contraire, il change le sens de recherche.

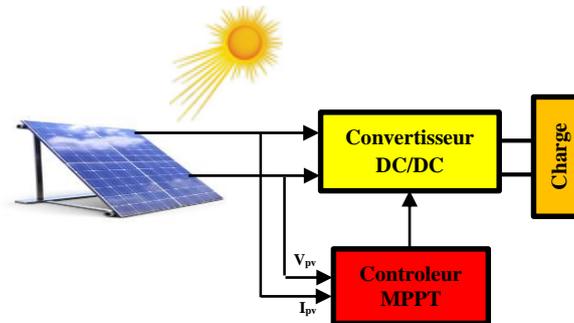


Figure.2.11 Schéma bloc du système de contrôle avec rétroaction de puissance.

Parmi les contrôleurs issus de ces méthodes, nous pouvons citer : le contrôleur le plus populaire et le plus largement utilisé de par sa simplicité d'implémentation à savoir le contrôleur *Perturbe and Observe*, et le contrôleur à base de l'*Incrémentacion de la Conductance (Incremental conductance)*.

Dans ce qui suit, nous allons donner un bref aperçu de chacun de ces contrôleurs. La méthode à base du contrôleur *Perturbe and Observe* servira comme modèle de référence lors des différentes simulations et études comparatives de nos prochains chapitres.

2.4.2.1.1 Méthode basée sur le contrôleur *Perturb and Observe (P&O_MPPT)*

Cette méthode basée sur le contrôleur *perturbe and observation (P&O)* est la plus utilisée en tant que commande "MPPT" [37-39]. Comme son nom l'indique, la méthode est basée sur une perturbation du système à travers une incrémentation ou une décrémentation de la tension de sortie du panneau ou en agissant directement sur le rapport cyclique du convertisseur DC-DC, suivi d'une observation des effets de ces perturbations sur la puissance de sortie du panneau.

Si la valeur de la puissance actuelle $P(k)$ du panneau est supérieure à la valeur précédente $P(k-1)$ alors on garde la même direction de la perturbation précédente sinon on inverse la perturbation du cycle précédent.

Lorsque le PPM est atteint, la tension V_{PV} oscille autour de la valeur optimale V_{OP} . Ceci cause une perte de puissance qui augmente avec la taille du pas de la perturbation. Si la largeur du pas est grande, l'algorithme MPPT réagira rapidement aux changements soudains des conditions de fonctionnement. Cependant, les pertes en puissance vont être plus importantes lorsque les conditions de fonctionnement sont stables ou doucement changeantes.

Autrement dit, si la taille du pas est petite, les pertes de puissance sous des conditions de fonctionnement stables ou lentement changeantes seront inférieures mais le système ne pourra pas répondre rapidement aux changements brusques de température ou d'éclairement.

L'inconvénient de la méthode basée sur le control P&O est que dans le cas d'un changement rapide dans les conditions d'éclairement, tel qu'une voiture électrique qui rentre dans un tunnel, cette méthode peut déplacer le point de fonctionnement dans la mauvaise direction.

L'organigramme du contrôleur P&O est illustré dans la figure.2.12.

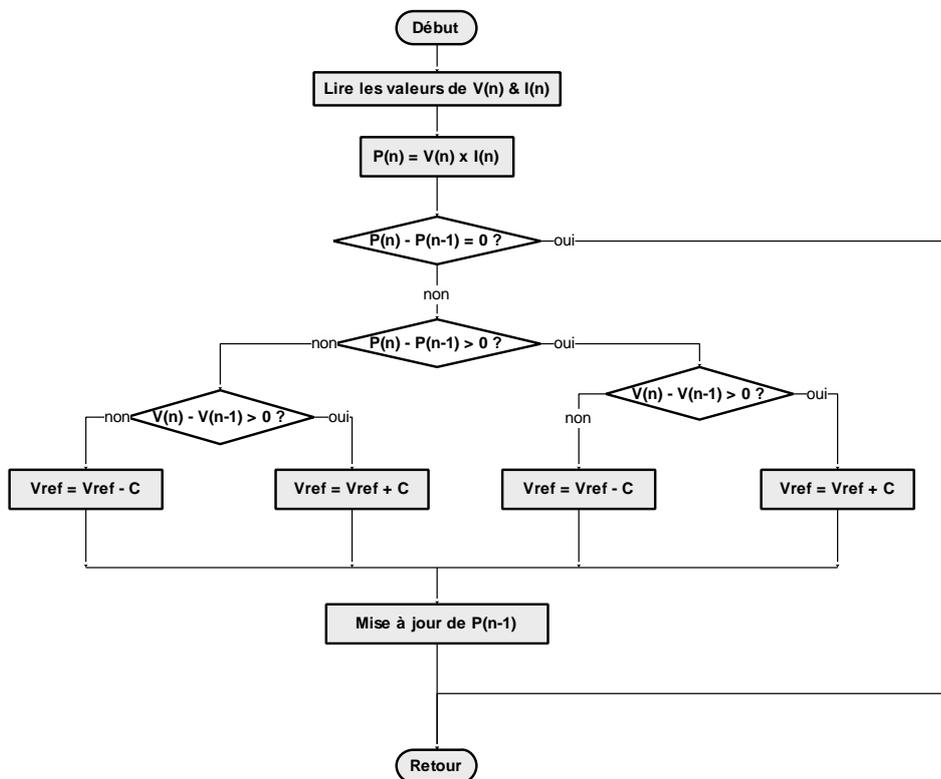


Figure.2.12 Organigramme du contrôleur P&O.

2.4.2.1.2 Méthode basée sur le contrôleur d'incrémental d'inductance (IncCond)

Pour répondre aux inconvénients de la méthode P&O, une autre méthode MPPT dite à incrémental de conductance (IncCond) [40-42] a été développée en 1995. Cette dernière est basée sur le fait que la pente de la courbe caractéristique de la puissance du panneau est nulle au MPP, positive à gauche et négative à droite.

Cela se traduit par les équations suivantes :

$$\frac{dP}{dV} = 0 \quad \text{au point de puissance maximale PPM} \tag{5}$$

$$\frac{dP}{dV} > 0 \quad \text{à gauche du point de puissance maximale PPM} \tag{6}$$

$$\frac{dP}{dV} < 0 \quad \text{à droite du point de puissance maximale PPM} \tag{7}$$

L'organigramme de la méthode est illustré par la figure ci-après.

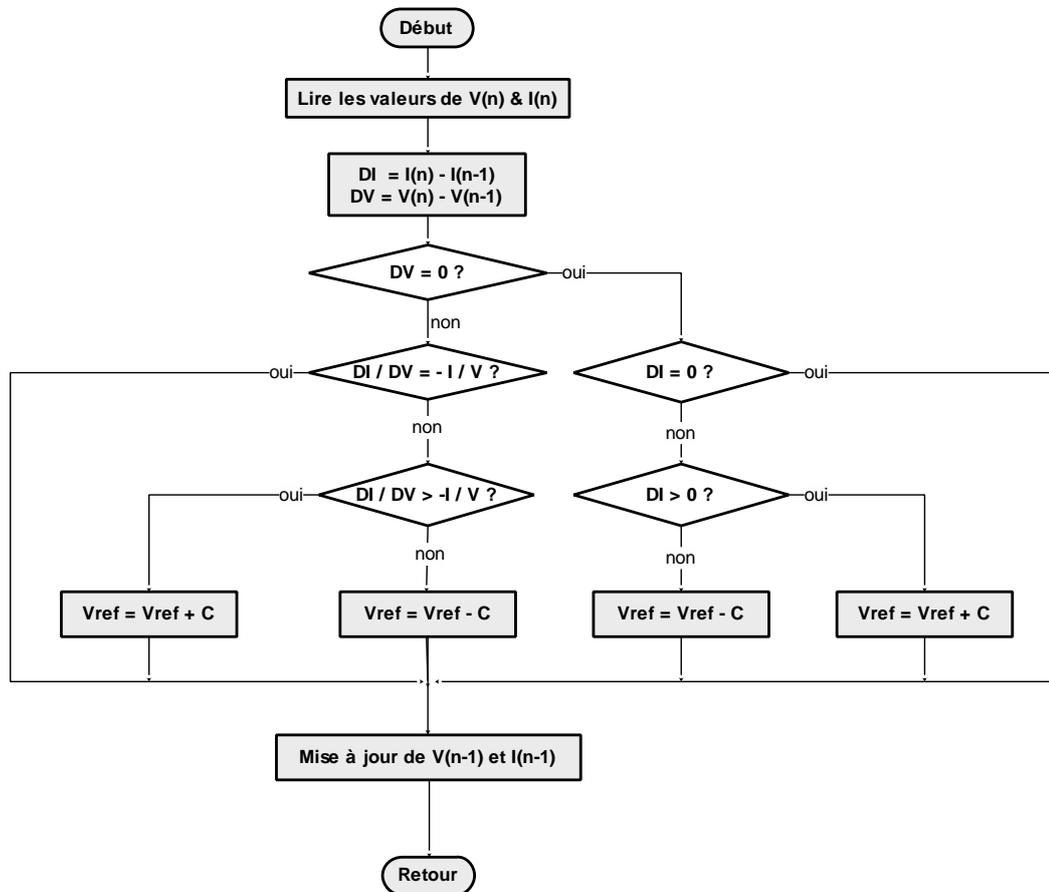


Figure.2.13 Organigramme du contrôleur Inc Cnd.

Combien même les méthodes conventionnelles citées pour la recherche du PPM efficacement dans des conditions normales soient simples et faciles à implémenter, elles présentent un certain nombre d'inconvénients dont :

- Les oscillations continues qui se produisent autour du PPM résultant en une perte considérable de la puissance ;
- Le problème des maxima locaux. En effet, ces méthodes ne sont pas adaptées en cas de variations brusques des conditions atmosphériques (éclairage et/ou température) et en cas d'ombrage partiel qui génère plusieurs maximums sur la caractéristique P-V. Ces dernières convergent vers des maxima locaux et ne sont pas adaptées aux variations atmosphériques.

Afin de pallier à ces multiples problèmes, de nouvelles méthodes basées sur l'approche *Soft Computing* ont été introduites.

2.5 LES MÉTHODES SOFT COMPUTING (MSC)

Durant ces dernières années, de nouvelles techniques d'optimisation ont vu le jour. Leur principe repose sur la recherche de solutions en tenant compte de l'incertitude, de l'imprécision de l'information réelle et en utilisant de plus en plus la notion d'apprentissage. Désormais, le but n'est plus de trouver des solutions exactes, mais plutôt des solutions suffisamment satisfaisantes, à coût approprié. C'est sur la base de ces nouvelles méthodes que le concept "*Soft Computing*" a été introduit par Zadeh [43,44].

Il est à signaler que le terme "*Soft Computing*" se réfère à un ensemble de techniques de calcul (computational techniques) utilisées dans plusieurs domaines, notamment, *l'ingénierie, le traitement d'images, la robotique...etc.*

Par ailleurs et afin de concevoir des contrôleurs MPPT performants, d'autres avantages liés aux MSC sont exploités, à savoir : *la flexibilité, la robustesse et la tolérance à l'imprécision.*

Comme l'illustre la figure.2.6, les méthodes *Soft Computing* regroupent plusieurs techniques basées sur des inspirations diverses, notamment la logique floue, les réseaux de neurones et le calcul évolutif. En général, ces méthodes reposent particulièrement sur les processus biologiques et sociologiques et considèrent les êtres vivants évoluant dans la nature comme modèles d'inspiration.

Ainsi, les méthodes basées sur l'approche SC ont suscité un immense intérêt ces dernières années de la part de la communauté des chercheurs [29,45], en raison de leur capacité à remédier aux problèmes des oscillations continues, des maxima locaux autour du PPM liés aux conditions atmosphériques (éclairage et/ou température) et d'ombrage partiel engendrant des pertes considérables de puissance.

Au regard de l'évolution algorithmique ayant caractérisé les MSC, une panoplie de nouveaux contrôleurs MPPT ont été proposés pour la poursuite du PPM. Nous les avons subdivisés en deux grandes catégories : les *Méthodes basées sur l'Intelligence Artificielle* (MIA) et les *Méthodes Bio Inspirées* (MBI).

2.5.1 Les méthodes basées sur l'intelligence artificielle (MIA)

De plus en plus, ces dernières années, comme alternative aux méthodes conventionnelles, on assiste à l'utilisation des méthodes basées sur l'intelligence artificielle (MIA) [46]. Ces dernières, peuvent être regroupées en trois grandes classes : celles qui se basent sur les Réseaux de Neurones Artificiels (ANN), celles qui s'appuient sur la Logique Floue (FLC) et enfin, celles qui reposent sur la combinaison de plusieurs méthodes intelligentes à savoir : les méthodes hybrides.

Ces méthodes ont été introduites pour résoudre les problèmes d'optimisation de la puissance dans les systèmes PV i.e recherche du PPM. Plusieurs travaux de recherches ont été présentés.

Parmi lesquels, nous pouvons citer les contrôleurs MPPT a base de *Réseaux de Neurones Artificiels* (ANN_MPPT) [47-51], de *Logique Floue* (FLC_MPPT) [52-57] et des méthodes hybrides telles que *les Algorithmes Génétiques combinée avec la Logique Floue* (GA_FLC_MPPT)[58-60], *les Algorithmes Génétiques combinée avec les Réseaux de Neurones Artificiels* (GA_ANN_MPPT) [61,62] et le *système d'Inférence Floue Neuro Adaptative* (ANFIS_MPPT) [63-67].

Dans ce qui suit, un aperçu de ces différentes méthodes sera donné.

2.5.1.1 Contrôleur MPPT à base de réseaux de neurones artificiels (ANN_MPPT)

Pour traiter des problèmes de grande complexité, l'utilisation des algorithmes à base de réseaux de neurones s'est avérée très efficace. L'utilisation de ces algorithmes dans divers domaines d'application ne cesse d'augmenter vu qu'ils fonctionnent à partir d'une boîte noire qui n'exige pas d'informations détaillées sur le fonctionnement du système [66]. Grâce à leurs performances, ces algorithmes ont été appliqués pour le contrôle et la recherche du PPM. Ces derniers se sont avérés efficaces pour localiser le point de puissance maximale réduisant le nombre d'itérations et des oscillations autour du PPM.

Généralement, le type de réseaux de neurones utilisé pour l'élaboration de contrôleur MPPT se compose de trois couches comme l'illustre la figure.2.14 Une *couche d'entrée*, une *couche cachée*, et une *couche en sortie*.

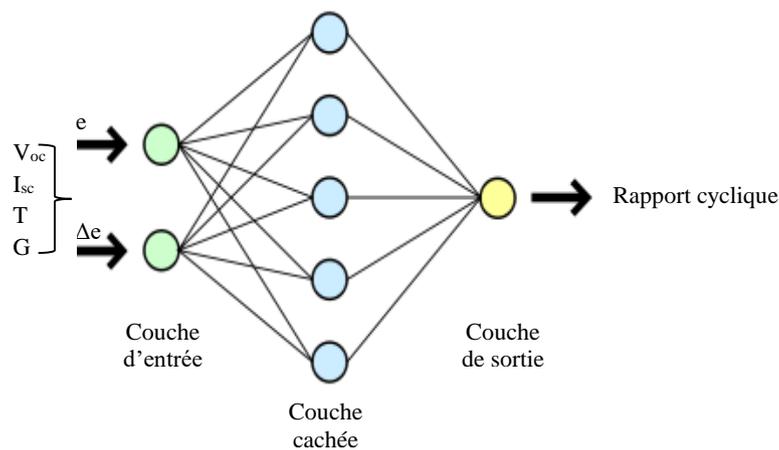


Figure.2.14 Architecture d'un réseau de neurones.

Dans la majorité des contrôleurs développés et cités dans le cadre de l'optimisation de puissance à base de ces méthodes, les variables d'entrée du réseau de neurones sont les paramètres du module PV : la tension V_{oc} , le courant I_{sc} , la température et l'éclairement. La variable de sortie étant le rapport cyclique utilisé pour forcer le convertisseur de puissance à fonctionner autour du PPM.

La figure ci-dessous illustre l'implémentation d'un contrôleur MPPT à base de réseaux de neurones.

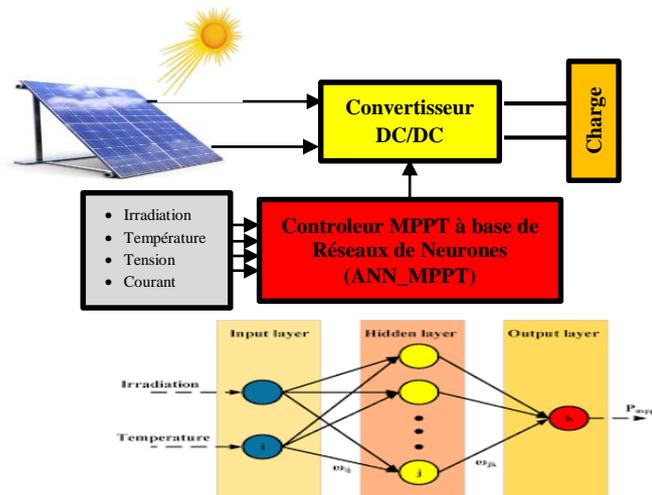


Figure.2.15 Mise en œuvre du contrôleur MPPT basée sur les réseaux de neurones.

Le tableau ci-dessous résume quelques travaux de recherches issus de la littérature relatifs à l'application des réseaux de neurones artificiels pour le développement de contrôleurs MPPT.

Tableau.2.1 Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application des réseaux de neurones artificiels pour le développement de contrôleurs MPPT.

Auteurs	Référence	
Y.H.Liu, C.L.Liu, et al	[47]	Le réseau de neurone est utilisé pour calculer les paramètres d'un émulateur de localisation du PPM qui sera intégré dans le système de control MPPT. Le système ainsi proposé permet : - une recherche rapide du PPM, - méthode très simple nécessitant quelques opérations multiplication et addition - et une efficacité de suivi statique / dynamique élevée.
A.K. Rai,N.D. Kaushika, et al	[48]	Le contrôleur ANN permet d'estimer les tensions et les courants correspondant à une puissance maximale fournie par un réseau photovoltaïque pour une température et un rayonnement solaire variable. Le réseau Back propagation est utilisé pour entrainer le système.
M.S.Ait Cheikh, M.Haddadi, et al	[49]	La méthode diffère des autres, montre que nous n'avons pas besoin de mesures du courant et de la tension pour évaluer la puissance de sortie et ainsi, pour détecter la position du PPM.
R.Essefi, M.Souissi, et al	[50]	Le contrôleur ANN développé permet de suivre le PPM sous variations rapides de conditions climatiques.
D.Vasarevicius, R.Martavicius, et al	[51]	Le réseau de neurone ANN développé est utilisé pour augmenter les performances du système.

2.5.1.2 Contrôleur MPPT à base de logique floue (FLC_MPPT)

Les méthodes à base de logique floue ont l'avantage de pouvoir traiter des systèmes non-linéaires sans avoir besoin d'un modèle précis du système pris en considération.

Comme pour les méthodes à base de réseaux de neurones, le contrôle MPPT à base de logique floue s'est avérée aussi efficace [52-57]. De manière générale, le principe du contrôleur utilisant la logique floue se base sur deux variables d'entrées qui sont l'erreur (E), sa variation

(ΔE) et une variable en sortie (ΔD), correspondant à la variation du rapport cyclique qui pilote le signal de commande du convertisseur statique pour la recherche du PPM.

Le contrôleur flou se compose de trois blocs de traitement consécutifs comme l'illustre la figure.2.16 le premier bloc est celui de la *fuzzification* qui permet le passage du domaine réel au domaine flou ; le deuxième est consacré aux *règles d'inférences* alors que le dernier est réservé à l'opération de *defuzzification* permettant le retour au domaine réel.



Figure.2.16 Structure de base du contrôleur MPPT à base logique floue.

La figure ci-dessous illustre l'implémentation du contrôleur MPPT à base de logique floue.

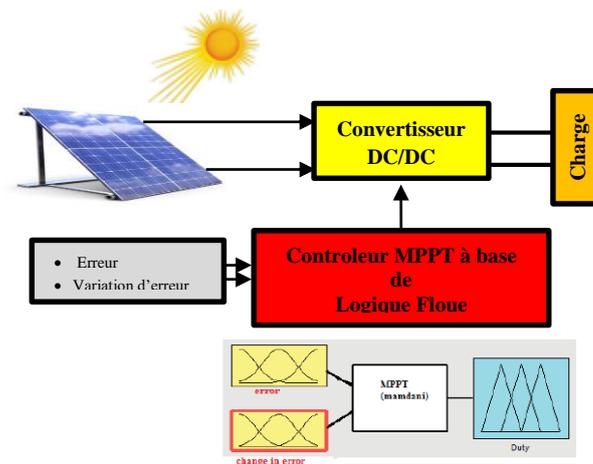


Figure.2.17 Mise en œuvre du contrôleur MPPT basée sur la logique floue.

Le tableau ci-dessous résume quelques travaux de recherches relatifs au développement de contrôleurs MPPT à base logique floue.

Tableau.2.2 Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de la logique floue pour le développement de contrôleurs MPPT.

Auteurs	Référence	
M.S.Ait Cheikh, C.Larbes, et al	[52]	La méthode emploie un contrôleur de logique floue appliqué à un dispositif de convertisseur DC-DC. Il permet de trouver le PPM en un temps plus court comparé au contrôleur P & O bien connu.
F.Bouchafa, D.Beriber, et al	[53]	Une méthode de contrôle intelligente pour le suivi du PPM d'un système photovoltaïque sous variations rapides de conditions climatiques.
F.Chekired, C.Larbes, et al	[54]	Les auteurs proposent une méthode de suivi du point PPM basée sur le contrôleur de logique floue (FLC), appliquée à un système photovoltaïque autonome sous variations rapides de conditions climatiques. Le contrôleur est implémenté sur circuit FPGA.
M.A.Othmana, M.M.ElArinia, et al	[55]	Les auteurs proposent un contrôleur MPPT à base de logique floue. Une étude comparative avec un contrôleur P&O est établie. Les simulations sont faites avec des données réelles pour le rayonnement solaire
H.Bounechba, A.Bouzid, et al	[57]	Les auteurs proposent un contrôleur MPPT pour le convertisseur boost basé sur la logique floue (FLC).Il est développé et comparé à l'algorithme de (P&O).

2.5.1.3 Contrôleur MPPT à base de méthodes hybrides

Les systèmes intelligents hybrides combinent plusieurs méthodes intelligentes. L'application de celles-ci dans le domaine de l'optimisation de puissance dans les systèmes PV a montré de bonnes performances [58-67].

Parmi ces dernières, nous pouvons citer :

2.5.1.3.1 Méthodes basées sur les systèmes neuro-flous

Ces méthodes d'optimisation basée sur le système *neuro-flou* combinent les concepts de deux paradigmes : les réseaux de neurones artificiels et la logique floue. Ces derniers mettent à profit les deux méthodes : à savoir la capacité d'apprentissage de la première, la lisibilité et la souplesse de la seconde.

Un système neuro-flou est donc un réseau neuronal multicouche avec des paramètres flous. Ainsi, le contrôleur neuro-flou est constitué de deux entrées (E) et (ΔE), et d'une seule sortie (D), qui représentent respectivement l'erreur, la variation d'erreur et le rapport cyclique qui pilote le signal de commande du convertisseur statique pour la recherche du point de puissance maximale.

La figure.2.18 illustre l'architecture du contrôleur ANFIS.

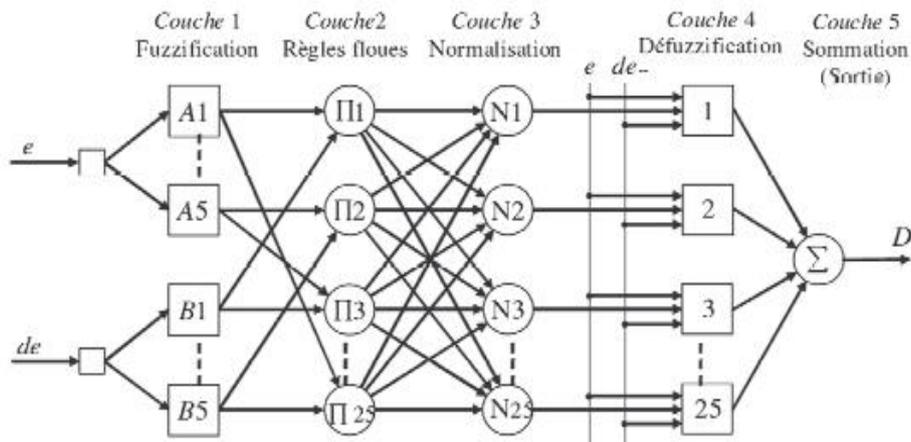


Figure.2.18 Architecture du réseau ANFIS.

La figure ci-dessous illustre l'implémentation du contrôleur MPPT à base de la méthode neuro floue.

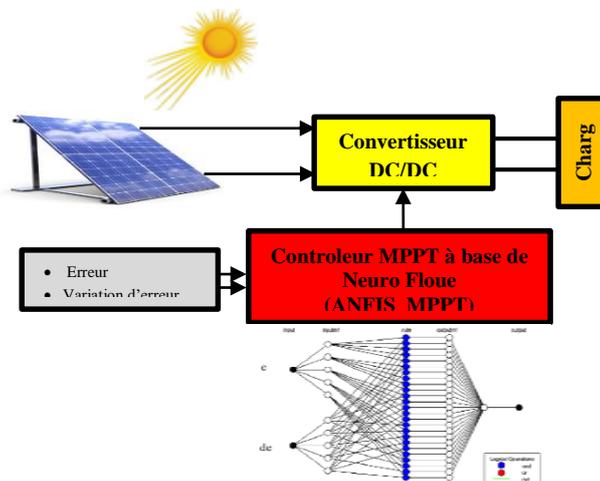


Figure.2.19 Mise en œuvre du contrôleur MPPT à base de Neuro Floue ANFIS sur la logique floue.

2.5.1.3.2 Méthodes basées sur la logique floue et les algorithmes génétiques

La méthode d'optimisation de la logique floue par des algorithmes génétiques qui, comme son nom l'indique, combine les algorithmes génétiques et la logique floue. La méthode est apparue suite aux difficultés rencontrées dans la conception des commandes floues, qui nécessitent des connaissances spécialisées et un bon choix des paramètres flous et de leurs fonctions d'appartenance, ce qui a amené les chercheurs à privilégier l'utilisation des algorithmes génétiques pour leur aptitude adaptative, leur efficacité dans la génération des règles floues et leur robustesse.

Ces caractéristiques font de ces derniers des outils potentiels pour établir les fonctions d'appartenance des commandes floues et trouver les paramètres optimaux comme indiqué dans [58].

Ainsi, le contrôleur MPPT flou optimisé par les algorithmes génétiques (FL_GA_MPPT) est constitué de deux entrées (E), (ΔE) et d'une sortie (D). Chacune de ces variables est définie par cinq fonctions d'appartenance. La population se compose d'un ensemble d'individus, chacun ayant trois chromosomes : E, ΔE et D tel que rapporté dans [58].

La figure ci-dessous illustre l'implémentation du contrôleur MPPT à base de la méthode flou optimisé par les algorithmes génétiques.

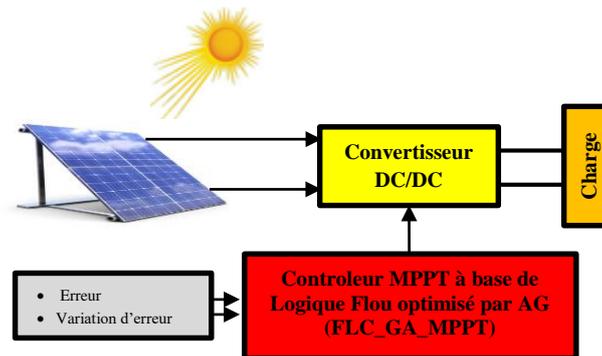


Figure.2.20 Mise en œuvre du contrôleur MPPT à base de la méthode floue optimisée par les Algorithmes Génétiques.

Tableau.2.3 Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de méthodes hybride pour le développement de contrôleurs MPPT.

Auteurs	Référence	
C.Larbes, S.M.Ait Cheikh et al	[58]	Les auteurs proposent un contrôleur MPPT avec une méthode hybride. Les AGs sont utilisés pour trouver et optimiser les fonctions d'appartenance du contrôleur FLC.
F. Chekired, A. Mellit, et al	[59]	Les auteurs proposent plusieurs contrôleurs intelligents pour la poursuite du PPM. Une étude comparative en terme de performances des différents contrôleurs est effectuée.
A. Messai, A. Mellit, et al	[60]	Le contrôleur MPPT élaboré utilise un contrôleur flou optimisé par les AGs. Le contrôleur FLC_GA_MPPT ainsi conçu, ainsi que les composants de l'unité de contrôle du système PV ont été implémentés efficacement sur un FPGA.
A.A.Kulaksiz, R.Akkaya, et al	[61]	Les auteurs proposent un contrôleur MPPT à base d'un réseau de neurones entraîné à l'aide d'une base de données obtenue à partir d'un résultat expérimental. Les AGs sont utilisés pour définir la taille du réseau de neurone.
R.Ramaprabha, V.Gothandaraman, et al	[62]	Le contrôleur MPPT proposé est basé sur un réseau neuronal artificiel (ANN) optimisé par les algorithmes génétiques (GA). Les valeurs de puissance optimisées et les valeurs de tension correspondantes pour différents niveaux de températures et d'insolation ont été utilisées pour élaborer le réseau de neurones. Ensuite, les AGs sont utilisés pour fournir la Tension de référence correspondant à la puissance maximale.
Syafaruddin, E.Karatep, et al	[63]	Le contrôleur MPPT est proposé pour le contrôle d'un SPV partiellement ombragé utilisant les réseaux neurones (ANN) et de la logique floue. L'ANN est entraîné sous plusieurs conditions d'ombrage partiel pour déterminer la tension MPP globale. La logique floue utilise la tension V_{MPP} globale comme tension de référence pour générer le signal de commande requis pour la conversion de puissance.

A..M.Z Alabedin, EF.El- Saadany, et al	[64]	Les auteurs proposent la conception de deux contrôleurs pour le suivi du PPM des systèmes photovoltaïques en utilisant la logique floue et les réseaux neuronaux artificiels.
F.Chekired, C.Larbes,et al	[65]	Les auteurs proposent un contrôleur MPPT neuro flou ANFIS_MPPT. Une simulation en temps réel réalisée suivi d'une implémentation sur circuit FPGA.
F.Chekired, C.Larbes, A.Mellit	[66]	Les auteurs présentent une étude comparative entre deux contrôleurs MPPT : le premier à base de logique floue, le second à base de neuro-floue. Les deux contrôleurs sont implémentés sur circuit FPGA
F.Belhachet	[67]	L'auteur présente une étude à travers le développement et l'implémentation d'un contrôleur MPPT neuro-flou

Les méthodes MIA offrent d'importants avantages en raison de leur capacité à résoudre les problèmes de non linéarité, ce qui les rend très attrayantes pour le contrôle et l'optimisation de puissance des systèmes PV.

De ce fait, les contrôleurs MPPT basés sur les MIA sont développés et utilisés afin d'améliorer l'efficacité de la conversion d'énergie indépendamment des variations atmosphériques ce qui n'est pas possible avec les méthodes conventionnelles.

Néanmoins, malgré les mérites liés aux MIA, il n'en demeure pas moins que ces derniers présentent des inconvénients.

Les contrôleurs MPPT à base de *Réseaux de Neurones Artificiels* (ANN_MPPT) nécessitent des mois d'apprentissage pour assurer le suivi du PPM surtout lors des variations atmosphériques. De plus, le nombre de nœuds et des couches du réseau de neurones affectent la recherche du PPM. Un nombre de nœuds important augmenterait la précision générant un temps de calcul considérable pour assurer le suivi de la PPM affectant ainsi la vitesse de convergence.

En outre, les contrôleurs à base de *Logique Floue (Fuzzy Logic Controller)* (FLC_MPPT) exigent des connaissances afin de créer des règles de base d'inférence pour le suivi du PPM, donc une grande taille de la mémoire. De plus, les variations rapides des conditions atmosphériques affectent la recherche du PPM. Ceci est dû à la table des règles d'inférence et les fonctions d'appartenance qui sont statiques, alors que la variation des conditions atmosphériques change de façon dynamique.

Afin de surmonter ces contraintes, les contrôleurs MPPT à base de méthodes hybrides ont été proposés et ont démontré leur efficacité même lors de variations rapides des conditions

atmosphériques. Toutefois, le problème lié au temps de calcul important demeure toujours existant ce qui influe sur la convergence.

Pour atténuer cet inconvénient majeur, de toutes les méthodes citées ci-dessus, récemment, de nouvelles méthodes intelligentes ont été développées afin de surmonter les limitations des méthodes intelligentes artificielles traditionnelles.

La particularité de ces méthodes intelligentes qui mérite d'être signalée réside dans le fait que leur mode de fonctionnement soit plus réaliste et mieux compréhensible.

Généralement, ces méthodes sont beaucoup plus efficaces que les méthodes traditionnelles d'intelligence artificielle. Afin de les distinguer de celles-ci, on les définit comme étant des méthodes **Bio Inspirée** (MBI). Ces dernières nouvellement introduites dans le domaine des énergies renouvelables, plus particulièrement les systèmes PV, suscitent un vif engouement de la part de la communauté scientifique.

2.5.2 Les méthodes bio inspirées (MBI)

2.5.2.1 Concepts de base du bio inspiré

La recherche dans le domaine des méthodes **Bio Inspirées** (MBI) [68] s'explique par l'émerveillement ancestral de l'être humain lorsqu'il observe la nature, particulièrement les êtres vivants qui s'attèlent à user de techniques et de prouesses dans leur vie quotidienne pour répondre à des besoins basiques : se nourrir, construire des abris, procréer, se déplacer, bâtir des sociétés...etc.

Utiliser ces observations dans le monde de la recherche et du développement, comme source d'inspiration, est devenu au fil des ans, une piste de recherche riche et prometteuse, voire passionnelle pour certains chercheurs qui œuvrent dans l'élaboration de solutions novatrices répondant à des problèmes en ingénierie, parfois complexes.

En effet, la nature avec ses surprenants phénomènes, notamment la vie des êtres vivants qui la composent, nous prodigue des solutions, parfois surprenantes, à des problèmes particuliers d'ingénierie. Les caractéristiques de ces solutions bio inspirées tirées des différents travaux de recherche issus de la littérature [69,70], sont résumées dans la figure ci-après :

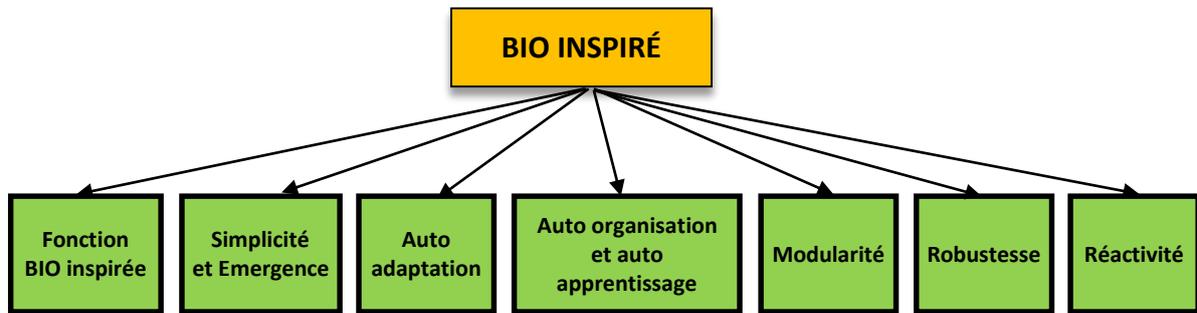


Figure.2.21 Caractéristiques du bio inspiré.

- **Fonction bio inspirée** : le mécanisme de travail est très proche du fonctionnement naturel biologique ou éthologique des organismes vivants. Les algorithmes *bio-inspirés* imitent donc, au plus près, la nature biologique, afin de répondre et résoudre des problèmes du monde réel ;
- **Simplicité et Emergence** : la stratégie et le calcul sont très basiques, mais les effets qui en résultent sont très intéressants. Les éléments simples qui interagissent vont accomplir des tâches surprenantes. C’est le principe de l’émergence ;
- **Auto-organisation et auto-apprentissage** : ces algorithmes peuvent augmenter la capacité d’adaptation par l’auto-apprentissage et l’auto-organisation et permettent de réaliser des évolutions avec succès. Cela signifie que la structuration interne du système s’organise de manière automatique sans être affectée, orientée ou dirigée par une source extérieure à ce système ;
- **Robustesse** : Ces algorithmes ont une forte robustesse et sont très résistants aux changements des environnements (même lorsqu’un élément n’est plus en état de remplir son rôle, le groupe continu à exécuter sa tâche). Cela induit une bonne flexibilité de ces algorithmes ;
- **Modularité & flexibilité** : Le système se compose d’un ensemble d’éléments simples qui concourent collectivement dans le but d’atteindre un objectif global. Ainsi, le système peut être qualifié comme étant évolutif ;
- **Réactivité** : En évoluant collectivement, les éléments du système coopèrent et communiquent entre eux à travers des interactions locales. Ainsi, ils peuvent réagir instantanément dès que des changements d’environnement s’opèrent ;
- **Auto-adaptation** : Cela signifie que le système peut modifier ses paramètres en assurant un fonctionnement satisfaisant, en dépit des variations de son environnement ;

Les chercheurs ont alors exploité ces caractéristiques pour fournir des algorithmes pour lesquels les méthodes conventionnelles et intelligentes n'ont pas donné de solutions satisfaisantes.

La figure ci-après illustre les étapes de l'élaboration de ces algorithmes :

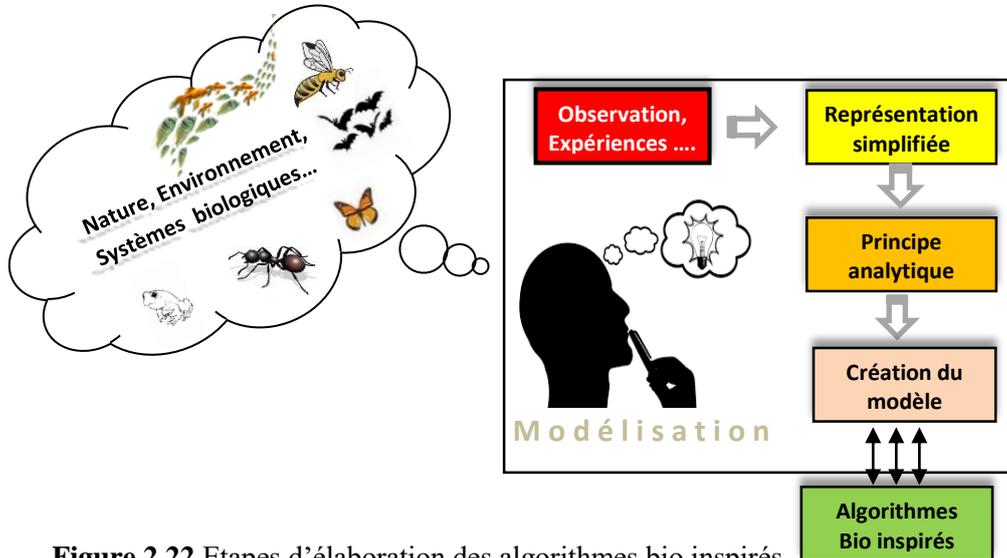


Figure.2.22 Etapes d'élaboration des algorithmes bio inspirés.

Ces algorithmes ont un fonctionnement simple basé sur la sélection naturelle. Afin de résoudre des problèmes d'optimisation, ils entament la résolution par un échantillon d'individus constituant une population initiale, choisi arbitrairement dans un intervalle de recherche prédéfini.

Par la suite, cette population est améliorée au fur et à mesure en générant de nouvelles solutions en actualisant la position de chacun des individus en fonction de la meilleure solution identifiée précédemment. Cette étape est répétée jusqu'à ce que la meilleure solution soit atteinte pour un nombre d'itération définis au départ (convergence).

Au cours de ces dernières années, les algorithmes issus des méthodes *Bio Inspirée* (MBI) ont été appliqués dans de nombreux domaines en raison de leur aptitude à optimiser la capacité de recherche sans fournir trop de calculs mathématiques.

De nos jours, ces méthodes sont encore au stade de développement, cependant, il n'existe pas un standard pour une classification uniforme de ces dernières. Ainsi, de par les multiples définitions recueillies dans la littérature, les deux principaux groupes issus de ces méthodes sont : la biologie et les théories de l'évolution qui ont inspirés les *Algorithmes Evolutionnaires* (Evolutionary Algorithms) et l'éthologie avec les algorithmes à base

d'*Intelligence en Essaim (Collective)* plus communément connue sous le nom de « *Swarm Intelligence* ».

La figure.2.23 illustre les deux principaux groupes du domaine Bio-Inspiré.

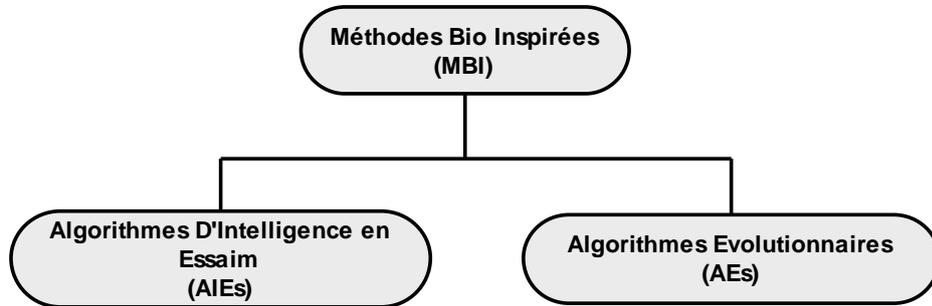


Figure.2.23 Groupes du domaine Bio Inspiré.

Les méthodes bio inspirées les plus récentes, utilisées dans le domaine des énergies renouvelables pour la recherche du PPM, sont présentées ci-après :

2.5.2.2 Méthodes basées sur les algorithmes évolutionnaires (MAE)

Ces méthodes se basent sur des techniques inspirées de l'évolution biologique des espèces, elles sont apparues à la fin des années 50. Il s'agit de générer des populations de solutions potentielles à des problèmes donnés, de sélectionner les meilleures solutions et de les répliquer en faisant quelques modifications pour chercher une meilleure solution aux problèmes.

La figure ci-dessous illustre La structure d'un algorithme évolutionnaire.

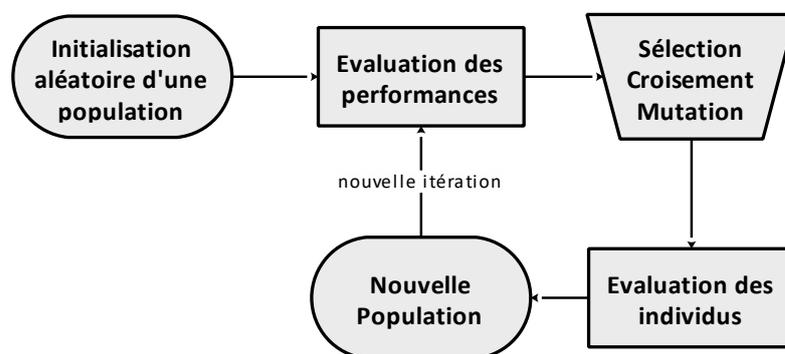


Figure.2.24 Structure d'un algorithme évolutionnaire.

Parmi les variantes des algorithmes évolutionnaires, on peut citer les Algorithmes Génétiques (AG) (Genetic Algorithms), et les algorithmes à Evolution Différentielle (ED) (Differential Evolution).

Avec ces méthodes, plusieurs types de contrôleurs ont été développés. A titre d'exemple, on pourra énumérer :

2.5.2.2.1 Contrôleur MPPT à base des algorithmes génétiques (GA_MPPT)

Les algorithmes génétiques (AG) ont été initialement développés par John Holland (1975) [71]. Ils comptent parmi les approches issues des méthodes évolutionnaires utilisées pour résoudre les différents problèmes d'optimisations dans le domaine des énergies renouvelables. Plus particulièrement pour la poursuite du point de puissance maximale dans les systèmes PV afin de profiter au maximum de l'énergie solaire et suivre rapidement ce point [72-74].

L'organigramme illustrant les différentes étapes pour la mise en œuvre du contrôleur MPPT basé sur les AG est schématisé dans la figure.2.25 ci-dessous.

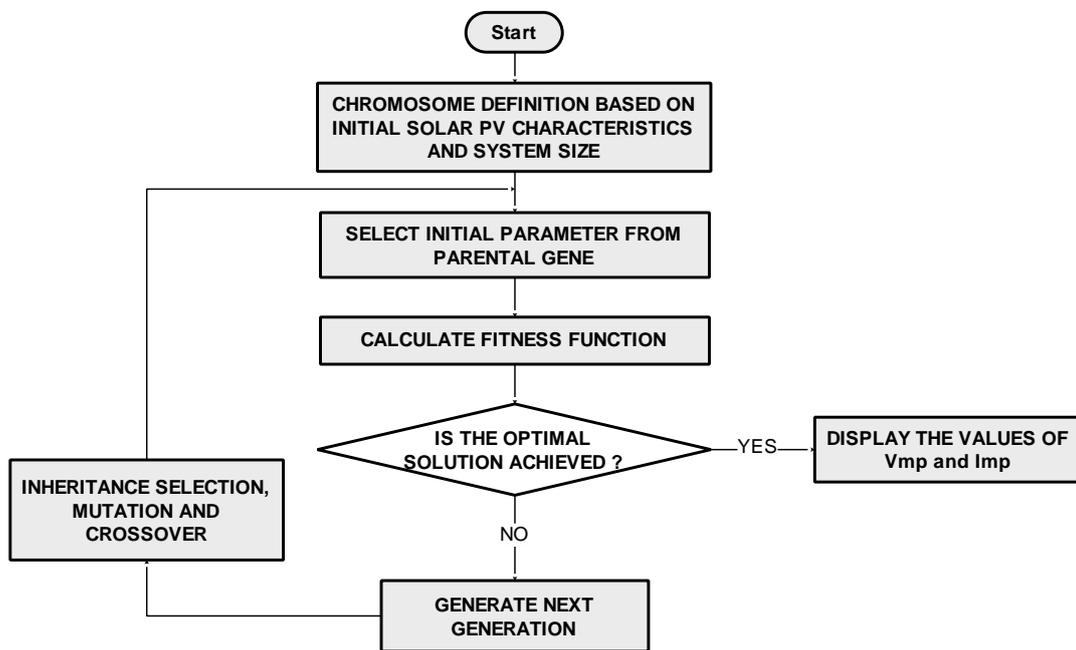


Figure.2.25 Organigramme du contrôleur GA [72].

Dans ce cas, la population parent initiale est présentée comme suit :

$$X^i = [Parent^1, Parent^2, \dots, Parent^n]$$

Où n représente la taille de la population et chaque parent représente la valeur initiale de la tension pour laquelle l'algorithme démarre le processus d'évaluation. La fonction objective $f(xi)$ est la puissance générée à la sortie du système photovoltaïque.

Les valeurs de la fitness pour chaque position sont évaluées par la fonction objective et sont utilisées pour évoluer et améliorer la population.

Le tableau.2.4 résume quelques travaux de recherches relatifs au développement de contrôleurs MPPT à base d'Algorithme Génétique (GA_MPPT), nous pouvons citer :

Tableau.2.4 Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application des Algorithmes Génétiques pour le développement de contrôleurs MPPT.

Auteurs	Référence	
S.Hadji, J.P.Gaubert, et al	[72]	Les auteurs proposent un contrôleur MPPT à base d'AGs, l'approche consiste à mesurer la tension de circuit ouvert V_{oc} et le courant de court-circuit I_{sc} , et récupérer directement la tension optimale V_{op} afin que le rapport cyclique du convertisseur puisse être ajusté. Une étude comparative avec les algorithmes conventionnels P&O et IncCond est effectuée.
R.Ramaprabha,B.L. Mathur	[73]	Les auteurs proposent un contrôleur à base d'AGs qui prend en charge les problèmes d'ombrage partiel pour la recherche du GMPPT.
H.R.Mohajeri,M.P. Moghaddam, et al	[74]	

Cependant, ces contrôleurs présentent un inconvénient lié à l'utilisation des relations de probabilité ce qui induit des résultats parfois imprévisibles pour lesquels les performances ne peuvent être contrôlées.

Ainsi, la méthode GA combinée à d'autres méthodes classiques/intelligentes est privilégiée pour augmenter ces performances [58,60-63].

2.5.2.2.2 Contrôleur MPPT à base de l'algorithme à évolution différentielle (DE_MPPT)

L'algorithme à Evolution Différentielle (ED) est une méthode proposée par Storn & Price en 1995 [75]. La méthode est basée sur le principe de l'évolution naturelle. Elle considère des informations issues de la population courante pour déterminer l'évolution future probable des individus qui la composent.

Cette méthode a été appliquée dans le domaine des énergies renouvelables et particulièrement pour la poursuite du PPM dans le cas d'ombrage partiel. Dans ce cas, la fonction objective correspond à la puissance du PV et le rapport cyclique représente la population.

L'organigramme pour la mise en œuvre du contrôleur MPPT basé sur ED est illustré dans la figure ci-après.

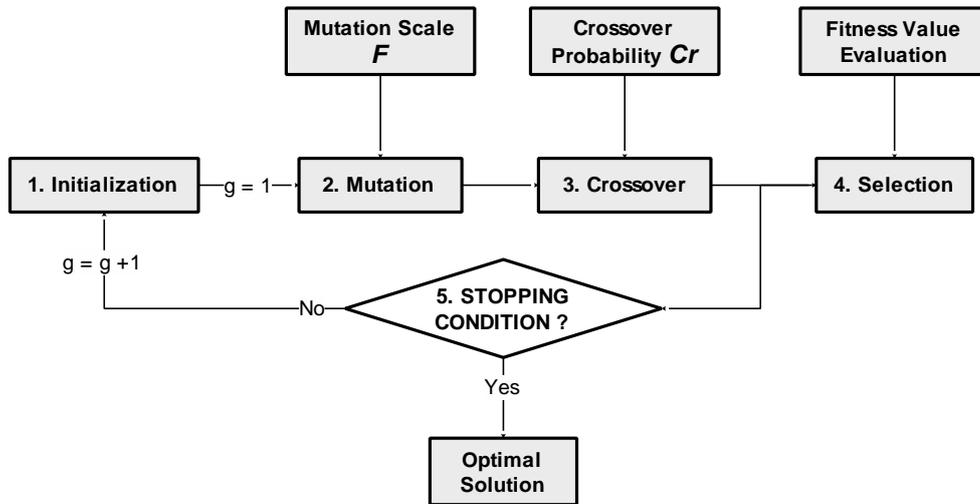


Figure.2.26 Organigramme du contrôleur DE [76].

Un résumé de quelques travaux de recherches relatifs au développement de contrôleurs MPPT à base d'algorithme d'évolution différentielle (DE_MPPT) est illustré dans le tableau ci-dessous.

Tableau.2.5 Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application des Algorithmes à évolution différentielle pour le développement de contrôleurs MPPT.

Auteurs	Référence	
K.S.Tey, S.Mekhilef, et al	[76]	Les auteurs proposent une technique de DE utilisée pour un système photovoltaïque sous ombrage partiel. Le temps de traitement a été réduit. Les performances de l'algorithme sont évaluées sous différentes variations de températures et d'éclairement.
S. Ramasamy, S. Jeevananthan	[77]	
S.Taheri, Z.Salam, et al	[78]	
M.F.Tajuddin, M.A. Shahrin, et al	[79]	Une technique DE modifiée a été utilisée pour traiter le problème d'ombrage partiel. Elle est comparée à une méthode conventionnelle : <i>Hill Climbing</i> .

2.5.2.3 Méthodes basées sur l'intelligence en Essaim

L'intelligence en essaim (*collective*) résulte des aptitudes cognitives des membres composant une communauté d'individus qui interagissent entre eux. Ces agents aux comportements individuels basiques peuvent exécuter des tâches collectives complexes grâce à un mécanisme fondamental appelé « synergie ». Sous certaines conditions, la collaboration entre individus crée une synergie qui fera émerger des possibilités de représentation, de création ou d'apprentissage bien supérieurs à celles des individus pris isolément.

Les formes d'intelligence en essaims sont très variées selon le type de communauté formée par les membres qui la composent. Les systèmes collectifs ainsi constitués ont des degrés de sophistication variés.

La société humaine en particulier, n'obéit pas à des règles aussi mécaniques que d'autres sociétés qu'on trouve dans la nature. Pour des systèmes simples, les principales caractéristiques sont :

- L'information locale : chaque membre de la communauté ne possède qu'une connaissance partielle de l'environnement dans lequel il évolue et il n'a pas conscience de la totalité des éléments qui influencent le groupe ;
- L'ensemble des règles : chaque membre de la communauté suit un ensemble fini et restreint de règles simples par rapport au comportement global de la communauté vue dans son ensemble ;
- Les interactions multiples : chaque membre est en relation avec un ou plusieurs membres de la communauté ;
- La collectivité : chaque membre trouve un intérêt à collaborer avec les autres (parfois instinctivement) et sa performance dans le groupe est bien supérieure que s'il œuvrait de manière individuelle.

L'intelligence en essaim est observée de manière accrue chez les sociétés d'insectes (fourmis, abeilles, etc...) et les animaux en mouvement (oiseaux migrateurs, chauves-souris, bancs de poissons, etc...).

On observe alors des comportements organisés, adaptatifs et qualifiés d'intelligents. C'est de cette observation que plusieurs algorithmes basés sur le principe d'intelligence en essaim ont été élaborés et introduits dans le monde de l'engineering.

Ces derniers présentent un avenir prometteur pour l'élaboration de nouveaux contrôleurs MPPT, qui demeure relativement vierge au vu du nombre de travaux existants.

En effet, de nos jours, les contrôleurs MPPT développés sont orientés pour la recherche des maxima globaux réduisant ainsi les pertes d'énergies en élargissant l'espace de recherche du PPM permettant, ainsi, à ce dernier de fonctionner et d'atteindre le PPM en cas d'ombrage partiel.

Parmi les contrôleurs répertoriés dans la littérature, et illustré par la Figure.2.6, nous pouvons citer :

2.5.2.3.1 Contrôleur MPPT à base de l'algorithme d'optimisation par Essaim de particules (PSO_MPPT)

L'algorithme d'optimisation par essaim de particules (Particle Swarm Optimization (PSO) est une méthode d'optimisation stochastique, pour les fonctions non-linéaires proposée par Eberhart et Kennedy en 1995.

Il s'inspire à l'origine du monde du vivant, plus précisément la reproduction du comportement social des animaux évoluant en essaim, tels que les vols groupés d'oiseaux et les bancs de poissons illustré par la figure ci-dessous.



Figure.2.27 Essaim d'oiseaux et banc de poissons.

L'optimisation par Essaims Particulaires (OEP) a été proposée dans divers travaux de recherches liés à l'optimisation de puissance des systèmes PV, plus particulièrement dans le cas d'ombrage. Dans ce cas, la fonction objective correspond à la puissance du PV et le rapport cyclique représente la population. La figure.2.28 illustre l'organigramme de l'OEP (PSO).

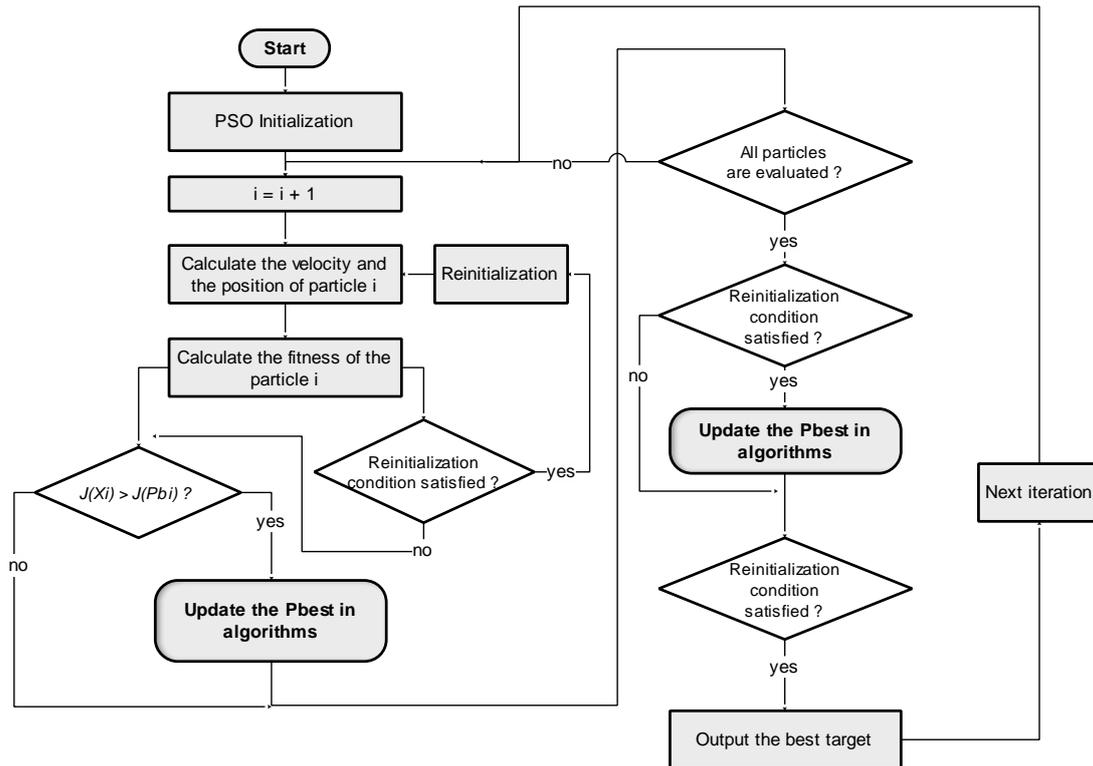


Figure.2.28 Organigramme du contrôleur PSO [80].

Le tableau.2.6 résume quelques travaux de recherches relatifs au développement de contrôleurs MPPT à base d'algorithme Optimisation par Essaim de Particules (OEP_MPPT).

Tableau.2.6 Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de l'algorithme d'optimisation par essaim de particules pour le développement de contrôleurs MPPT.

Auteurs	Référence	
Kashif Ishaque, Zainal Salam	[80]	Les auteurs proposent un contrôleur PSO améliorée pour surmonter les problèmes d'ombrage. La méthode possède de bonnes performances comparée à la méthode classique PSO. De plus, les oscillations autour du PPM sont quasi nulles.
Kashif Ishaque, Zainal Salam, et all	[81]	Les auteurs proposent un contrôleur DPSO à deux modes (mode global et local). Le contrôleur change son mode de fonctionnement en fonction des changements conditions d'irradiation.
S.R.Chowdhury, H.Saha	[82]	Les auteurs proposent un contrôleur où la convergence et la précision ont été améliorées en modifiant la direction de recherche des agents.
V.Phimmasone, T. Endo, et all	[83]	Les auteurs proposent un contrôleur PSO standard pour différentes configurations du système photovoltaïque et sous divers modèles d'ombrage partiel.
M.Miyatake, M.Veerachary, et all	[84]	
M.Syedmahmoudian, S.Mekhilef, et all	[85]	

2.5.2.3.2 Contrôleur MPPT à base de l'algorithme de colonies d'abeilles (ABC_MPPT)

L'algorithme d'Optimisation par Colonies d'Abeilles (AOCA), proposé par (Pham and Ghanbarzadeh, 2007) et (Phametal., 2006) s'inspire du vol de colonies d'abeilles à la recherche de fleurs pour butiner. L'AOCA explore le voisinage des sources de nourriture pour converger, et, pour diversifier, des solutions sont créées aléatoirement.

Dans le cas de l'optimisation de puissance d'un système PV, l'emplacement de la source de nourriture représente la solution possible au problème, dans ce cas, cela correspond à la valeur du rapport cyclique du convertisseur statique, et la quantité du nectar de cette source correspond à une valeur objective dite *fitness* de chaque solution (rapport cyclique) c.a.d. la puissance $P=I.V$.

L'organigramme de l'algorithme est schématisé par la figure.2.29.

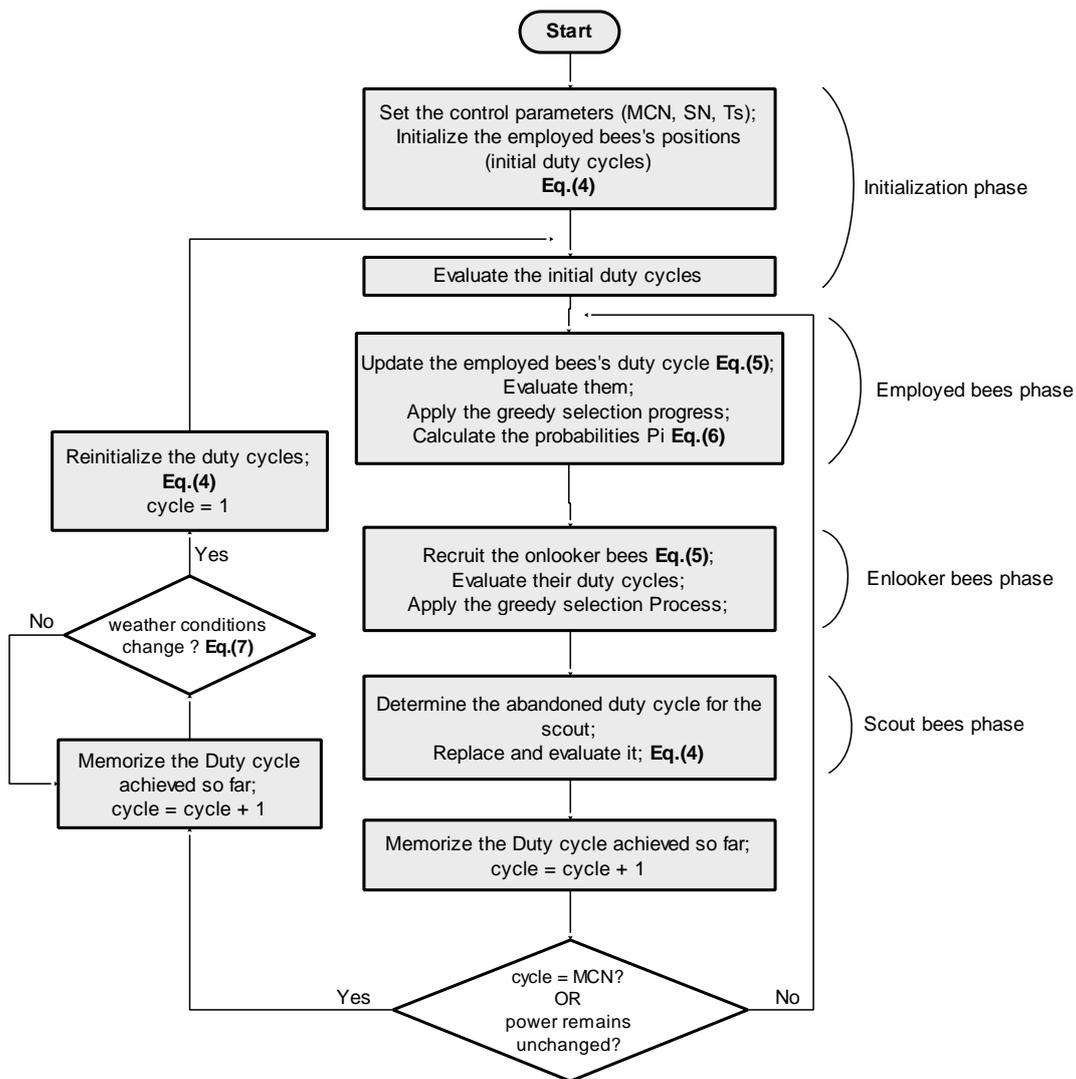


Figure.2.29 Organigramme du contrôleur ABC [86].

Le tableau.2.7 résume quelques travaux de recherches relatifs au développement de contrôleurs MPPT à base d'algorithme d'optimisation par colonies d'abeilles (AOCA_MPPT).

Tableau.2.7 Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de l'algorithme d'optimisation par colonies d'abeilles pour le développement de contrôleurs MPPT.

Auteurs	Référence	
Abou soufyane Benyoucef, et all,	[86]	Les auteurs proposent un contrôleur qui assure un suivi correct pour des conditions atmosphériques variables où une convergence appropriée est assurée. Le contrôleur MPPT utilise une méthode de contrôle directe sous ombrage partiel (GMPPT). Les performances du contrôleur sont vérifiées en analysant divers cas d'ombrage. La méthode proposée a été comparée au contrôleur PSO.
B.Bilal, et all	[87]	L'algorithme est utilisé pour le suivi du PPM, les performances sont comparées à l'algorithme P&O.
A. S.Oshaba, E. S. Ali, et all	[88]	L'algorithme ABC a été utilisé pour concevoir deux contrôleurs PI pour un système de pompe PV-DC. Les contrôleurs proposés visent à suivre et à contrôler la vitesse du système de pompe à moteur au PPM du générateur photovoltaïque.
H.Salmi, A.Badri, et all	[89]	Les auteurs proposent un contrôleur MPPT qui prend en charge l'ombrage partiel pour la recherche du PPMG. Des résultats expérimentaux sont réalisés pour valider l'approche.
Kinattungal Sundareswaran, Peddapati Sankar, et all	[90]	Les auteurs proposent un contrôleur ABC_MPPT pour le suivi du PPMG dans le cas d'ombrage partiel. Les simulations sont effectuées pour différentes configurations d'ombrage.
B. Babar et A. Crăciunescu	[91]	Le contrôleur ABC_MPPT est mis en œuvre pour suivre le PPMG d'un système PV. En outre, les résultats obtenus sont comparés à ceux des contrôleurs P&O, FLC et les AGs.

2.5.2.3.3 Contrôleur MPPT à base de l'algorithme de luciole (FA_MPPT)

L'algorithme de Luciole ou Firefly Algorithm (FA) est une technique bio-inspirée récente introduite par Xin-She Yang en 2007 [92,93], inspirée de la nature, conçue pour résoudre les problèmes d'optimisation non linéaire.

L'algorithme est basé sur le comportement d'un essaim de luciole dans la nature, en simulant le principe d'attraction entre les lucioles selon leur intensité lumineuse. Figure 2.30.



Figure.2.30 Essaim de lucioles.

L'organigramme de l'algorithme est schématisé par la figure.2.31.

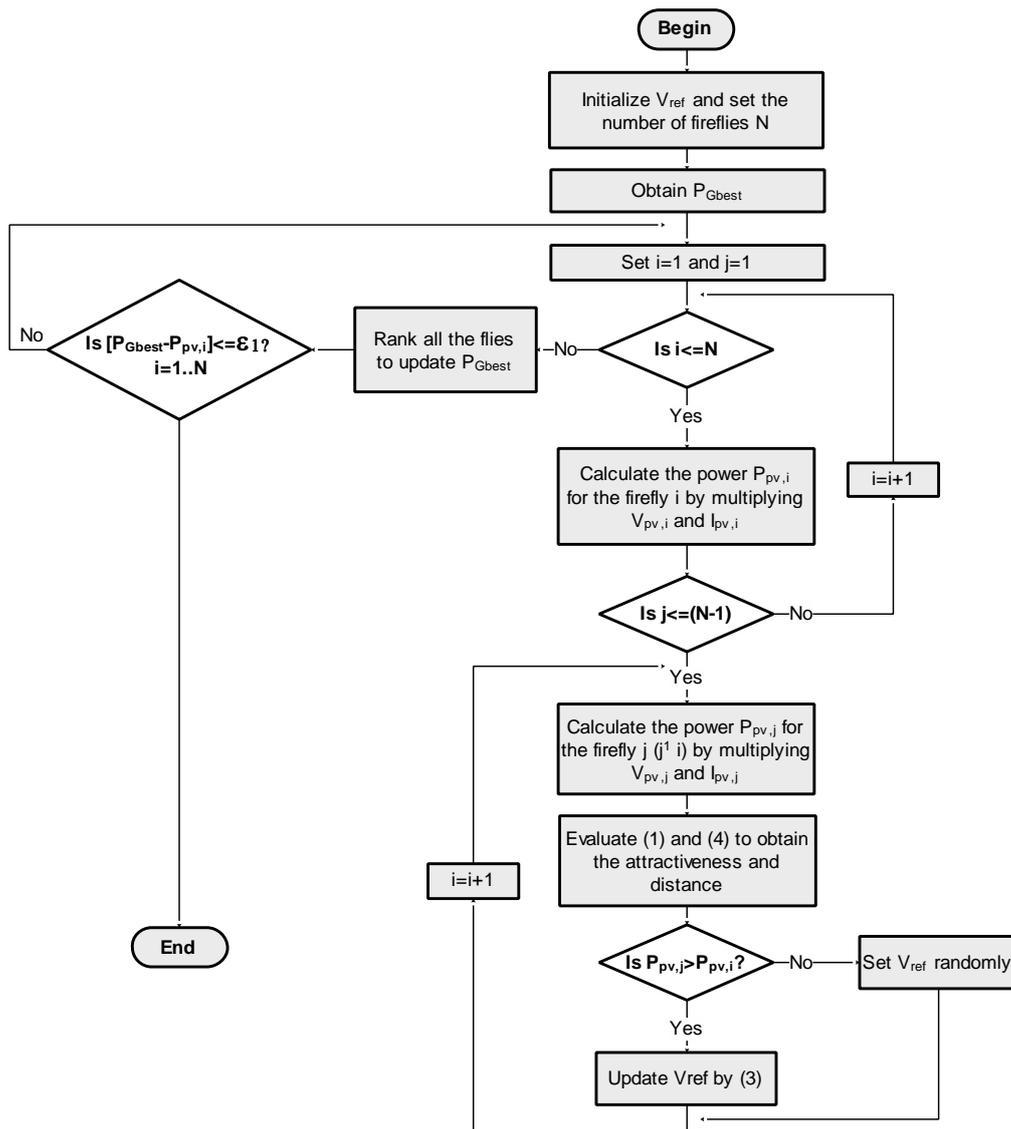


Figure.2.31 Organigramme du contrôleur FA [94].

Dans le cas de l'optimisation de puissance, la fonction objective correspond à la puissance générée par le PV. A notre connaissance, peu d'articles ont abordé le problème d'optimisation de puissance dans les systèmes PV en utilisant cette approche.

Le tableau ci-après résume les travaux recensés relatif au contrôleur MPPT à base d'algorithme luciole [94,95].

Tableau.2.8 Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de l'algorithme d'optimisation de lucioles pour le développement de contrôleurs MPPT.

Auteurs	Référence	
Kinattungal Sundareswaran, et al	[94]	Les auteurs proposent un contrôleur FA_MPPT pour différentes configurations du système photovoltaïque et sous divers cas d'ombrage partiel. Les performances du contrôleur sont comparées respectivement aux contrôleurs P&O et PSO.
Bhavana Prasad B .M A. Chitra	[95]	Les auteurs proposent un contrôleur MPPT à base de l'algorithme de luciole. Les performances du contrôleur sont comparées avec ceux de l'Incremental Conductance.
D.F.Teshome, C. H. Lee, et al	[96]	Les auteurs proposent un algorithme FA modifiée (MFA) pour réduire le nombre d'itérations et le temps de convergence vers le PPMG comparé à la méthode FA existante.

2.5.2.3.4 Contrôleur MPPT à base de l'algorithme d'optimisation de meute de loups (GWO_MPPT)

L'algorithme d'optimisation de la meute de loups, ou en anglais Gray Wolf Optimisation algorithm GWO est une proposition récente développée par Mirjalili et al. En 2014. [97]

L'algorithme est inspiré par l'optimisation de la technique d'attaque utilisée par les loups gris pendant la chasse. Il imite la hiérarchie de leadership et le mécanisme de chasse des loups gris dans la nature. (Figure.2.32).

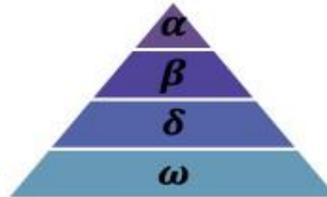


Figure.2.32 Hiérarchie au sein d'une meute de loups.

Comme l'illustre la figure ci-dessous, il y a essentiellement trois étapes pour l'algorithme GWO, à savoir la chasse, recherche de proie (repérage) en formant un groupe, puis en encerclant la proie et enfin l'attaque de la proie.

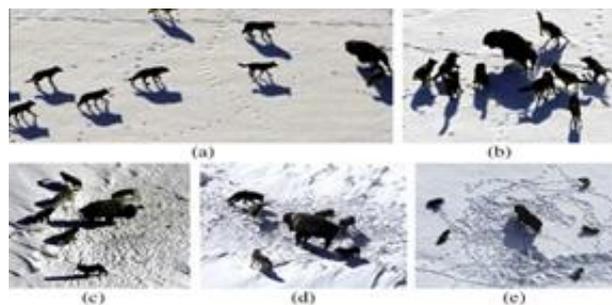


Figure.2.33 Différentes étapes : (a-c) recherche de proie, (d) repérage en cercle de la proie, (e) attaque de la proie.

Cet ensemble de mécanisme de chasse est mis en œuvre lors de la conception du GWO pour l'optimisation de puissance d'un système PV i.e. MPPT [98]. Pour l'implémentation du contrôleur MPPT, le rapport cyclique est défini comme un loup gris et la fonction objective la puissance générée par le système PV.

L'organigramme résumant les différentes étapes de l'algorithme pour une commande MPPT est illustré par la figure ci-dessous.

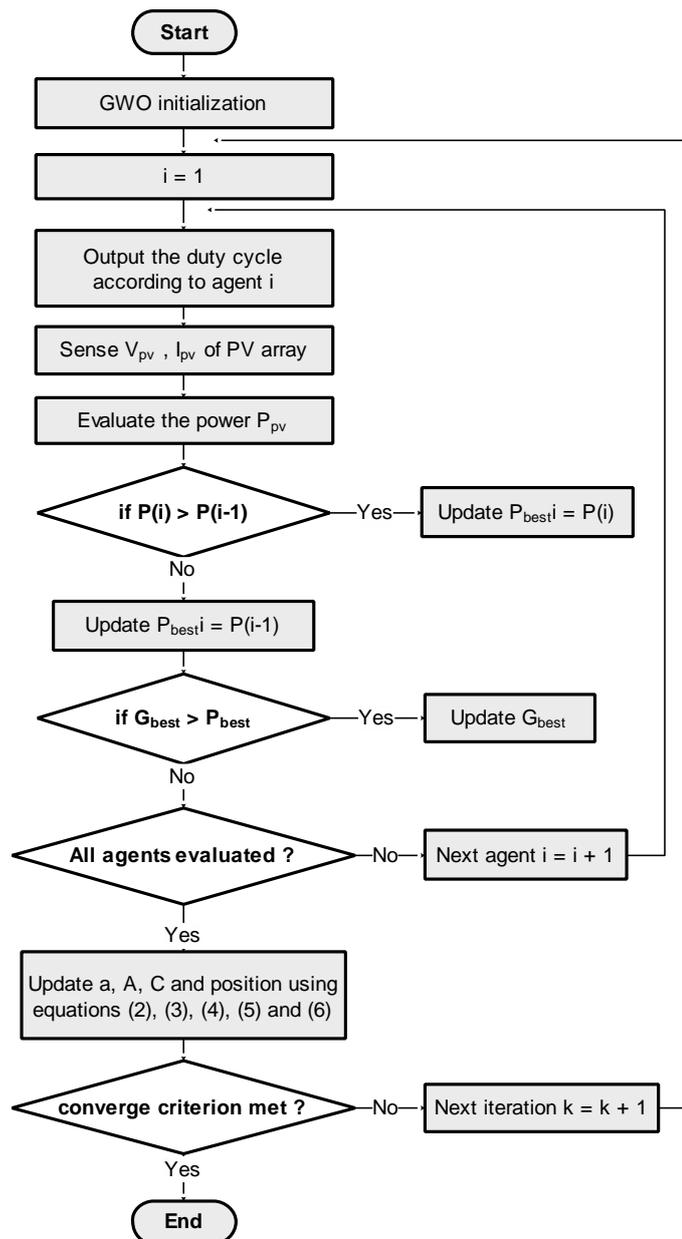


Figure.2.34 Organigramme du contrôleur GWO [98].

Le tableau.2.9 ci-après résume les travaux recensés.

Tableau.2.9 Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de l'algorithme d'optimisation de la meute de loup pour le développement de contrôleurs MPPT.

Auteurs	Référence	
Satyajit Mohanty, et al	[98]	Les auteurs proposent un contrôleur MPPT basé sur GWO algorithme. Le contrôleur prend en charge le problème d'ombrage. Les performances du contrôleur sont comparées à celle du P&O-MPPT et IPSO-MPPT. Les auteurs présentent un nouveau contrôleur opérant sous n'importe quelle variation d'insolation. Le contrôleur peut atteindre le global MPPT (GPMPT) sous toutes les conditions météorologiques. Le principal avantage de l'algorithme proposé est la substitution du contrôleur PI au système de contrôle du duty cycle.
	[99]	
Santhan kumar Cherukuri, Srinivasa Rao Rayapudi	[100]	Un contrôleur MPPT basée sur (EGWO) est proposé pour suivre le GMPP du système photovoltaïque sous ombrage partiel. Les résultats sont analysés et comparés aux algorithmes GWO et PSO. La méthode proposée est efficace pour la poursuite du GMPP avec une meilleure précision et moins de temps de calcul par rapport à PSO et GWO.

2.5.2.3.5 Contrôleur MPPT à base de l'algorithme de pollinisation des fleurs (FPA_MPPT)

L'algorithme de pollinisation des fleurs (FPA) est une nouvelle métaheuristique proposée en 2012 par Xin-She Yang [101]. La pollinisation est un phénomène qui se rapporte au transfert de pollen d'une espèce à une autre. Ce processus génère de nouvelles fleurs qui dépendent de fleurs mises en cause dans le processus de pollinisation.

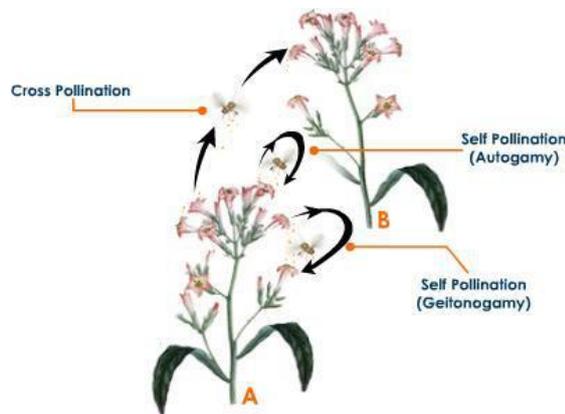


Figure.2.35 Processus de pollinisation.

Étant donné que la méthode de la FPA a deux étapes dans le calcul de la variable de contrôle : pollinisation global et locale, la méthode est bien adaptée pour l'optimisation des problèmes non linéaire tels que le contrôle MPPT. Dans ce cas, la fonction objective correspond à la puissance à maximiser et le rapport cyclique comme variable de contrôle.

L'organigramme résumant les différentes étapes de l'algorithme pour le contrôle MPPT est illustré par la figure ci-dessous.

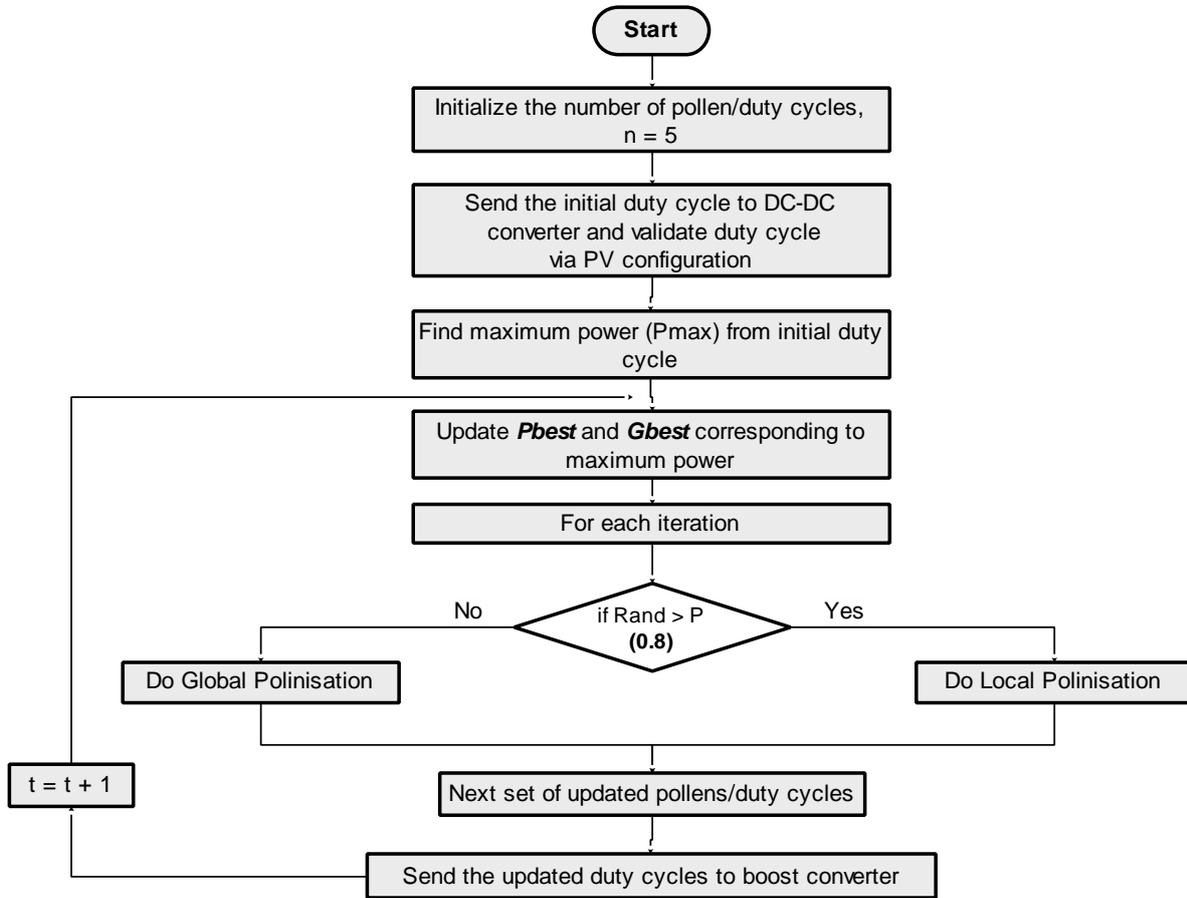


Figure.2.36 Organigramme du contrôleur FPA [102].

Le tableau.2.10 ci-dessous résume les travaux recensés pour l'optimisation de puissance à base d'algorithme de pollinisation de fleurs.

Tableau.2.10 Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de l'algorithme de pollinisation de fleurs pour le développement de contrôleurs MPPT.

Auteurs	Référence	
J. Prasanth Ram, et al	[102] [103]	Un nouvel algorithme de pollinisation des fleurs est proposé pour suivre le PPMG d'un système photovoltaïque indépendamment des conditions atmosphériques (Température, Eclairement, Ombrage).
A.Z.Diab, H.Rezk	[104]	Les auteurs proposent un contrôleur MPPT basé sur l'algorithme de pollinisation des fleurs (FBA) pour atténuer les problèmes liés à l'ombrage partiel. Une étude comparative avec les contrôleurs à base DE et PSO est établie pour différents cas d'ombrage.

2.5.2.3.6 Contrôleur MPPT à base de l’algorithme d’optimisation de ver luisant (GWSO_MPPT)

L’algorithme d’optimisation « Ver Luisant » ou Glow-worm Swarm Optimisation (GSO) est un nouvel algorithme pour l’optimisation des fonctions multimodales. Il a été proposé en premier par Krishnanand et Ghose [105], qui ont observé la communication chez les vers luisants pour échanger les informations lors de la recherche de nourriture.

Compte tenu de la présence de plusieurs pics sur la courbe caractéristique P-V causé par la non-uniformité de l’irradiation solaire (ombrage partiel), l’algorithme GSO est très approprié pour la poursuite du PPM, puisqu’il affiche des performances exceptionnelles en vue de trouver la solution optimale pour les fonctions multimodales.

L’organigramme de l’algorithme est illustré par la figure.2.37.

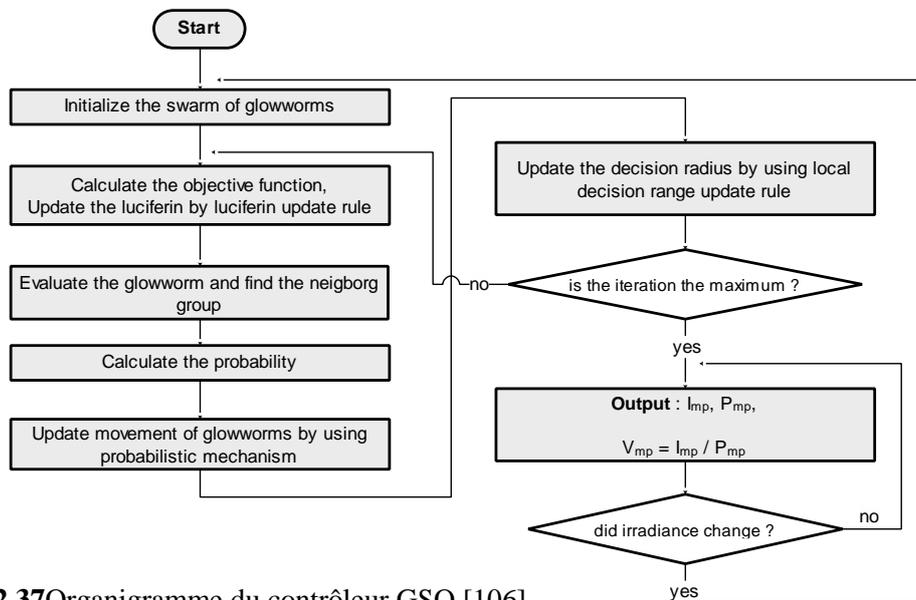


Figure.2.37 Organigramme du contrôleur GSO [106].

Cependant à l’heure actuelle, ce dernier est très peu utilisé pour le control MPPT dans les systèmes PV. Le tableau ci-dessous illustre les travaux recensés relatifs à l’optimisation de puissance à base de l’algorithme de ‘Ver Luisant’ ou Glow-worm swarm optimisation (GSO).

Tableau.2.11 Aperçu de quelques travaux de recherches issus de la littérature relatifs à l’application de l’algorithme d’optimisation de ver luisant pour le développement de contrôleurs MPPT.

Auteurs	Référence	
Yi Jin, et al	[106]	Les auteurs proposent un contrôleur GSO_MPPT pour différentes configurations du système photovoltaïque et sous divers cas d’ombrage partiel. Les performances du contrôleur sont comparées à ceux du P&O et FOCV.
W. Hou, et al	[107]	Les auteurs développent un nouveau contrôleur GSO_MPPT pour déterminer la valeur optimale de la tension de référence dans un système photovoltaïque. Les performances du contrôleur GSO proposé sont comparées au P&O et évalués en termes de convergence et précision

2.5.2.3.7 *Contrôleur MPPT à base de l'algorithme d'optimisation de l'enseignement-apprentissage (TLBOA_MPPT)*

Le concept de l'algorithme basé sur l'enseignement et l'apprentissage ou «Teaching-Learning-Based Optimization Algorithm » TLBO introduit en 2011 par Rao, Savsani, and Vakharia [108] consiste à simuler le processus d'apprentissage entre un enseignant et ses étudiants. Son principe consiste à améliorer les performances de la classe (notes des étudiants) grâce aux instructions prodiguées par l'enseignant ainsi que l'apprentissage mutuel entre les élèves [109,110]. L'organigramme de l'algorithme est illustré par la figure ci-dessous.

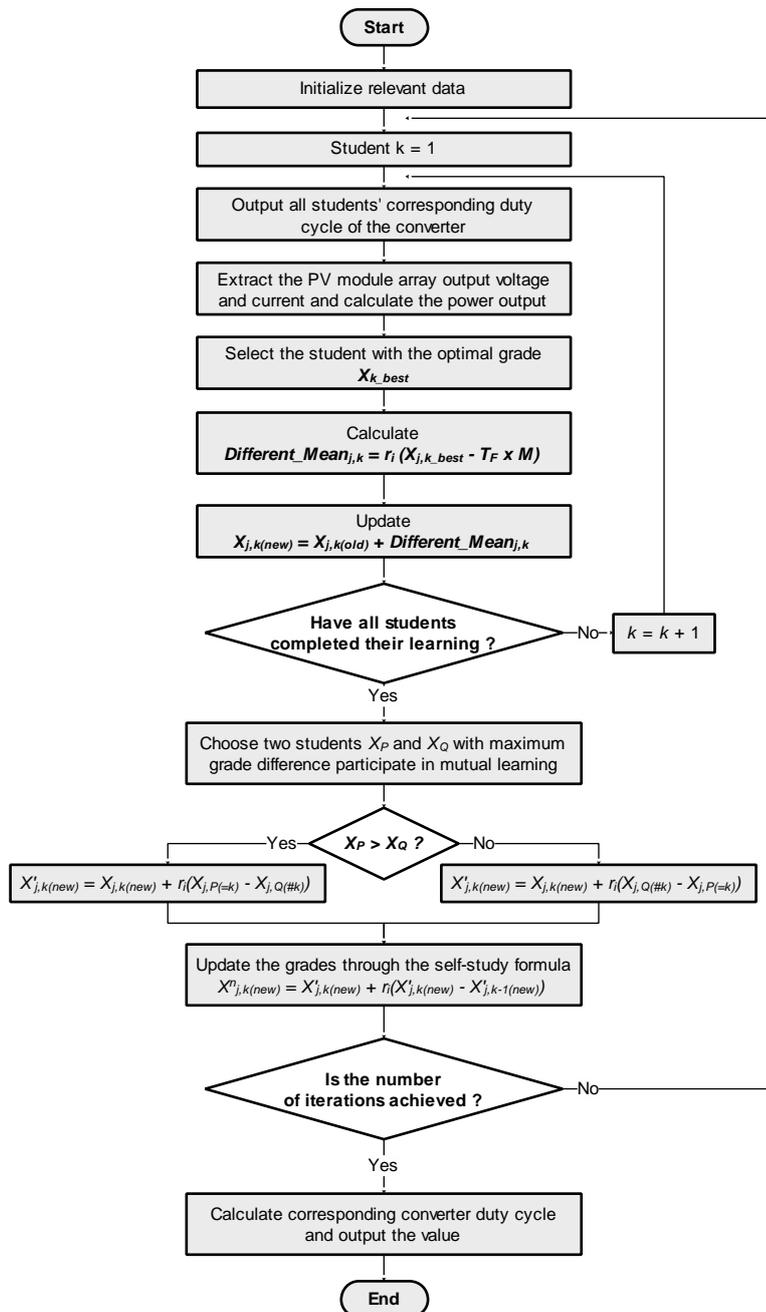


Figure.2.38 Organigramme du contrôleur TLBO [111].

Le tableau ci-dessous illustre les travaux recensés relatifs à l'optimisation de puissance à base de l'algorithme d'optimisation basée sur l'enseignement-apprentissage ou Teaching-Learning-Based Optimization Algorithm.

Tableau.2.12 Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de l'algorithme d'optimisation basée sur l'enseignement-apprentissage pour le développement de contrôleurs MPPT.

Auteurs	Référence	
Ahmed fathy, et al	[111]	L'algorithme (ITLBO) est proposée pour suivre le PPMG extrait d'un système photovoltaïque SPV dans des conditions d'ombrage partiel. Les résultats sont comparés à ceux des contrôleurs P&O, PSO et TLBO. Les résultats obtenus avec ITLBO sont plus précis et donnent une puissance de convergence rapide.
Kuei-Hsiang Chao	[112]	Les auteurs proposent une version améliorée de l'algorithme I-TLBO pour augmenter l'efficacité et les performances du contrôleur sous ombrage partie. Les résultats de simulation pour la poursuite du PPMG pour 5 patterns d'ombrage prouvés que l'I-TLBO proposé est meilleur que le TLBO conventionnelle.

2.5.2.3.8 Contrôleur MPPT à base de l'algorithme de recherche coucou (CS_MPPT)

La recherche coucou (Cuckoo Search CS) est une métaheuristique très récente. Proposée en 2009 par Yang et Deb [113]. La recherche coucou s'inspire du comportement de reproduction d'une espèce spéciale d'oiseaux parasites de nids appelés « Coucous », qui s'inspire du mode de reproduction de certaines espèces de coucous.



Figure.2.39 Oiseaux parasites de nids « Coucou ».

La figure.2.40 représente un organigramme résumant les étapes de l'algorithme de la recherche coucou.

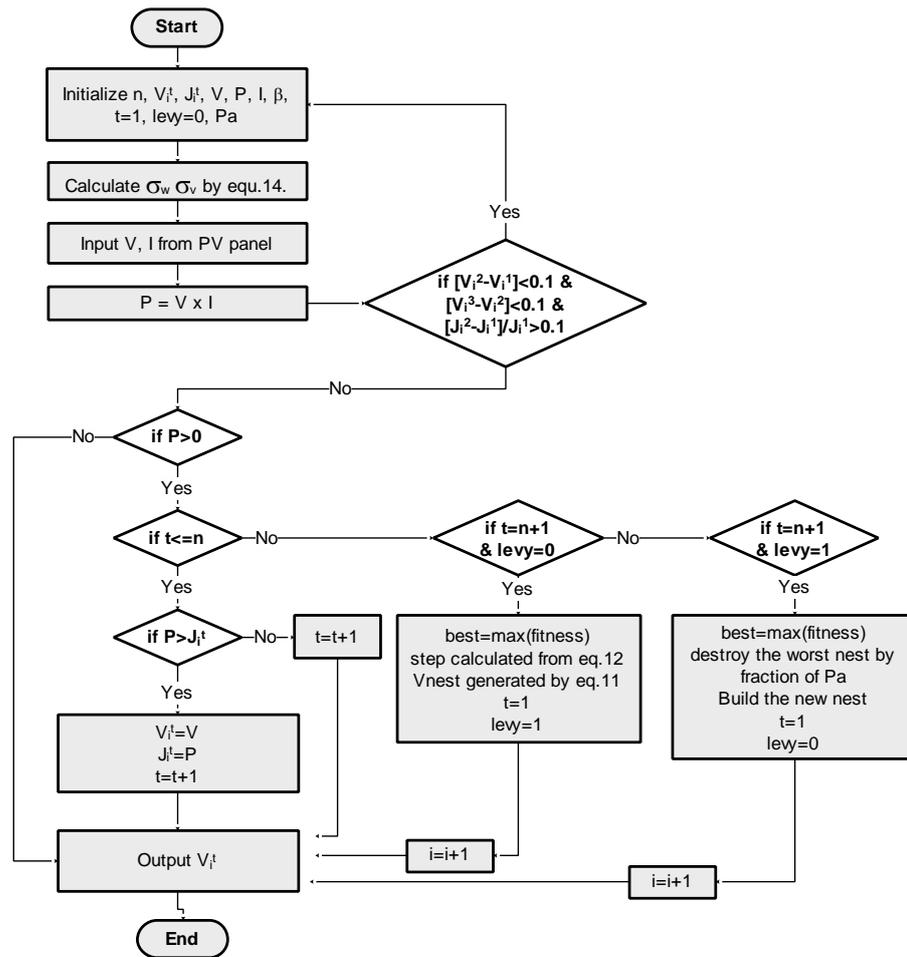


Figure.2.40 Organigramme du contrôleur CS [114].

Etant donné que la méthode est nouvelle, peu de travaux sont répertoriés dans le cadre du control et suivi du PPM dans les systèmes PV, le tableau ci-dessous illustre les travaux existants.

Tableau.2.13 Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de l'algorithme d'optimisation basée sur la recherche coucou pour le développement de contrôleurs MPPT.

Auteurs	Référence	
Jubaer Ahmed, Zainal Salam	[114]	Les auteurs développent un contrôleur qui prend en charge le problème d'ombrage partiel. Les résultats de simulations prouvent la supériorité du contrôleur comparé au P&O et PSO
J.Ahmed, Zainal Salam	[115]	Les auteurs proposent une nouvelle approche pour la poursuite du PPM utilisant l'algorithme de recherche coucou CS_MPPT. Le contrôleur prend en charge le problème d'ombrage partiel et présente des performances supérieures à celles des méthodes conventionnelles.

2.5.2.3.9 Contrôleur MPPT à base d'algorithme de saut de grenouille (SFLA_MPPT)

L'algorithme d'optimisation basé sur le saut de grenouille SFLA proposé par Dr. Eusuff en 2003 [116], est un algorithme imitant l'évolution d'un groupe de grenouilles cherchant un endroit qui dispose d'une quantité maximale de nourriture étalée aléatoirement sur des pierres dans un étang. Chaque grenouille est définie comme une solution au problème et toute la population de grenouilles est divisée en plusieurs petits groupes.

Chaque groupe de grenouilles a la capacité de juger la valeur de la source alimentaire qui est affectée à d'autres groupes. Pour chaque itération, la meilleure et la plus mauvaise grenouille de chaque groupe sont localisées. Celle qui est qualifiée comme étant la plus mauvaise est ainsi mise à jour. Après un certain nombre d'itérations, tous les groupes de grenouilles sont mélangés dans la population entière de grenouilles. Alors, elles s'échangent et partagent les informations avec chaque groupe de grenouilles mises à jour. Ces étapes se répètent jusqu'à atteindre le nombre d'itérations fixé au départ (fonction objective).

L'organigramme est illustré dans la figure.2.41 ci-dessous.

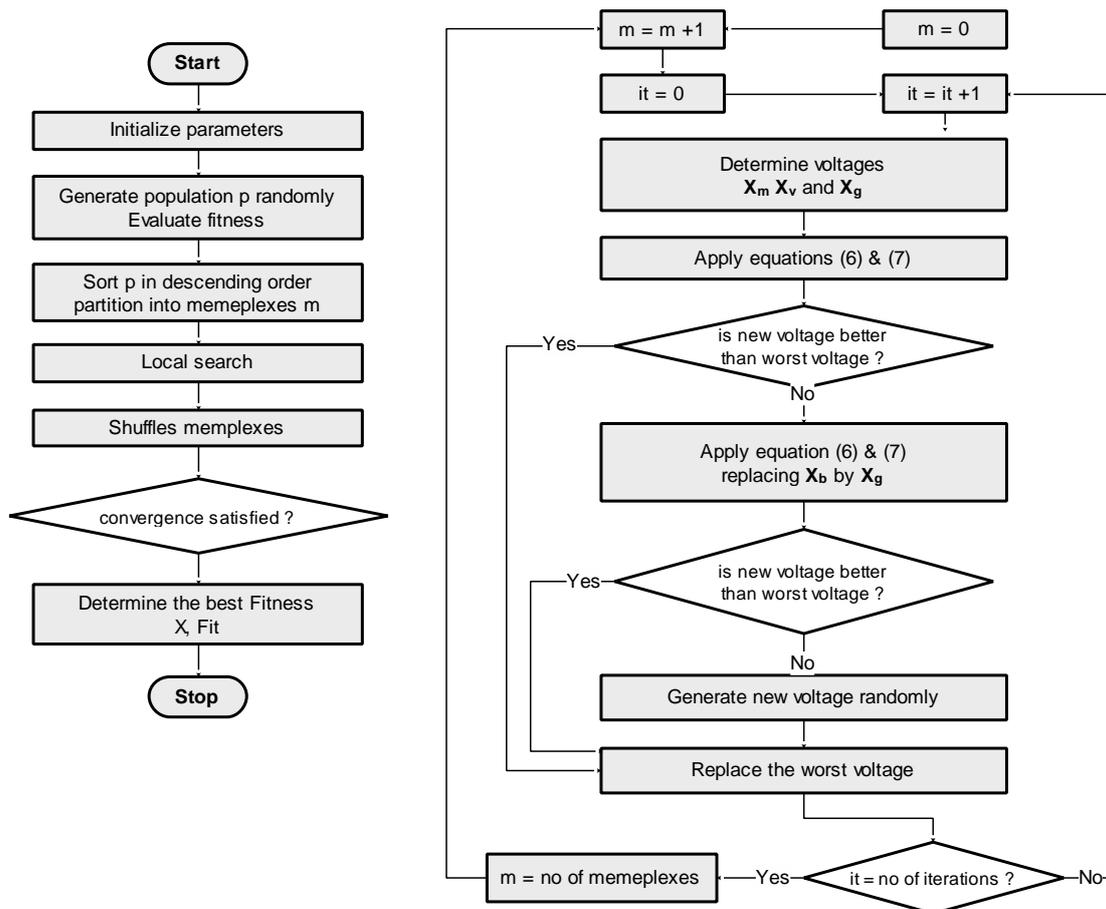


Figure.2.41 Organigramme du contrôleur SFLA [117].

Le tableau ci-dessous illustre les travaux recensés relatifs à l'optimisation de puissance à base de l'algorithme d'optimisation basé sur le saut de grenouille.

Tableau.2.14 Aperçu de quelques travaux de recherches issus de la littérature relatifs à l'application de l'algorithme d'optimisation basée sur le saut de grenouille pour le développement de contrôleurs MPPT.

Auteurs	Référence	
R. Sridhar, et al	[117]	Les auteurs proposent l'algorithme de saut de grenouille SFLA pour identifier le PPMG. Le contrôleur SFFA-MPPT est comparé à deux contrôleurs : le P&O et le PSO. Les résultats de la simulation révèlent que le contrôleur proposé présente de meilleures performances.

2.5.2.3.10 Contrôleur MPPT à base de l'algorithme d'optimisation de colonie de fourmis (ACO_MPPT)

L'algorithme de colonie de fourmis ACO a été développé par Marco Dorigo. L'algorithme est une méthode bio inspirée consistante et effective s'inspirant principalement du comportement des fourmis lors de la recherche de la nourriture [118,119].

La figure.2.42 illustre le déplacement des fourmis lors de la recherche de la nourriture.

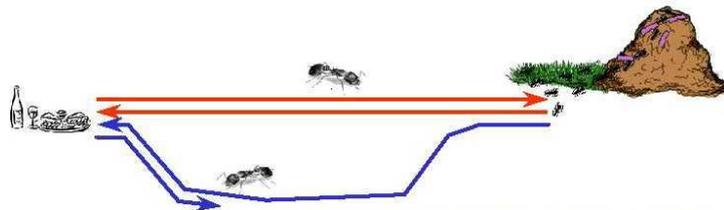


Figure.2.42 Déplacement des fourmis lors de la recherche de nourriture.

Au cours des dernières années, l'algorithme a été généralisé et étendu aux problèmes d'optimisation continue [120-124]. De ce fait, l'algorithme est bien adapté pour l'optimisation des problèmes non linéaires tels que le control MPPT. Dans ce cas, la fonction objective correspond à la puissance à maximiser et le rapport cyclique comme variable de contrôle.

L'organigramme de l'algorithme est illustré en figure.2.43.

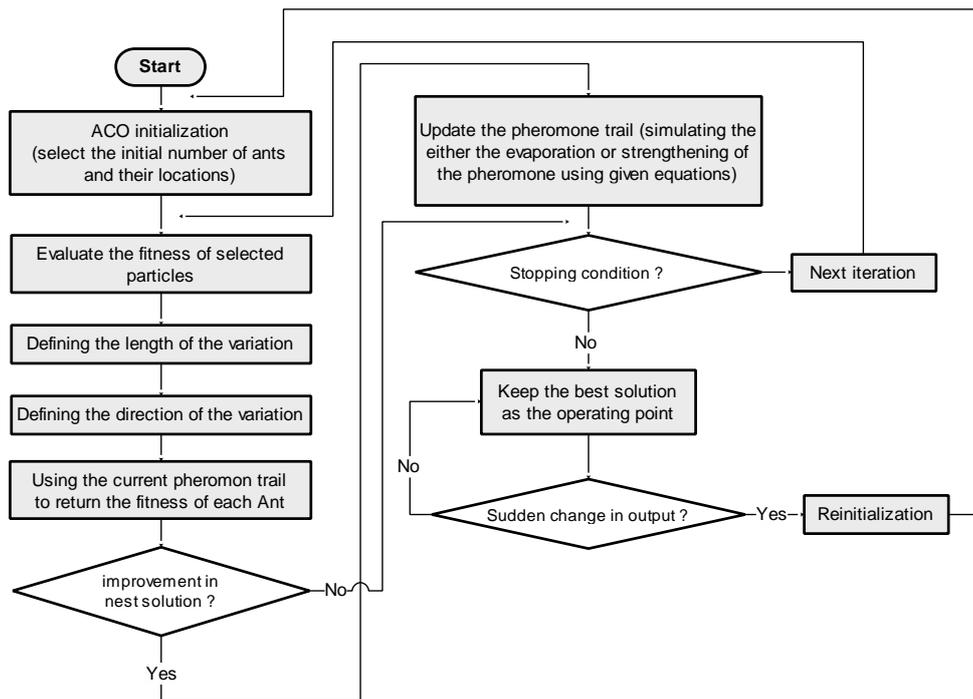


Figure.2.43 Organigramme du contrôleur ACO [127].

Le tableau ci-dessous illustre les travaux recensés relatifs à l’optimisation de puissance à base de l’algorithme de colonie de fourmis.

Tableau.2.15 Aperçu de quelques travaux de recherches issus de la littérature relatifs à l’application de l’algorithme d’optimisation basée sur la de colonie de fourmis pour le développement de contrôleurs MPPT.

Auteurs	Référence	
A.H.Besher, M.Adly	[125]	Les auteurs proposent un contrôleur pour un SPV autonome, ou la méthode <i>Fractional Open circuit</i> est utilisée pour suivre le PPM et l’algorithme ACO sert à optimiser les paramètres du contrôleur PI.
M.Adly, A.H.Besher	[126]	Les auteurs proposent un contrôleur basé une métaheuristique utilisée pour contrôler le système. En développant une méthode automatisée pour adapter les paramètres d’un contrôleur flou (FLC) et le gain d’un PID en utilisant une approche d’optimisation par colonie de fourmis. La technique est utilisée pour accélérer le suivi du MPP.
L.Jiang, L.Douglas et al	[127]	Les auteurs proposent un contrôleur MPPT pour la recherche du PPM. Le contrôleur est testé sous différents patterns d’ombrage partiel.

2.5.2.3.11 Contrôleur MPPT à base de l'algorithme d'optimisation de chauve-souris (BSO_MPPT)

L'algorithme des chauves-souris, dont l'appellation d'origine est Bat Algorithm ou Bat Swarm Optimization Algorithm (BSO), est une méthode bio inspirée très récente inspirée principalement du comportement des chauves-souris. Le premier article a été proposé en 2010 par Xin-She Yang [129].

L'algorithme proposé a été développé pour imiter le comportement d'un groupe de chauves-souris qui suivent des proies/aliments en utilisant leur capacité d'écholocation et cela même dans l'obscurité la plus totale. La figure.2.44 illustre le déplacement des chauves-souris lors de la phase de recherche de nourriture.



Figure.2.44 Déplacement de Chauve-souris lors de la recherche de nourriture.

Ce principe de recherche est appliqué pour l'élaboration de la commande MPPT, où la puissance correspond à la proie recherchée, et le rapport cyclique la variable de contrôle. L'organigramme de l'algorithme est illustré en figure.2.45.

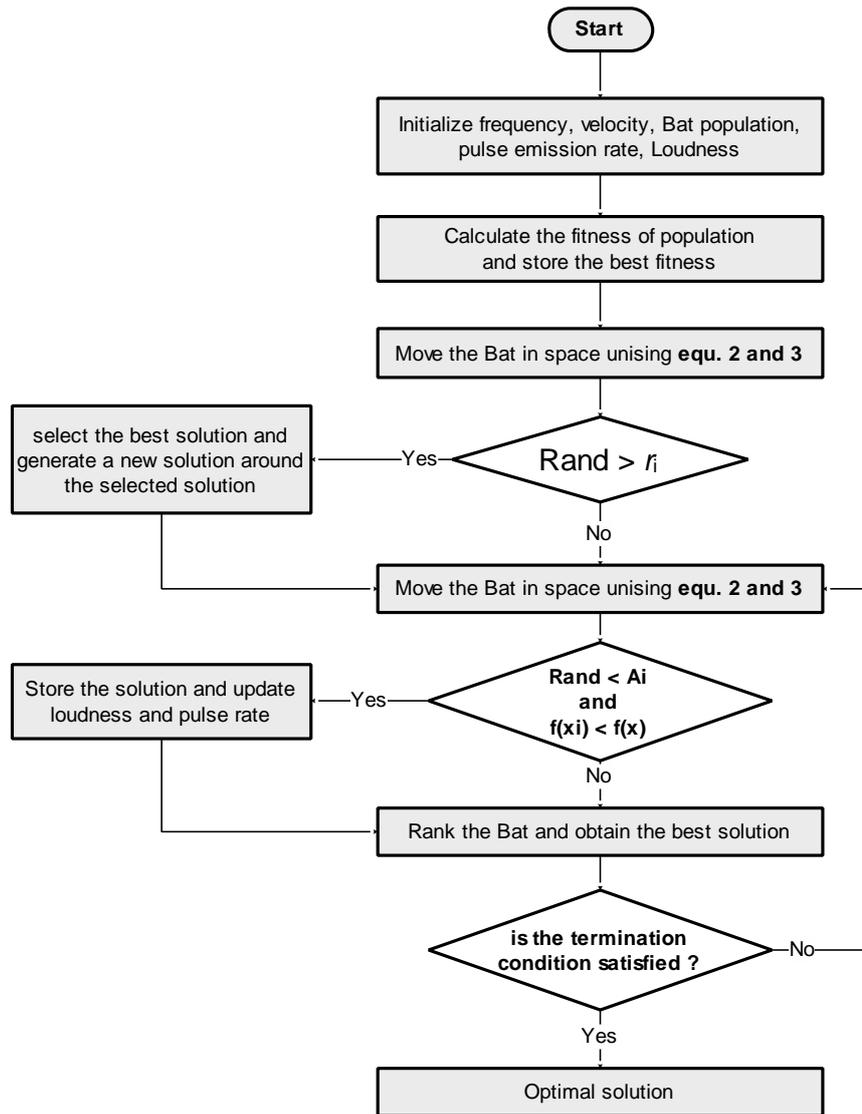


Figure.2.45 Organigramme du contrôleur BSO [130].

Le tableau ci-dessous illustre les travaux recensés relatifs à l’optimisation de puissance à base de l’algorithme de chauve-souris.

Tableau.2.16 Aperçu de quelques travaux de recherches issus de la littérature relatifs à l’application de l’algorithme d’optimisation basée sur la chauve-souris pour le développement de contrôleurs MPPT.

Auteurs	Référence	
A.S.Oshaba, E.S.Ali	[130]	Les auteurs proposent un contrôleur MPPT pour un système SPV alimenté par un moteur commuté (MRS). Ou l’algorithme de chauve-souris sert à optimiser les paramètres du régulateur PID.
	[131]	
M.K.Karagoz, H.Demirel	[132]	Les auteurs proposent un contrôleur hybride utilisant l’algorithme de chauve-souris pour déterminer la zone de localisation du PPMG, et l’algorithme P&O pour suivre le point de puissance maximum dans la zone sélectionnée.

2.6 DISCUSSION

Tous les contrôleurs cités ci-dessous ont leurs propres avantages et inconvénients. De ce fait, au travers des recherches publiées dans la littérature, il demeure difficile au concepteur d'évaluer et de sélectionner la meilleure méthode, du fait que ces derniers (contrôleurs), comme mentionné au chapitre 2.2.1, malgré le fait qu'ils soient conçus pour les mêmes objectifs, à savoir la poursuite du point de puissance maximale, diffèrent considérablement en termes de *complexité*, de *flexibilité*, du *nombre de capteurs nécessaires*, de *robustesse*, de *vitesse de convergence*, *cout*, et de *l'efficacité*.

De plus, les applications, les conditions environnementales et les bancs de test utilisés pour vérifier les performances des contrôleurs présentés dans la littérature ne sont pas similaires. Généralement, un contrôleur est sélectionné en fonction des spécifications relatives au projet (applications, conditions environnementales,...etc.). De ce fait, la connaissance de la nature du projet et ses limites est un facteur essentiel. Il est à retenir qu'un contrôleur MPPT idéal devrait :

- pouvoir atteindre le PPM ;
- être indépendant du système ;
- avoir une faible complexité d'implémentation ;
- être robuste face aux variations brusques (Température, Eclairage, Ombrage) ;
- le temps de réponse ;
- et être efficace.

Le tableau.2.17 résume conformément à la littérature les critères majeurs relatifs aux différents contrôleurs MPPT cités ci-dessus. Ainsi, nous les avons classés selon : leur *dépendance au panneau photovoltaïque utilisé (DPV)* ; le *type de capteurs utilisés* ; le *type d'implémentation (Analogique ou Digitale)* ; la *stabilité et la précision* ; la *vitesse de convergence* ; la *robustesse* et la *complexité*.

Tableau.2.17 Critères majeurs relatifs aux différents contrôleurs MPPT

				Performances							
Contrôleurs MPPT		DPV	Captueur	Complexité	Stabilité	Convergence	Robustesse	Ombrage			
MÉTHODES CONVENTIONNELLES	Méthodes indirectes	Contre réaction de tension [32-34]		Oui	V	Simple	Faible	Moyenne	Non	Non	
		Contre réaction de courant [1935,36]		Oui	I	Simple	Faible	Moyenne	Non	Non	
	Méthodes directes	Contre réaction de puissance	P&O_MPPT [37-39]		Non	I, V	Simple	Faible	Moyenne	Non	Non
			Inc Cnd_MPPT [40-42]		Non	I, V	Simple	Moyenne	Moyenne	Non	Non
MÉTHODES SOFT COMPUTING	Méthodes Intelligence Artificielle	ANN_MPPT [47-51]		Oui	I, V	Moyenne	Élevée	Rapide	Moyen	Non	
		FLC_MPPT [52-57]		Oui	I, V	Moyenne	Élevée	Rapide	Moyen	Non	
	Méthodes Hybrides	FL_GA_MPPT [58-60]		Oui	I, V	Moyenne	Élevée	Rapide	Moyen	Non	
		GA_ANN_MPPT [61,62]		Oui	I, V	Moyenne	Élevée	Rapide	Moyen	Non	
		ANFIS_MPPT [63-67]		Oui	I, V	Moyenne	Élevée	Rapide	Moyen	Oui	
	MAE	GA_MPPT [72-74]		Non	I, V	Moyenne	Elevée	Rapide	Élevée	Oui	
		DE_MPPT [72-79]		Non	I, V	Moyenne	Elevée	Rapide	Élevée	Oui	
	Méthodes Bio Inspirées	MIE	PSO_MPPT [80-85]		Non	I, V	Moyenne	Très élevée	Très rapide	Très élevée	Oui
			ABC_MPPT [86-91]		Non	I, V	Moyenne	Très élevée	Très rapide	Très élevée	Oui
			FA_MPPT [94-96]		Non	I, V	Moyenne	Très élevée	Très rapide	Très élevée	Oui
			GWO_MPPT [98-100]		Non	I, V	Moyenne	Très élevée	Très rapide	Très élevée	Oui
			FPA_MPPT [102-104]		Non	I, V	Moyenne	Très élevée	Très rapide	Très élevée	Oui
			GWSO_MPPT [106,107]		Non	I, V	Moyenne	Très élevée	Très rapide	Très élevée	Oui
			TLBOA_MPP [111,112]		Non	I, V	Moyenne	Très élevée	Très rapide	Très élevée	Oui
CS_MPPT [114,115]			Non	I, V	Moyenne	Très élevée	Très rapide	Très élevée	Oui		
SFLA_MPPT [117]			Non	I, V	Moyenne	Très élevée	Très rapide	Très élevée	Oui		
ACO_MPPT [125-127]			Non	I, V	Moyenne	Très élevée	Très rapide	Très élevée	Oui		
BSO_MPPT [130-132]		Non	I, V	Moyenne	Très élevée	Très rapide	Très élevée	Oui			

*MIA : Méthodes d'Intelligence Artificielles, MIE : Méthodes d'Intelligence par Essaims, MAE: Méthodes Algorithmes Evolutionnaire, DPV : Dépendance au Panneau photovoltaïque

Comme l'illustre le Tableau.2.17, au vu des avantages tirés des méthodes *Bio Inspirée* et de l'engouement de la part de la communauté scientifique, ces dernières présentent un avenir prometteur pour l'optimisation de puissance dans les systèmes PV, qui demeure relativement vierge au vu du nombre de travaux existants comparés aux travaux relatifs aux méthodes intelligentes et conventionnelles.

Notre intérêt portera tout particulièrement sur deux algorithmes, à savoir : l'algorithme de colonies de fourmis « Ant Colony Optimization (ACO) » et l'algorithme de la chauve-souris « Bat Swarm Optimization (BSO) ».

En particulier, l'application des deux algorithmes ACO et BSO pour l'optimisation de puissance dans les systèmes PV a retenu notre attention, pour les raisons suivantes :

- premièrement, pour le principe et les avantages des deux algorithmes ACO [118,119] et BSO [129] ;
- et deuxièmement, parce qu'il y a seulement quelques propositions relatives à l'utilisation de ces derniers pour le contrôle des systèmes PV [125-127], [130,131].

En effet, dans [125,126], les résultats ont montré une bonne capacité de poursuite, mais la complexité du système a été augmentée car l'algorithme n'est pas utilisé de façon directe, mais pour optimiser les paramètres d'un PID. De plus, le problème d'ombrage partiel n'est pas pris en compte.

En revanche dans [127], les auteurs L.L.Jiang et al, ont proposé une nouvelle méthode d'optimisation basée sur l'algorithme ACO_R [124], principalement introduite dans le but de fournir un contrôleur simple et efficace pour le suivi du PPM dans les systèmes PV sous conditions d'ombrage partiel. Les performances de l'algorithme sont vérifiées par simulation et comparées avec d'autres techniques traditionnelles MPPT. Les résultats montrent que l'algorithme proposé peut suivre efficacement le PPM face aux variations des conditions climatiques.

Cependant, le principal inconvénient de cette méthode est qu'à chaque fois qu'une variation dans les conditions climatiques se produit, l'algorithme est réinitialisé et exécuté à nouveau.

De plus, la quantité de phéromone pour l'algorithme ACO reflète l'information la plus pertinente pour la construction de la solution de la fonction objective (calcul de l'équation de mise à jour de la phéromone).

Selon [133], en règle générale, l'algorithme ACO_R appliqué à l'optimisation dans le domaine continue, nécessite plus de temps de calcul pour calculer l'équation de mise à jour des phéromones et peut converger vers un optimum local, en particulier lorsque les valeurs de l'archive sont très proches les unes des autres.

Pour surmonter ces problèmes, et réduire le temps de convergence, une nouvelle variante de l'algorithme ACO pour l'optimisation dans le domaine continue *Ant Colony with New Pheromone Updating* (ACO ACO_{NPU}) a été proposée par Serap et al. [134].

Concernant l'application de l'algorithme de chauve-souris pour l'optimisation de puissance, il y a très peu de proposition qui existe dans la littérature. En effet, dans [130,131] l'algorithme n'est pas utilisé de façon directe pour le control et la recherche du PPM, mais, ce dernier est utilisé pour optimiser les paramètres du contrôleur PID.

Dans ce contexte, en prenant acte du critère de convergence qui demeure un critère important, et afin d'améliorer les performances des systèmes PV, nous proposons deux nouveaux contrôleurs MPPT :

- le premier basé sur le nouvel algorithme ACO : *ACO_{NPU}* pour *Ant Colony Optimization with New Pheromene Update*), utilisant une nouvelle stratégie de mise à jour des phéromones, qui tend à réduire le temps de calcul lors de la mise à jour des phéromones ;
- Le second consiste en une implémentation directe de l'algorithme de chauve-souris pour le contrôle du PPM.

2.7 CONCLUSION

Dans ce chapitre, nous avons présenté un état de l'art des méthodes de recherches du PPM, à l'issue duquel, nous avons proposé une nouvelle approche pour la classification de ces différentes méthodes. Les concepts de base liés aux méthodes bio inspirés ont été introduits.

Une synthèse, à travers un tableau comparatif résumant, conformément à la littérature, les critères majeurs relatifs aux différents contrôleurs MPPT a été établie. Ce dernier (tableau) peut aider tout concepteur dans le choix pour l'élaboration d'un contrôleur en fonction des spécifications relatives au projet.

A travers cette synthèse, nous avons par la suite sélectionné les méthodes ciblées pour l'implémentation de nos deux contrôleurs. Le prochain chapitre présentera de manière détaillée chacune d'elle, depuis leurs conceptions de base jusqu'à leur implémentation.

PARTIE II

**APPROCHE SOFTWARE POUR
L'IMPLÉMENTATION DES CONTROLEURS
MPPT APPLIQUÉE AUX ALGORITHMES
ACO_NPU ET BSO**

Chapitre 3

**MÉTHODES BIO INSPIRÉES APPLIQUÉES
POUR LA RECHERCHE DU POINT DE
PUISSANCE MAXIMALE**

3.1 INTRODUCTION

Dans le chapitre précédent, nous avons proposé une nouvelle classification des différentes méthodes de recherche du PPM partant des méthodes conventionnelles (MC) aux méthodes Soft Computing (MSC) regroupant les méthodes à base d'intelligence artificielle (MIA) et les méthodes bio inspirées (MBI) utilisées pour l'optimisation du point de puissance dans les systèmes photovoltaïques.

Nous avons alors sélectionné deux algorithmes issus des méthodes bio inspirées pour l'implémentation de nos contrôleurs MPPT. Il s'agit d'une nouvelle adaptation de l'Algorithme de Colonie de Fourmis : « *Ant Colony Optimization with New Pheromon Update* » (ACO_NPU) et de l'algorithme de la Chauve-Souris « *Bat Swarm Optimization* » (BSO).

Dans ce qui suit, nous présenterons, pour chacun des deux algorithmes retenus, le principe de base et son application pour le développement du contrôleur MPPT, respectivement : *ACO_NPU_MPPT* pour le contrôleur MPPT basé sur l'algorithme de colonie de fourmis ; et *BSO_MPPT* pour le contrôleur MPPT basé sur l'algorithme de la chauve-souris.

3.2 APPLICATION DES ALGORITHMES *ACO_NPU* et *BSO* POUR LA RECHERCHE DU POINT DE PUISSANCE MAXIMALE

En règle générale, dans un système photovoltaïque, l'optimisation de la puissance revient à assurer une production maximale de celle-ci. Cela est rendu possible en tenant compte de trois paramètres importants à savoir :

- Proposer une topologie de sorte à avoir une configuration optimale des modules photovoltaïques ;
- Développer un contrôleur MPPT efficace ;
- Prendre en considération le problème des variations atmosphériques.

Dans notre travail, nous nous sommes focalisés sur les deux dernières options qui consistent à développer des algorithmes bio inspirés efficaces pour concevoir des contrôleurs MPPT assurant une production optimale dans un système photovoltaïque.

Ainsi, dans ce qui suit, nous présenterons les différentes phases de conception pour l'élaboration de nos propres contrôleurs *ACO_NPU_MPPT* et *BSO_MPPT*.

3.2.1 Conception du contrôleur ACO_NPU_MPPT

Dans cette section, nous allons présenter le principe de base de l’algorithme ACO, suivi des différentes étapes permettant l’élaboration du contrôleur pour la recherche du PPM.

3.2.1.1 Principe de base de l’algorithme ACO : de l’inspiration au modèle mathématique

Les fourmis, insectes omniprésents dans la nature, possèdent des particularités fascinantes. En effet, dans la nature, elles dévoilent une aptitude captivante à localiser un chemin entre une fourmilière et une source de nourriture. Pour l’être humain, cela semble comme étant une action anodine, alors que pour ce qui est des fourmis, cela relève de la résolution d’un problème très complexe.

En effet, étant donné que la vision du milieu dans lequel évolue une fourmi est restreinte, de ce fait, la résolution du problème pour trouver le chemin vers la source de nourriture devient compliqué. Les biologistes ont alors remarqué que lors de la phase de recherche de nourriture, les fourmis suivent une méthode très efficace et sont capables de trouver le chemin le plus court de la fourmilière vers la source de nourriture (Figure.3.1)

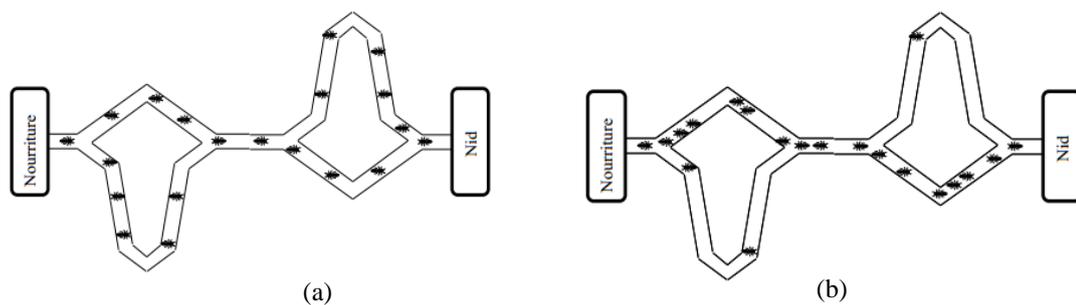


Figure.3.1 (a) Déplacement aléatoire de fourmis,
(b) Déplacement guidé par la phéromone.

En effet, lors des différentes observations, les biologistes ont remarqué que, initialement, les fourmis commencent une recherche aléatoire par l’exploration de plusieurs chemins distincts. Lorsqu’une fourmi atteint une source de nourriture, elle évalue la quantité et la qualité de la nourriture, emportant une petite quantité vers la fourmilière. Sur son chemin du retour, la fourmi dépose à chaque passage, une substance chimique odorante appelée : la *phéromone*.

Cette substance permet de créer une piste, que les autres fourmis vont suivre grâce à des capteurs sur leurs antennes.

Une communication indirecte s'établit alors entre les fourmis de façon locale. De ce fait, les autres fourmis vont être attirées par la quantité de *phéromone* déposée et vont se diriger vers la source trouvée. Au bout d'un certain temps, la totalité des fourmis vont suivre le même chemin optimal (Figure.3.2).

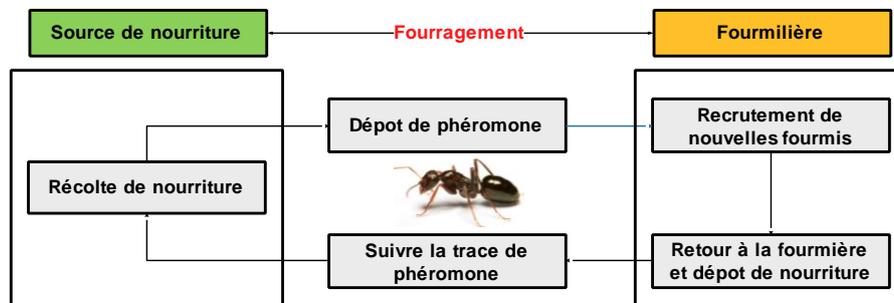


Figure.3.2. Comportement des fourmis lors de la recherche de nourriture.

Par analogie, ce comportement naturel des fourmis réelles a été modélisé par des fourmis artificielles, formalisé et adapté pour la résolution de problèmes d'optimisation.

Ainsi, une fourmi artificielle joue un rôle basique qui consiste à imiter le comportement d'une fourmi réelle qui se déplace de la fourmilière à la source de nourriture et revient. Le code qui modélise ce comportement permet alors aux fourmis artificielles de se déplacer dans l'espace de recherche constitué de différents éléments utilisés pour le problème à résoudre. Ainsi, la fourmi construit une solution.

C'est ainsi que le premier algorithme d'optimisation, basé sur les colonies de fourmis, intitulé « *Ant System* » a été proposé par Dorigo [118] en 1992, pour résoudre un problème combinatoire.

Cependant, comme l'espace de recherche dans le domaine continu diffère de celui de l'espace discret, il est à noter qu'il est difficile de simuler le déplacement des fourmis dans un graphe, ce qui a mené les chercheurs, au cours de ces dernières années, à proposer des adaptations de l'algorithme ACO pour le domaine continu [120-124].

En effet, l'algorithme ACO_R initié par Dorigo [124], qui est une généralisation de l'ACO dans le domaine continu, introduit de nouveaux mécanismes pour la mise à jour de phéromone dans le cas des problèmes continus.

Ce dernier se déroule en trois phases principales à savoir :

- **Phase d'initialisation** : afin de sauvegarder les solutions représentées par les différentes fourmis, l'algorithme ACO_R utilise une archive de solutions de taille prédéfini K en l'initialisant avec des valeurs générées aléatoirement dans l'espace de recherche continu, ces valeurs sont ordonnées selon leurs fitness ;
- **Phase de déplacement des fourmis** : durant cette étape, pour chaque itération, on génère de nouvelles positions pour l'ensemble des fourmis (x_1, x_2, \dots, x_n) en se basant sur les valeurs de l'archive et du taux de phéromones ;
- **Phase de mise à jour de l'archive** : à la fin de chaque itération, une mise à jour de l'archive est effectuée.

Comme nous l'avons mentionné dans la Section 2.6, pour le développement de notre contrôleur, nous avons sélectionné une nouvelle variante de l'algorithme ACO dans le domaine continu, ACO_{NPU} dont le principe de fonctionnement sera détaillé dans la prochaine section.

3.2.1.2 Description de l'algorithme ACO_{NPU}

L'algorithme présenté ci-dessous est basé sur la méthode ACO_{NPU} [134] pour la recherche d'extremums d'une fonction continue dans un intervalle donné. Cet algorithme est basé principalement sur un nouveau processus de mise à jour de la phéromone lors de la phase d'exécution. Cette dernière est effectuée en sélectionnant les meilleures solutions. En effet, les fourmis sont toujours attirées vers les meilleures solutions en raison de l'intensification de la valeur de la phéromone. L'algorithme ACO_{NPU} utilise une technique de recherche avec une distribution aléatoire qui est différente de l'approche ACO_R [123,124].

L'algorithme se déroule en trois étapes principales :

- Phase d'initialisation de l'archive

Afin de sauvegarder les solutions représentées par les différentes fourmis, l'algorithme utilise une archive de solutions de taille prédéfinie K .

- En premier, l'archive ($x_k, k=1 \dots \text{taille de l'archive}$) est initialisée en utilisant des valeurs générées aléatoirement dans l'intervalle $[a, b]$ (l'espace de recherche) ;
- Initialiser la fitness ;
- Positionner les fourmis ;

- Initialiser la quantité de phéromone τ_i pour chaque fourmi i par les formules suivantes :

$$D_i = |x_i - x| \quad (8)$$

$$\Phi_i = e^{-\frac{D_i^2}{2t}} \quad (9)$$

$$\tau_i = \frac{\Phi_i}{\sum_{j=1}^m \Phi_j} \quad (10)$$

D_i : distance entre la $i^{\text{ème}}$ fourmi et la meilleure solution ;
 Φ_i : une valeur gaussienne liée à la solution ;
 τ_i : la quantité de phéromone liée à la solution.

- Phase de déplacement des fourmis :

- Sélection d'un point de référence x ;
- Déplacer la $i^{\text{ème}}$ fourmi par une distance aléatoire en utilisant la relation :

$$x_i(t) = x(t - 1) + d_x \quad (11)$$

- Phase de mise à jour de l'archive :

- Garder les k meilleures solutions de l'archive selon la fitness parmi les $K+N$ solution ;
- Réinitialiser l'archive par les meilleures solutions ;
- Réinitialiser la quantité de phéromone ;
- La condition d'arrêt de l'algorithme est donnée par un nombre d'itérations fixé au préalable.

3.2.1.3 Application de l'algorithme ACO_NPU pour la commande MPPT

L'algorithme ACO_NPU décrit dans la section précédente est utilisé pour concevoir le nouveau contrôleur MPPT. En conséquence, le processus de recherche de la nourriture par les fourmis dans ACO_NPU est imité, et la fonction objective (P, puissance générée du PV) est utilisée pour trouver la tension de fonctionnement à laquelle le maximum de puissance i.e le PPM du système PV est atteint.

Ainsi, la fonction objective du système PV est donnée par l'équation :

$$P_{PV} = V_{PV} \cdot I_{PV}(T, G) \quad (12)$$

Où,

V_{PV} représente la tension du panneau ;
 I_{PV} représente le courant du panneau ;
 T représente la température du panneau ;
 G représente l'irradiation du panneau.

L'organigramme de l'algorithme proposé est décrit dans la Figure.3.3.

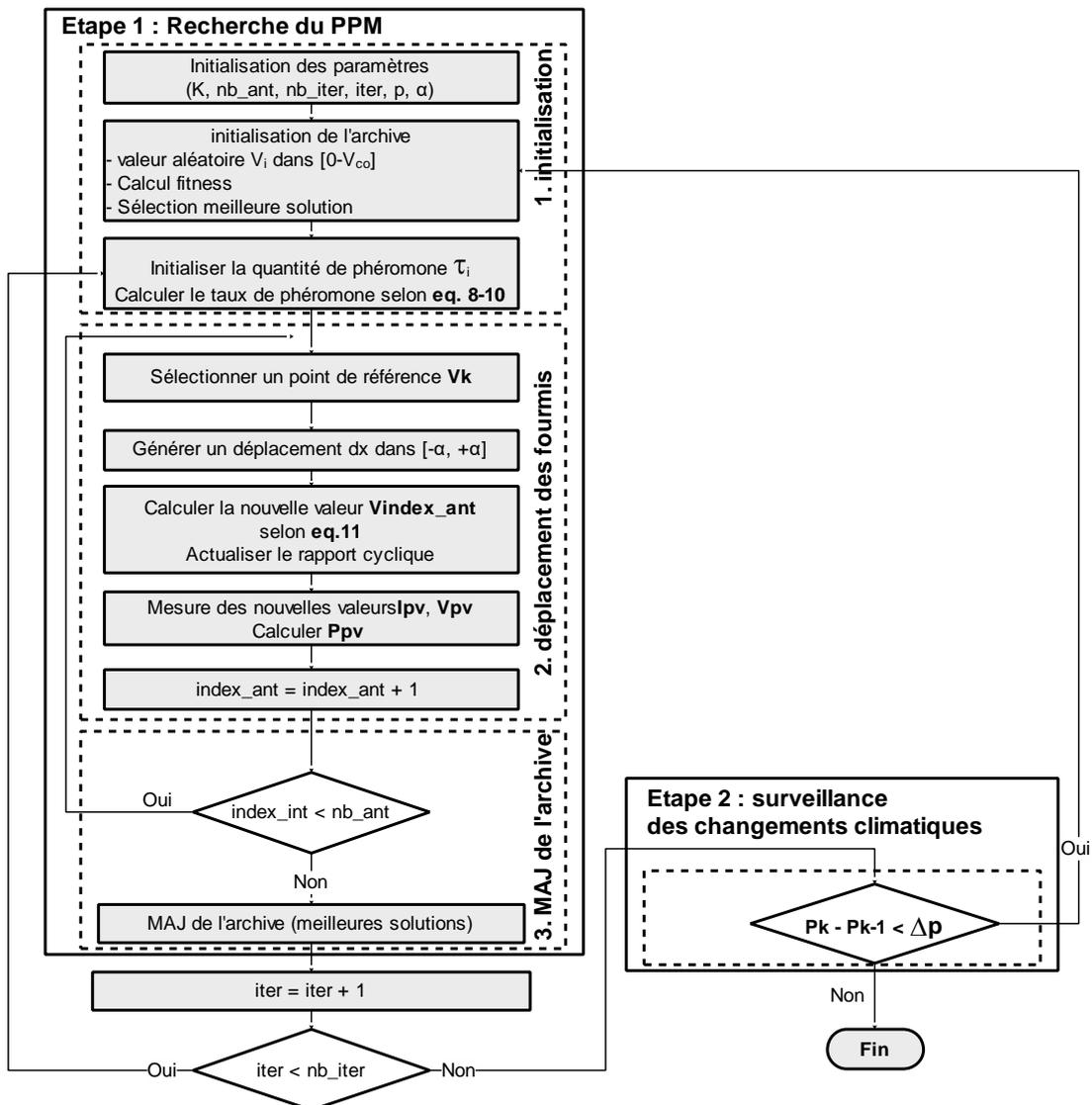


Figure.3.3 Organigramme du contrôleur ACO_NPU MPPT proposé.

L'algorithme se déroule en deux étapes :

- La première étape consiste en la recherche du PPM ;
- La deuxième consiste à contrôler les changements atmosphériques qui peuvent avoir une incidence sur le point de fonctionnement du système et réagir convenablement à cet effet.

3.2.1.3.1 Recherche du point de puissance maximale PPM

La figure.3.4 illustre les différentes étapes de recherche du PPM.



Figure.3.4 Phase de la recherche du PPM avec l’algorithme *ACO_NPU*.

Au départ, on initialise une archive dans laquelle on sauvegarde quelques solutions (une solution consiste en une valeur de la tension délivrée par le générateur photovoltaïque) ; puis à chaque itération, de nouvelles solutions sont construites en se basant sur les solutions de l’archive et les valeurs du taux de phéromone. Par la suite, à la fin de chaque itération, l’archive est mise à jour. Par itérations successives, les fourmis se déplacent vers le point d’optimisation, c’est-à-dire le PPM. Pour la fonction objective, la puissance délivrée par le système PV est utilisée comme dans la plupart des travaux. Les détails de chaque phase sont donnés ci-dessous :

- Phase d’initialisation

L’algorithme propose de stocker un ensemble de solutions dans une archive. Cette archive sert comme une référence pour la génération de nouvelles solutions liées aux différentes fourmis. La figure ci-dessous illustre les différentes étapes relatives à la phase d’initialisation.

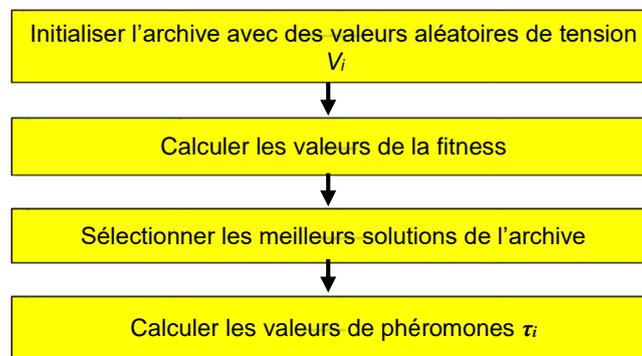


Figure.3.5 Phase d’initialisation.

Cette phase se déroule comme suit :

- On initialise l’archive avec des valeurs de tension $V_i, i=1...K$ générées aléatoirement sur l’intervalle $[0, V_{oc}]$;
- Pour chaque solution V_i , la fitness (la puissance P_i) est mesurée en changeant le rapport cyclique d_i du convertisseur DC-DC ;
- On sélectionne les m meilleures solutions de l’archive, avec m le nombre de fourmis ;
- On initialise les quantités de phéromone τ_i des m solutions (où chaque solution correspond à une fourmi) par les formules :

$$D_i = |V_i - V_{best}| \tag{13}$$

$$\Phi_i = e^{-\frac{D_i^2}{2t}} \tag{14}$$

$$\tau_i = \frac{\Phi_i}{\sum_{j=1}^m \Phi_j} \tag{15}$$

D_i : la distance entre la $i^{\text{ème}}$ solution V_i et la meilleure solution V_{best} ,

Φ_i : une valeur gaussienne liée à la solution V_i

τ_i : la quantité de phéromone liée à la solution V_i

- Phase de déplacement des fourmis

Le déplacement des fourmis est conditionné par les valeurs de phéromone τ_i précédemment calculées ; pour chaque fourmi :

- On sélectionne un point de référence V_k (la solution à laquelle la fourmi est attirée) selon la probabilité τ_k .
- Une fois le point de référence $V_k (t-1)$ sélectionné, la position de la $i^{\text{ème}}$ fourmi est obtenue par la relation :

$$V_i(t) = V_k(t - 1) + d_x \tag{16}$$

d_x : variable générée aléatoirement sur l'intervalle $[-\alpha, \alpha]$.

- Phase de mise à jour de l'archive

A la fin de la deuxième étape de l'algorithme, m nouvelles solutions sont générées ; on garde dans l'archive les K meilleures solutions parmi tous l'ensemble $(m + K)$. Une fois l'archive mise à jour par les meilleures solutions, on sélectionne à nouveau les m meilleures solutions et on met à jour les quantités de phéromone avec les formules (13-15). La condition d'arrêt de l'algorithme est donnée par un nombre d'itérations fixé au préalable ou bien s'il n'y a pas de grands changements dans la valeur de puissance (fitness) :

$$|P_k - P_{k-1}| < P_{thr} \tag{17}$$

La Figure.3.6 illustre respectivement les différentes phases de l'algorithme :

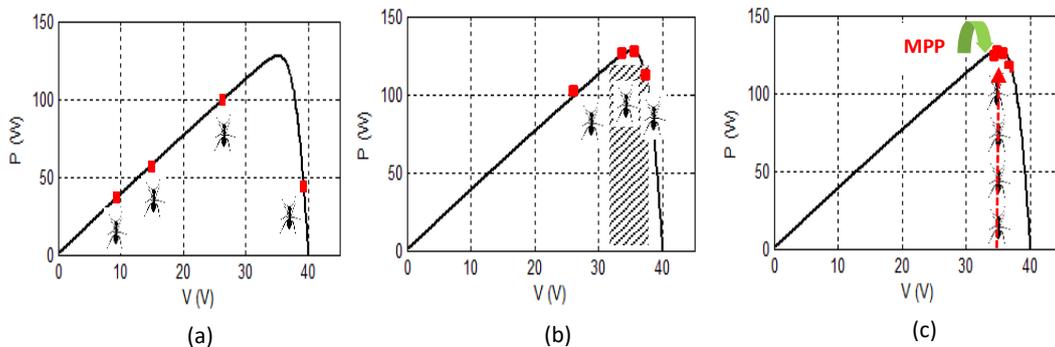


Figure.3.6 (a) Initialisation et distribution fourmis selon fonction objective, (b) les fourmis se déplacent vers un espace de recherché restreint, (c) les fourmis convergent vers le point optimum, i.e. the MPP.

3.2.1.3.2 Surveillance des changements climatiques

L'algorithme ACO-NPU développé par [134] est plutôt dédié aux problèmes où la fonction objective ne change pas en fonction du temps. De ce fait, ce dernier ne prend pas en compte les changements climatiques (augmentation ou diminution de l'irradiation solaire).

Afin de l'adapter et prendre en compte ces situations, nous avons pensé à réinitialiser les valeurs de l'archive et à ré-exécuter l'algorithme de nouveau. La réinitialisation de l'archive est une étape déterminante pour la convergence de l'algorithme au nouveau PPM.

La Figure.3.7 qui est inspirée de [134], résume le processus d'exécution de l'algorithme ACO_NPU MPPT.

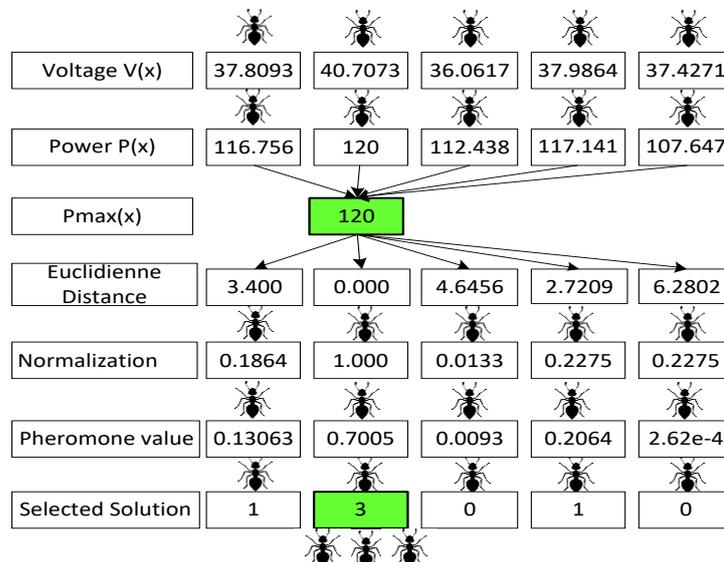


Figure.3.7 Processus d'exécution de l'algorithme proposé ACO_NPU MPPT.

3.2.2 Conception du contrôleur BSO_MPPT

3.2.2.1 Principe de base de l'algorithme BSO : de l'inspiration au modèle mathématique

La chauve-souris est l'un des mammifères qui a longtemps fasciné l'Homme de par son mode de vie si peu connu. De nos jours, cette espèce encore méconnue, intéresse de plus en plus la communauté des scientifiques.

Sa capacité de vol lui permet de voler la nuit en évitant les obstacles grâce à son système d'écholocation très perfectionné. En émettant des pulsations sonores très fortes : ultrasons, elles sont en mesure de trouver et de distinguer différents types d'insectes tout en évitant les obstacles probables, et ce, même dans l'obscurité la plus totale.

Dès que ces ultrasons rencontrent des obstacles (végétations, constructions, proies ...autres.), ils rebondissent vers la chauve-souris qui les capte grâce à ses oreilles. Son cerveau va alors traiter l'information en calculant la forme de l'objet détecté, sa position, sa distance et sa vitesse. (Figure3.8).

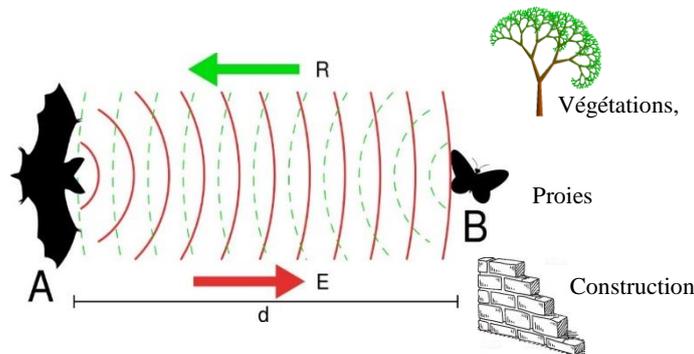


Figure.3.8 Echolocalisation des chauves-souris.

Les chauves-souris volent aléatoirement avec une vitesse v_i à la position x_i avec une fréquence f_{min} , longueur d'onde λ et l'intensité A_0 pour la recherche d'une proie dans l'espace de recherche.

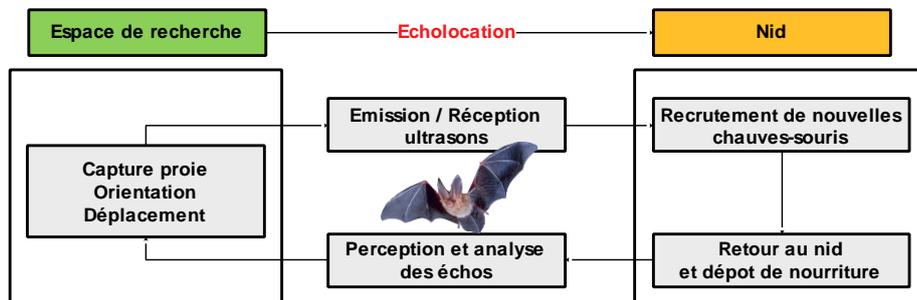


Figure.3.9 Comportement des chauves-souris dans un espace de recherche.

Ce comportement naturel d'écholocalisation des chauves-souris réelles a été modélisé par des chauves-souris artificielles, formalisé, associé à une fonction objective et adapté à la résolution de problèmes d'optimisation.

3.2.2.2 Description de l'algorithme BSO

L'algorithme de la chauve-souris se déroule en trois étapes essentielles :

- Phase d'initialisation de la population

Tout comme pour l'algorithme de colonie de fourmis, pour sauvegarder les solutions représentées par les différentes chauves-souris, l'algorithme utilise une archive de solutions de taille prédéfinie K .

- En premier, l'archive ($x_k, k=1\dots$ taille de l'archive) est initialisée en utilisant des valeurs générées aléatoirement dans l'intervalle $[a, b]$ (l'espace de recherche) ;
- Par la suite, il faut calculer la fitness ;
- En troisième lieu, positionner les chauves-souris selon la position x_i , vitesse v_i et la fréquence f_i et identification de la meilleure solution ;

- Phase de génération de nouvelles solutions

Au cours de cette phase, et pour chaque itération, une mise à jour de la position x , de la vitesse v_i et de la fréquence f_i est nécessaire en utilisant les équations (18-20).

Cette opération permet de générer de nouvelles solutions de sorte que chaque chauve-souris se rapproche de la cible.

$$f_i = f_{min} + (f_{max} - f_{min}) \cdot \delta \quad (18)$$

$$v_i^t = v_i^{t-1} + (x_i^t - x_*) \cdot f_i \quad (19)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (20)$$

Où :

- f_i : correspond à la fréquence d'émission de la chauve-souris x_i , et elle se trouve toujours entre f_{min} et f_{max} . Cet intervalle de fréquence correspond à la gamme de longueurs d'onde des chauves-souris $[\lambda_{min}, \lambda_{max}]$.
- δ : un nombre compris entre 0 et 1, pris aléatoirement lors de l'exécution de l'algorithme.

- Phase d'identification de la meilleure solution

Au cours de cette phase, et afin de se rapprocher de leur but (proie,...) rapidement, les chauves-souris se positionnent au tour de la meilleure solution identifiée à chaque fois qu'elles reçoivent un taux d'impulsion supérieure au taux d'impulsion émis.

$$x_{new} = x_{old} + \epsilon A^t \quad (21)$$

Ainsi elles sauvegardent les nouvelles solutions, diminuent leurs amplitudes (équ.22) et augmentent leur taux d'impulsion (équ.23) pour chaque identification d'une nouvelle meilleure solution en volant aléatoirement.

$$A_i^{t+1} = \beta \cdot A_i^t \quad (22)$$

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)] \quad (23)$$

Avec :

$\epsilon \in [-1,1]$, est un nombre aléatoire.

A^t , correspond à la moyenne de l'intensité de toutes les chauves-souris à l'instant t .

A_i^t, A_i^{t+1} , correspondent respectivement à l'intensité de la chauve-souris x_i à l'instant t et $(t+1)$.

γ, β , constantes définies lors de la phase de paramétrage de l'algorithme avec :

$$\gamma > 0 \text{ et } 0 < \beta < 1.$$

r_i^0 et r_i^{t+1} correspondent respectivement au taux d'impulsion initial de la chauve-souris x_i , à l'instant t et $(t+1)$.

3.2.2.3 Application de l'algorithme BSO pour la commande MPPT

L'algorithme BSO décrit dans la section précédente est utilisé pour concevoir un nouveau contrôleur MPPT afin d'optimiser la puissance dans un système PV. En conséquence, par analogie au processus de recherche de la nourriture par les chauves-souris dans le BSO, la fonction objective (P, puissance générée du PV) est utilisée pour trouver la tension de fonctionnement à laquelle le maximum de puissance i.e le PPM du système PV est atteint.

Ainsi, la fonction objective du système PV est donnée par l'équation :

$$P_{PV} = V_{PV} \cdot I_{PV}(T, G) \quad (24)$$

Où, V_{PV} représente la tension du panneau ;
 I_{PV} représente le courant du panneau ;
 T représente la température du panneau ;
 G représente l'irradiation du panneau.

L'organigramme de l'algorithme BSO proposé est décrit dans la Figure.3.10.

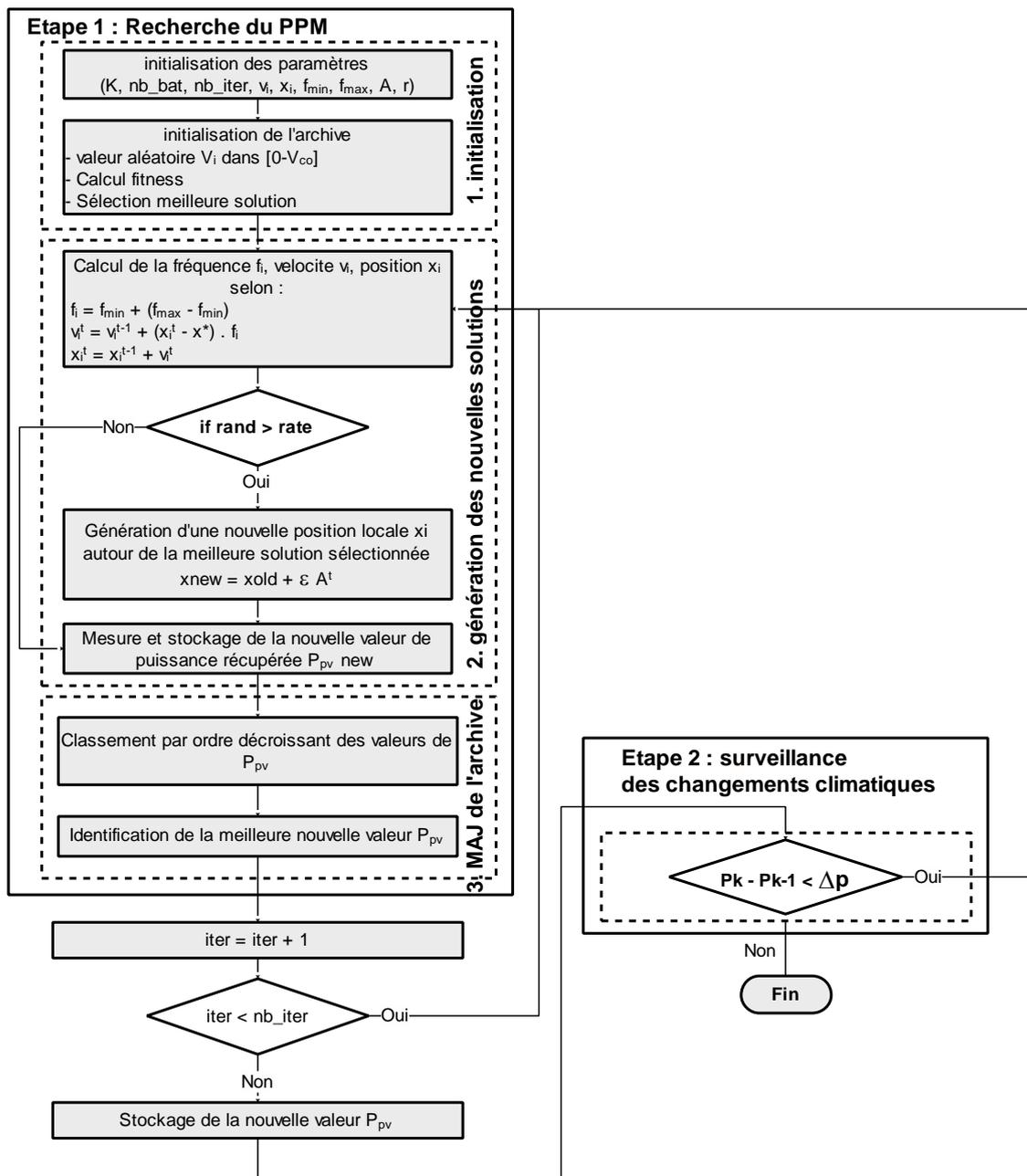


Figure.3.10 Organigramme de contrôleur BSO MPPT proposé.

3.2.2.3.1 Recherche du point de puissance maximale PPM

Comme mentionné précédemment, et comme illustré dans la Figure.3.11, la recherche du PPM s’effectue en trois étapes distinctes :

L’étape d’initialisation des paramètres de l’algorithme et de l’archive (population), l’étape de déplacement de la population avec la génération de nouvelle solution et l’étape d’identification des meilleurs solutions correspondant à la mise à jour de l’archive.



Figure.3.11 Phase de la recherche du PPM avec de l'algorithme BSO.

- Phase d'initialisation de la population

Au cours de cette phase, l'algorithme BSO utilise une archive pour stocker l'ensemble des valeurs 'tension : V_{PV} ', $V_i, i=1...K$ générées aléatoirement dans l'espace de recherche $[0, V_{co}]$.

Ces dernières sont utilisées comme référence pour la génération de nouvelles solutions.

La figure ci-dessous illustre les différentes étapes.

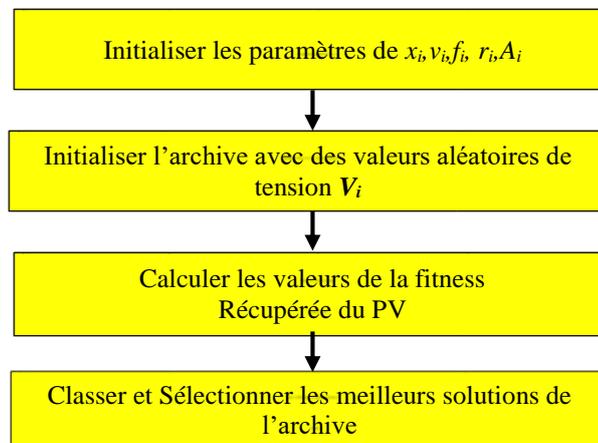


Figure.3.12 Phase d'initialisation.

Cette phase se déroule comme suit :

- On initialise l'archive avec des valeurs de tension $V_i, i=1...K$ générées aléatoirement sur l'intervalle $[0, V_{oc}]$;
- Pour chaque valeur V_i générée et stockée, la fitness ($P_i PV$) est mesurée en changeant le rapport cyclique d_i du convertisseur DC-DC ;
- L'algorithme classe les valeurs de fitness de la plus grande à la plus petite valeur, et sélectionne les meilleurs solutions de l'archive.

L'initialisation se termine lorsque le nombre de la taille de l'archive est atteint.

- Phase de génération de nouvelle solution

Afin que les chauves-souris se rapprochent le plus de leurs cibles, à savoir le PPM, l’algorithme génère de nouvelles solutions en mettant à jour la position x_i , la vitesse v_i et la fréquence f_i , de chaque chauve-souris en fonction de la meilleure valeur identifiée pour chaque itération. (équ.18-20 de la section 3.2.2.2).

La figure ci-dessous illustre les différentes étapes :

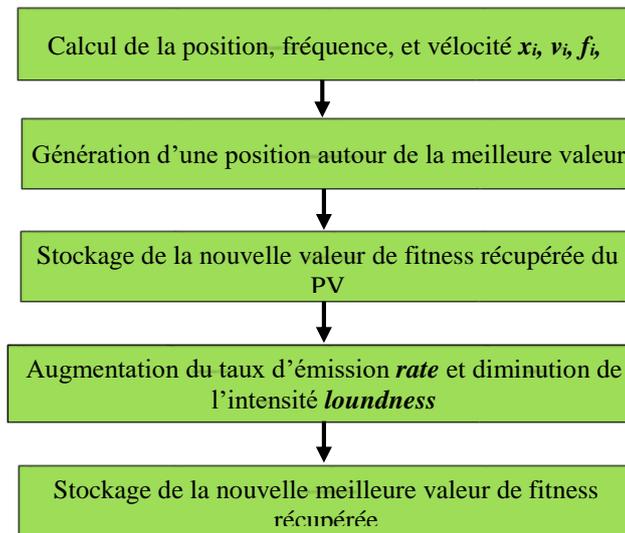


Figure.3.13 Phase de génération de nouvelle solution.

Dans notre cas, les valeurs des tensions correspondent aux positions de chacune des chauves-souris. De ce fait, si :

- La position de la chauve-souris est hors du domaine de recherche identifié au départ, l’algorithme récupère la chauve-souris dans le domaine de recherche.
- La vitesse d’impulsion perçue par la chauve-souris est supérieure à sa vitesse d’impulsion émise, l’algorithme oriente la chauve-souris vers la meilleure valeur trouvée. (équ.21 de la section 3.2.2.2).

Par la suite, la fitness correspondante à la puissance délivrée par le système PV pour la position de chaque chauve-souris est récupérée afin de sélectionner une nouvelle solution meilleure que les précédentes. Cette dernière est considérée comme étant la meilleur solution si :

- L’amplitude d’impulsion perçue est inférieure à l’amplitude d’impulsion émise de la chauve-souris ;
- La valeur de la nouvelle solution *fitness* doit être supérieure à la *fitness* de la chauve-souris stockée précédemment et cela pour chaque chauve-souris.

Une fois que ces deux conditions sont vérifiées, à savoir, que chaque chauve-souris se rapproche de plus en plus vers la meilleure valeur (solution), alors l'algorithme diminue l'amplitude d'impulsion (équ.22 de la section 3.2.2.2) et augmente la vitesse d'impulsion (équ.23 de la section 3.2.2.2) de la chauve-souris correspondante.

A chaque itération, la phase de génération de nouvelles solutions est répétée pour toute la population afin d'identifier de nouvelles solutions considérées comme étant meilleures en vue d'atteindre le PPM du système PV.

- Phase d'identification de la meilleure solution

Au cours de cette phase, l'algorithme classe les nouvelles meilleures solutions stockées précédemment par ordre décroissant avec les valeurs de tension correspondante, ainsi que la valeur correspondante à la meilleure solution courante.

Avant de passer à la prochaine itération et générer de nouvelles solutions, l'algorithme remplit de nouveau l'archive, afin d'actualiser l'archive avec les meilleures nouvelles solutions ordonnées.

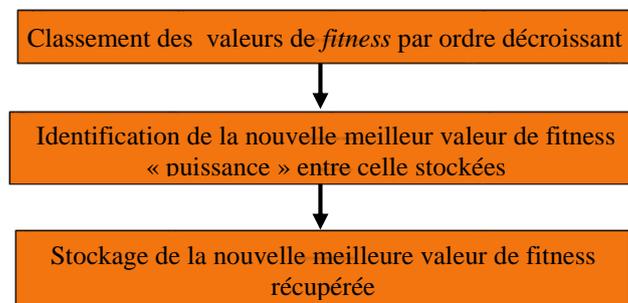


Figure.3.14 Phase d'identification de la meilleure solution.

En résumé comme l'illustre la figure.3.15, l'algorithme BSO pour l'optimisation de puissance se déroule en trois phases, au départ les chauves-souris survolent aléatoirement l'espace de recherche entre $[0, V_{co}]$.

Par la suite, ces dernières se dirigent vers la meilleure solution petit à petit dans la deuxième et la troisième phase jusqu'à atteindre un certain nombre d'itérations fixé au départ et de ce fait identifier le PPM du système PV.

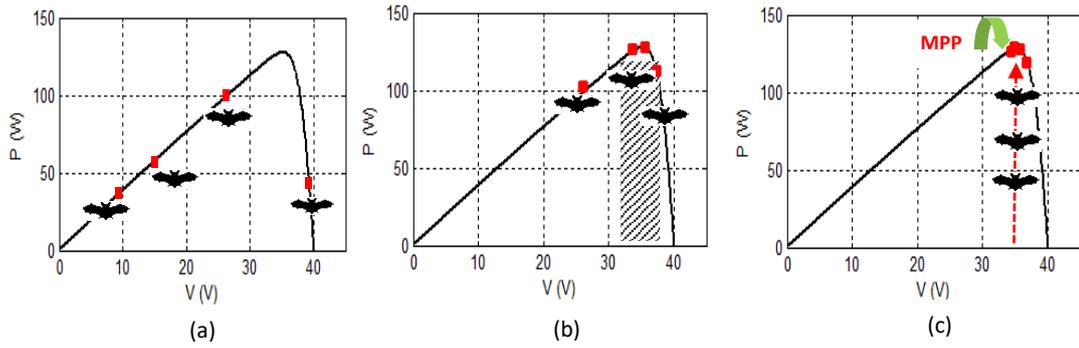


Figure.3.15 (a) Initialisation et distribution de la population,
 (b) génération de nouvelle solution et direction vers la meilleurs,
 (c) chauve-souris en direction du PPM.

3.3 CONCLUSION

Dans ce chapitre, nous avons exposé tout d'abord les principes de base des deux algorithmes ainsi sélectionnés à savoir : l'algorithme de colonie de fourmi et l'algorithme de chauve-souris. Par la suite, nous avons présenté notre contribution qui consiste à concevoir et développer les nouveaux contrôleurs *ACO_NPU_MPPT* et *BSO_MPPT*.

La mise en œuvre de ces deux algorithmes et l'évaluation des performances des deux contrôleurs ainsi développés seront étudiées dans le détail au *chapitre 4*.

Chapitre 4

**MISE EN ŒUVRE ET SIMULATION DES
CONTROLEURS ACO_NPU ET BSO MPPT :
ÉTUDE COMPARATIVE DES MÉTHODES
CONVENTIONNELLES, INTELLIGENTES ET
BIO INSPIRÉES**

4.1 INTRODUCTION

Afin de mettre en avant les deux contrôleurs MPPT ainsi développés issus des méthodes bio inspirées, à savoir, *ACO_NPU_MPPT* et *BSO_MPPT*, dans ce qui suit, nous présenterons la méthodologie adoptée pour la simulation de ces derniers.

Comme mentionné précédemment dans la section 2.2.1, il existe plusieurs critères comparatifs permettant de mettre en avant les performances d'un contrôleur MPPT. La figure. 2.1 illustre une proposition de sélection des principaux critères que les concepteurs devront considérer pour la sélection d'un contrôleur MPPT performant. Dans ce contexte, et afin d'évaluer les performances des deux contrôleurs développés, nous avons choisi de prendre comme référence les critères relatifs au *temps de réponse*, à la *stabilité face aux oscillations*, à la *précision* et la *robustesse* face aux variations atmosphériques.

Pour ce faire, nous avons suivi une méthodologie qui consiste, dans un premier temps, à modéliser le système PV, puis dans un deuxième temps et à l'issue de plusieurs tests et simulations, nous avons procédé au paramétrage des deux contrôleurs. Cela nous a permis de sélectionner les meilleurs paramètres (taille de la population : fourmis/chauves-souris, taille de l'archive, taux de phéromone, fréquence, ...etc.) pour l'élaboration d'un bon contrôleur et cela pour chacun des deux.

Suite à cela, en tenant compte des critères mentionnés précédemment, nous avons élaboré plusieurs scénarios afin de suivre l'évolution et le comportement des deux contrôleurs face aux différentes variations atmosphériques.

Pour finir et afin de justifier encore une fois le choix de nos deux contrôleurs issus des méthodes bio inspirées, nous avons effectué une étude comparative en termes de performances.

Pour cela, nous avons sélectionné des contrôleurs issus des méthodes conventionnelles et intelligentes, à savoir : *P&O_MPPT* [52,58], *ANN_MPPT* [49,59], *FLC_MPPT* [52,54,59], *ANFIS_MPPT* [59,65,67], *FL_GA_MPPT* [58,59]. Il est à noter que les contrôleurs mentionnés ci-dessus ont été développés au niveau du *Laboratoire des Dispositifs de Communication et de Conversion Photovoltaïque (LDCCP/ENP)*.

Les différents scénarios de test sont élaborés sous environnement Matlab/Simulink [135].

4.2 MÉTHODOLOGIE DE SIMULATION DES DIFFÉRENTS CONTRÔLEURS MPPT

Afin de mener à bien l'étude comparative entre les différents contrôleurs et de mettre en avant les contrôleurs MPPT issus des méthodes bio inspirées, nous avons adopté la méthodologie citée précédemment.

Cette dernière consiste, en premier lieu, à modéliser le système PV, qui sera suivi par l'étape de paramétrage des algorithmes proposés, puis à sélectionner les différents contrôleurs MPPT et les insérer, un à un, dans le système PV modélisé sous Matlab/Simulink lors de chaque phase de simulation en tenant compte des critères sélectionnés et des différents scénarios considérés.

4.2.1 Modélisation du système photovoltaïque

La figure.4.1 illustre le schéma bloc développé pour la simulation du système photovoltaïque. Le modèle global de simulation du système PV est une combinaison de différents blocs modélisant les éléments qui le constituent.

Ce dernier est modélisé sous environnement Matlab/Simulink. Il est composé des blocs suivants : un générateur PV, un convertisseur statique de type Boost, un bloc de charge résistive et un bloc contrôleur MPPT permettant d'insérer après sélection, les différents contrôleurs à simuler.

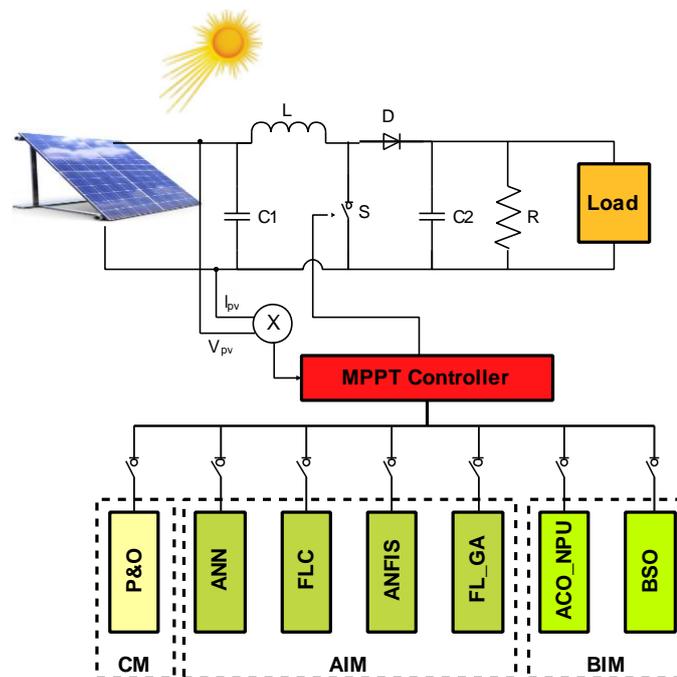


Figure.4.1 Schéma bloc pour la simulation du système PV.

4.2.2 Paramétrage des algorithmes *ACO_NPU* et *BSO*

La phase de paramétrage permettant la sélection des meilleurs paramètres des deux algorithmes est une étape cruciale dans le processus de conception et de mise en œuvre de nos deux contrôleurs.

En effet, les performances de ces derniers peuvent être influencées par le choix de paramètres (*taille de la population, coefficient constituant l'algorithme, nombre d'itération, etc...*) qui peuvent avoir un grand impact sur leur efficacité (i.e convergence lente, bloqué dans un maximum local, etc...).

A titre d'exemple, si la taille de la population (*fourmis/chauve-souris*) est grande, cela va élargir l'espace de recherche pour chaque itération. Cependant, cela va engendrer une augmentation du temps de calcul.

De plus, le nombre d'itération a , lui aussi, a une grande influence, à titre d'exemple : si le nombre d'itération est petit, cela peut bloquer le processus de recherche prématurément, tandis que pour un nombre élevé d'itérations, cela pourra augmenter le temps de calcul.

Pour ce faire, tenant compte de ces données et afin d'évaluer l'influence des paramètres lors de l'exécution des deux algorithmes *ACO_NPU* et *BSO* et de valider les deux contrôleurs *ACO_NPU_MPPT* et *BSO_MPPT* proposés, plusieurs tests sont effectués.

Ces derniers consistent à évaluer les contrôleurs en changeant les valeurs des paramètres respectifs à savoir : (*taille de l'archive, nombre de fourmis, nombre de chauves-souris, taux d'évaporation de la phéromone, amplitude, vitesse, fréquence, etc...*), permettant ainsi de sélectionner les plus appropriés de façon à réduire le temps d'exécution et d'améliorer l'efficacité des contrôleurs, i.e. atteindre le PPM.

Les différents tests sont effectués sous conditions standard (1000W/m², 25°C).

4.2.2.1 Paramétrage de l'algorithme ACO_NPU

La figure.4.2 illustre les résultats de simulations pour différentes valeurs de paramètres du contrôleur ACO NPU.

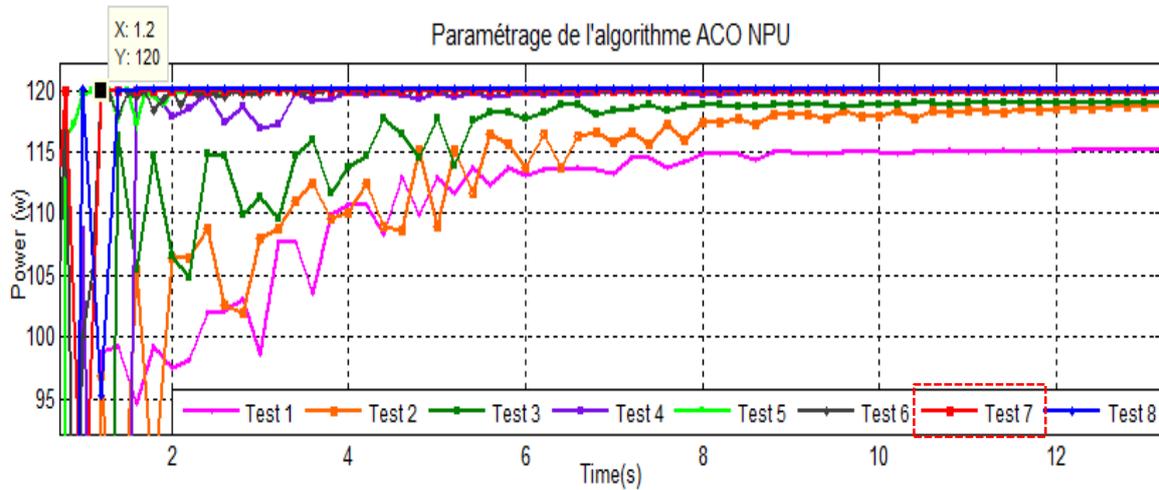


Figure.4.2 Simulation pour différentes valeurs de paramètres du contrôleur. ACO_NPU_MPPT.

Le tableau ci-dessous illustre les résultats de simulation pour différentes valeurs des paramètres :

- K* : la taille de l'archive ;
- m* : le nombre de fourmis ;
- α : la borne max de l'intervalle pour la génération de positions des fourmis ;
- p* : le taux d'évaporation de la phéromone.

Tableau.4.1 Résultats de simulation pour différentes valeurs des paramètres.

	T1	T2	T3	T4	T5	T6	T7	T8
Paramètres								
<i>K</i>	5	6	7	7	7	7	7	7
<i>m</i>	3	4	4	4	4	4	4	4
α	4	4	3.8	3.8	3.9	3.5	4	4
<i>p</i>	0.3	0.3	0.35	0.4	0.38	0.3	0.37	0.38
P_{max} (w)	115	118	119.74	119.89	119.99	120	120	120
t_{ps} (s)	8	7.7	7.4	3.4	2.7	3	1.2	1.3

Selon les différents tests effectués, l'utilisation d'une archive de taille 7 et un nombre de fourmis de 4 suffisent pour assurer la convergence du contrôleur ; ces valeurs sont les valeurs minimales pouvant être utilisées. En effet pour des valeurs inférieures à la taille de l'archive « *K* : la taille d'archive » et un nombre inférieur de fourmis « *m* : nombre de fourmis », la méthode peut converger vers un maximum local et ne pas atteindre la solution optimale comme montré par les premiers tests du tableau.4.1.

D'après les résultats du tableau (Tableau.4.1), le choix des paramètres « α » et « p » est très crucial. En augmentant la valeur du paramètre « alpha » on peut obtenir un temps d'exécution plus rapide mais on risque de trop osciller autour de la valeur MPP (la puissance maximale). Pour réduire le facteur d'oscillation on choisit un taux d'évaporation « p » permettant de diminuer la valeur du paramètre « alpha » à chaque itération de l'algorithme. Le bon compromis entre « α » et « p » est obtenu pour les valeurs 4.0 et 0.37 respectivement.

4.2.2.2 Paramétrage de l'algorithme BSO_NPU

La figure.4.3 illustre les résultats de simulations pour différentes valeurs de paramètres du contrôleur BSO_MPPT.

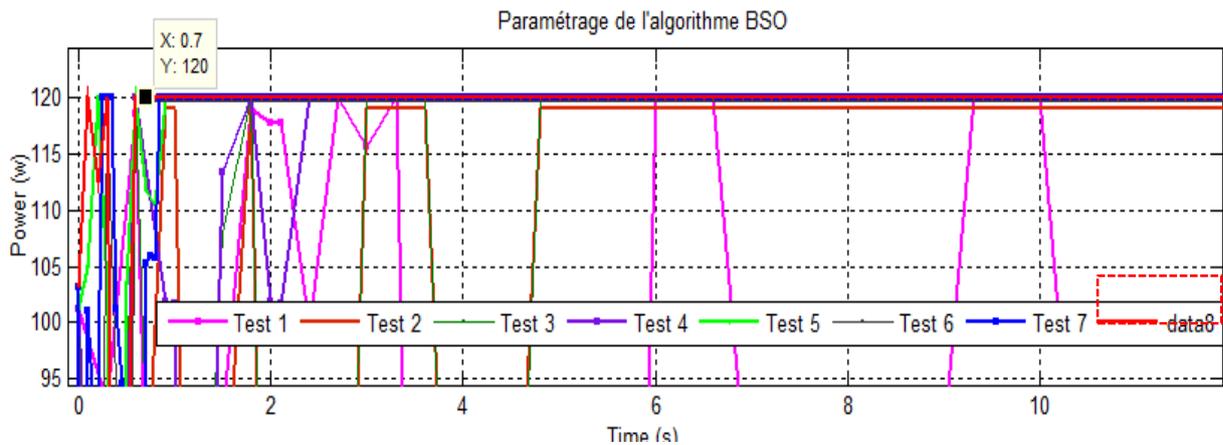


Figure.4.3 Simulation pour différentes valeurs de paramètres du contrôleur BSO_MPPT.

Le tableau ci-dessous illustre les résultats de simulation pour différentes valeurs des paramètres :

- K : la taille de l'archive ;
- Nb_bat : nombre de chauve souris ;
- A : amplitude d'impulsion émise « loudness » ;
- f_{max} : fréquence maximale d'émission ;
- γ : constante positive ;
- β : vecteur aléatoire $[0,1]$;
- r : taux d'émission de pulsation $[0,1]$;

Tableau.4.2 Résultats de simulation pour différentes valeurs des paramètres.

	T1	T2	T3	T4	T5	T6	T7	T8
Paramètres								
K	10	5	5	5	5	5	5	5
Nb_bat	10	5	5	3	3	3	3	3
$loudness$	0.01	0.01	0.02	0.02	0.1	0.2	0.2	0.2
$rate$	0.05	0.05	0.3	0.3	0.3	0.3	0.3	0.3
f_{max}	10	10	10	10	3	3	3	3
γ	0.001	0.001	0.7	0.7	0.9	0.8	0.9	0.8
β	0.001	0.001	0.7	0.7	0.9	0.8	0.8	0.8
$P_{max} (w)$	120	119	119.9	120	120	119.9	120	120
$t_{ps} (s)$	19	5	4.7	2.4	1	0.8	0.85	0.7

Selon les différents tests effectués, l'utilisation d'une archive de taille 5 et un nombre de chauve-souris de 3 suffisent pour assurer la convergence du contrôleur ; ces valeurs sont les valeurs minimales pouvant être utilisées.

En effet pour des valeurs supérieures de « K : la taille d'archive » et « m : nombre de chauve-souris », la méthode devient très gourmande en temps, peut converger vers un maximum local et ne pas atteindre la solution optimale, comme montré par le premier test du tableau.

4.2.3 Simulation des différents contrôleurs

Dans le but de visualiser la position du point de puissance maximale issu des différents contrôleurs, différents scénarios ont été élaborés afin de suivre l'évolution et le comportement de chacun des contrôleurs face aux différentes variations atmosphériques lors des différentes simulations.

L'étude comportementale des différents contrôleurs se fera sur la base des critères sélectionnés précédemment à savoir :

- *Convergence (temps de réponse,)* ;
- *Stabilité face aux oscillations* ;
- *Précision* ;
- *Robustesse*.

Afin de mettre en évidence chacun de ces critères, nous avons adopté les scénarios suivants :

- **Scénario 1** consiste à simuler le fonctionnement du système PV sous des conditions standard (25° , $1000\text{W}/\text{m}^2$) et visualiser le temps de réponse de chaque contrôleur permettant ainsi d'évaluer le critère relatif à la *convergence* des différents contrôleurs ;
- **Scénario 2** consiste à simuler le fonctionnement du système PV en modifiant les conditions atmosphériques (Irradiance et/ou de la Température) dans le but d'observer le comportement dynamique des différents contrôleurs MPPT permettant ainsi d'étudier les critères relatifs à la *stabilité*, *oscillation*, *précision* et *robustesse*. Les contrôleurs sont soumis aux scénarios suivants :
 - Simulations de fonctionnement du système sous des conditions variables d'ensoleillement où deux cas sont envisagés : diminution et augmentation de l'éclairement (lente et rapide) ;
 - Simulation de fonctionnement du système sous des conditions variables de la température.

Dans ce qui suit, nous allons appliquer chacun de ces scénarios pour la simulation des contrôleurs *ACO_NPU* et *BSO_MPPT* respectivement. Suite à cela, afin de justifier le choix de nos deux contrôleurs issus des méthodes BI, une étude comparative entre les méthodes conventionnelles et intelligentes en termes de convergence, stabilité face aux oscillations, précision et robustesse est établie.

Pour ce faire, nous avons sélectionné des contrôleurs issus des méthodes conventionnelles et intelligentes, à savoir : *P&O_MPPT* [52,58], *ANN_MPPT* [49,59], *FLC_MPPT* [52,54,59], *ANFIS_MPPT* [59,65,67], *FL_GA_MPPT* [58,59] contrôleurs.

4.2.3.1 Etude du critère de convergence

Comme mentionné précédemment, afin d’observer les temps de réponse de chaque contrôleur, le *scénario 1* est adopté. Il consiste à simuler les différents contrôleurs sous conditions standards ($S=1000W/m^2$, $T=25^{\circ}C$) et d’observer l’évolution de la puissance de sortie fournie à la charge de chaque contrôleur MPPT en fonction du temps.

Les figures.4.4 et .4.5 illustrent respectivement l’évolution de la puissance de sortie fournie à la charge pour les contrôleurs *ACO_NPU_MPPT* et *BSO_MPPT*.

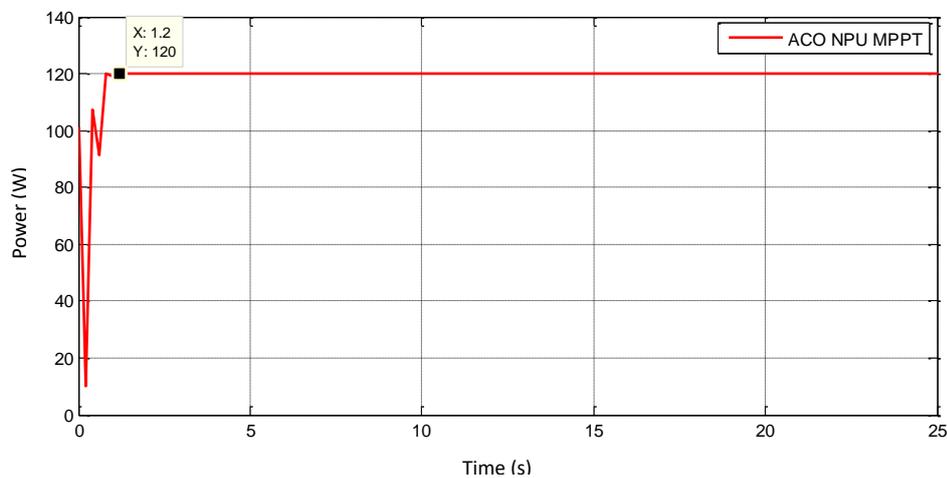


Figure.4.4 Evolution de la puissance de sortie du contrôleur *ACO_NPU_MPPT* sous conditions standards ($S=1000W/m^2$, $T=25^{\circ}C$)

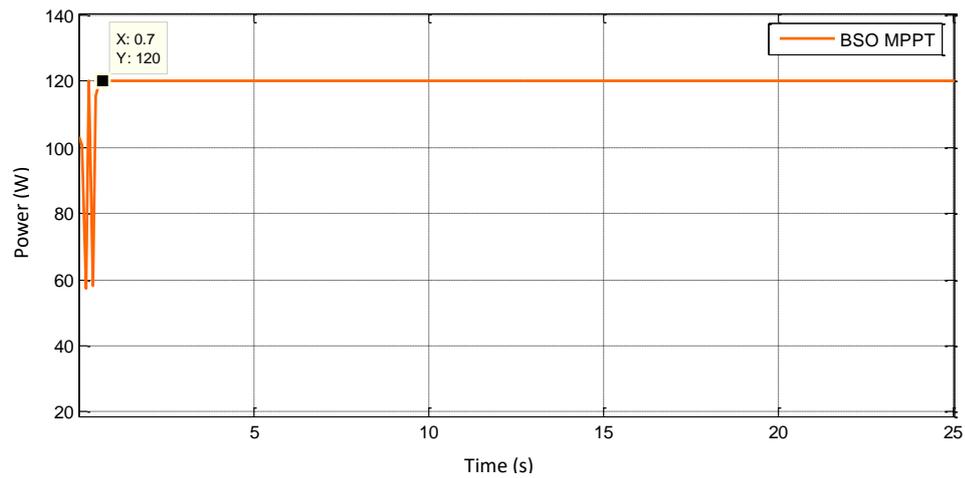


Figure.4.5 Evolution de la puissance de sortie du contrôleur BSO_ MPPT sous conditions standards ($S=1000W/m^2$, $T=25^{\circ}C$)

Les deux contrôleurs présentent un temps de réponse de $1.2s$ pour le contrôleur ACO_NPU_MPPT et de $0.7s$ pour le contrôleur BSO_MPPT.

Afin de mettre en évidence la progression des deux algorithmes ACO_NPU et BSO, les figures.4.6 et .4.7 illustrent respectivement les différentes étapes de ces derniers.

Comme illustré par les deux figures, la population (fourmis pour la figure.4.6. et chauve-souris pour figure.4.7.) est répartie aléatoirement dans le domaine de recherche selon la fitness, par la suite, au bout d'un certain nombre d'itérations, la population localise l'espace de recherche autour du PPM, vient alors l'étape de convergence ou le PPM est atteint.

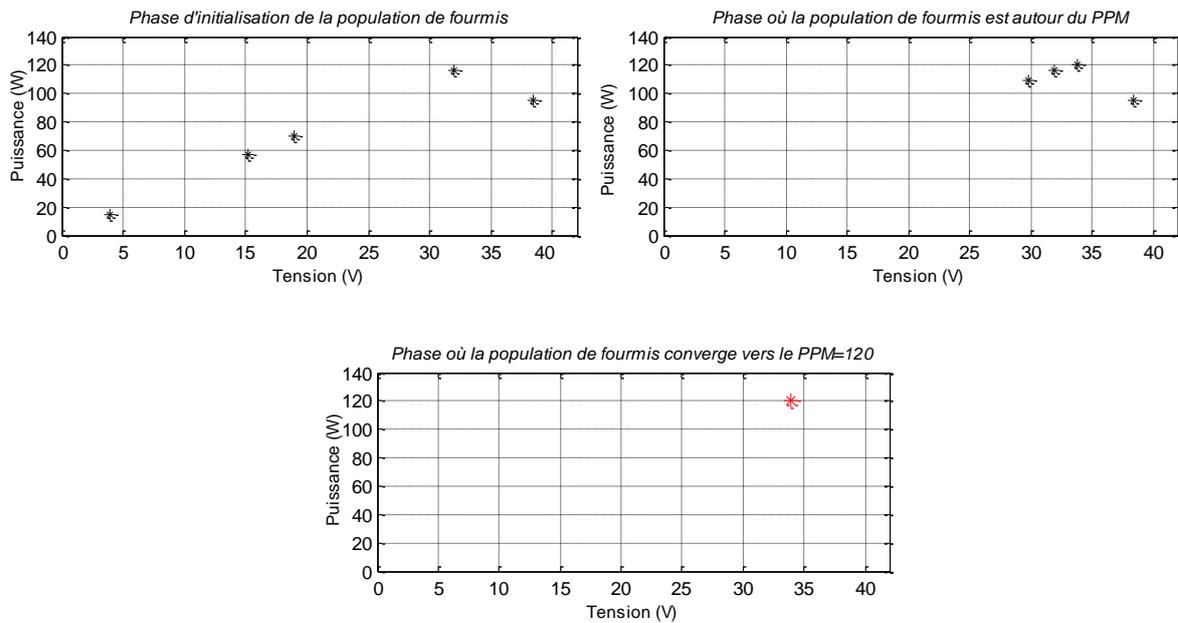


Figure.4.6 Différentes étapes du contrôleur ACO_NPU_MPPT pour atteindre le PPM.

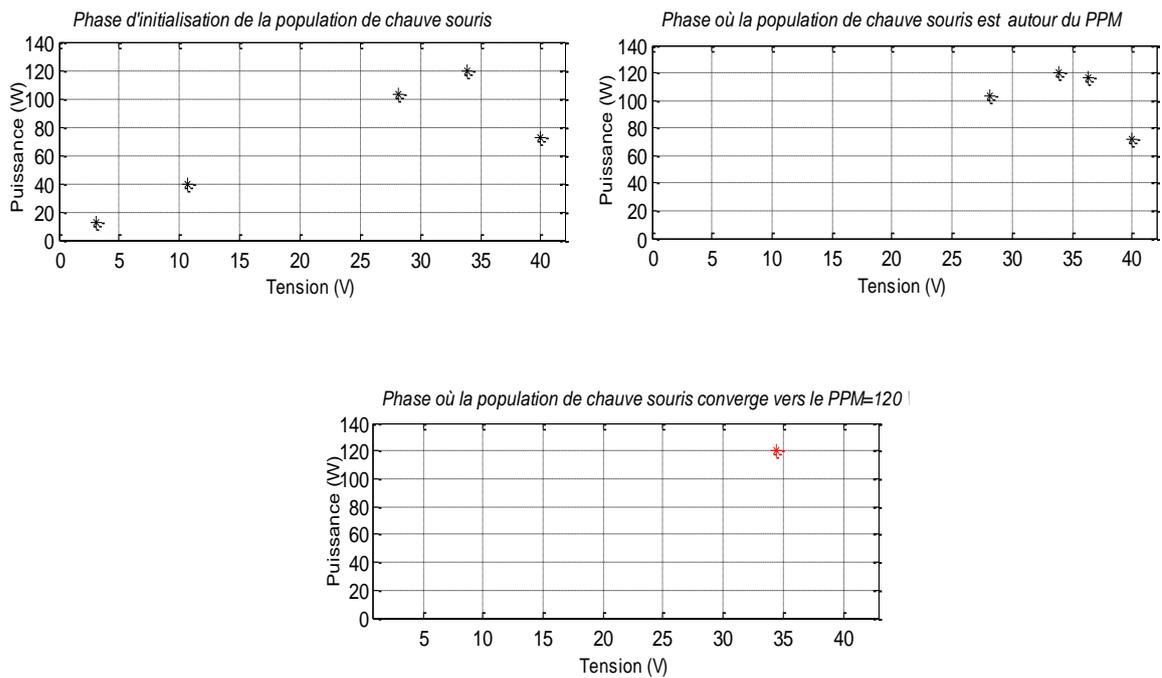


Figure.4.7 Différentes étapes du contrôleur BSO_MPPT pour atteindre le PPM.

La figure.4.8 illustre la réponse temporelle des contrôleurs ACO_NPU_MPPT, BSO_MPPT, GA_MPPT, ANFIS_MPPT, FL_MPPT, ANN_MPPT et P&O MPPT sous conditions environnementales standards. Les signaux de puissances sont superposés afin de pouvoir comparer leurs performances.

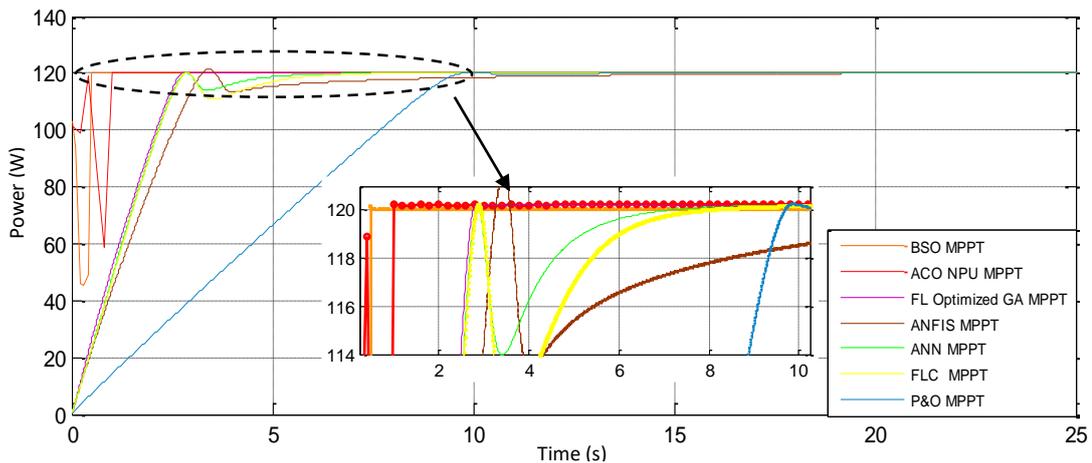


Figure.4.8 Evolution de la puissance de sortie des contrôleurs BSO, ACO_NPU, FL_GA, ANFIS, ANN, FLC and P&O MPPT sous conditions standards.

Discussion des résultats :

- Au vu des résultats obtenus lors des différentes simulations, nous constatons que le contrôleur P&O_MPPT appartenant à la catégorie des méthodes conventionnelles (CM) converge lentement vers le PPM avec un temps de réponse de 12.5s. De plus, ce dernier présente des oscillations autour du PPM qui ne s'éliminent pas causant ainsi une perte d'énergie ;
- Les contrôleurs issus des méthodes à base d'intelligence artificielle (AIM) à savoir : FLC-MPPT, ANN-MPPT, ANFIS-MPPT, GA-FL-MPPT convergent vers le PPM avec des temps de réponse respectifs de 9.16s, 8.09s, 6.03s et 3s. Les oscillations autour de ce point sont très faibles ;
- Pour ce qui est des contrôleurs proposés BSO_MPPT et ACO_NPU_MPPT issus des méthodes bio inspirées (BIM), ces derniers sont nettement plus rapide avec un temps de réponse de 0.7s pour BSO_MPPT et 1.2s pour ACO_MPPT. Les oscillations autour du PPM sont nulles.

La figure.4.9 illustre quant à elle le temps de réponse des différents contrôleurs.

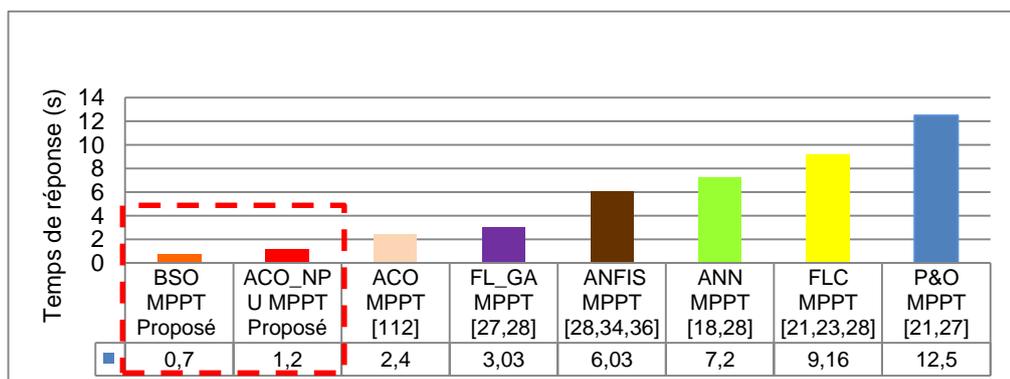


Figure.4.9 Temps de réponse des contrôleurs BSO, ACO_NPU, FL_GA, ANFIS, ANN, FLC et P&O MPPT.

Les contrôleurs *BSO_MPPT* et *ACO_NPU_MPPT* sont plus rapide que les contrôleurs issus des méthodes à base d'intelligence artificielle et conventionnelles. Les deux contrôleurs proposés ont donné de très bons résultats.

Comme conclusion, nous pouvons dire que les contrôleurs issus des méthodes à base d'intelligence artificielle ont apporté des améliorations en termes d'oscillations autour du PMM et de temps de réponse comparé aux méthodes conventionnelles. Mieux encore, les méthodes bio inspirées ont devancé les méthodes conventionnelles et les méthodes à base d'intelligence artificielle avec leur capacité d'adaptation par auto-organisation et leurs parallélisme en essence réduisant ainsi le problème d'oscillation autour du PPM et le temps de réponse de manière drastique.

4.2.3.2 Etude des critères de stabilité, de précision et de robustesse

L'efficacité des contrôleurs MPPT est évaluée par rapport à leur capacité à suivre le maximum de puissance, à leur robustesse, à leur stabilité face aux oscillations et à leur précision.

Pour estimer la précision, la stabilité, la robustesse et l'efficacité de chaque contrôleur, plusieurs scénarios ont été élaborés dans le but d'observer les réponses dynamiques des contrôleurs. Le système PV muni des différents contrôleurs est testé pour différentes variations des conditions environnementales (Température et/ou Eclairage). Nous présentons dans ce qui suit les résultats obtenus.

4.2.3.2.1 Augmentation rapide de l'éclairage

Au cours de ce test, une augmentation rapide de l'éclairage de 500 à 1000 W/m² est observée à l'instant t=20s. La température de la cellule est maintenue à 25°C pendant toute la durée du test.

Le figure.4.10 illustre respectivement l'évolution de la puissance de sortie fournie à la charge pour les contrôleurs *ACO_NPU_MPPT* et *BSO_MPPT*.

La figure.4.11 illustre l'évolution de la puissance de sortie fournie à la charge pour les contrôleurs *ACO_NPU_MPPT* et *BSO_MPPT* des méthodes Bio inspirées, ANFIS-MPPT, FLC-MPPT, ANN-MPPT, GA-FL-MPPT issus des méthodes à base d'intelligence artificielle et P&O des méthodes conventionnelles.

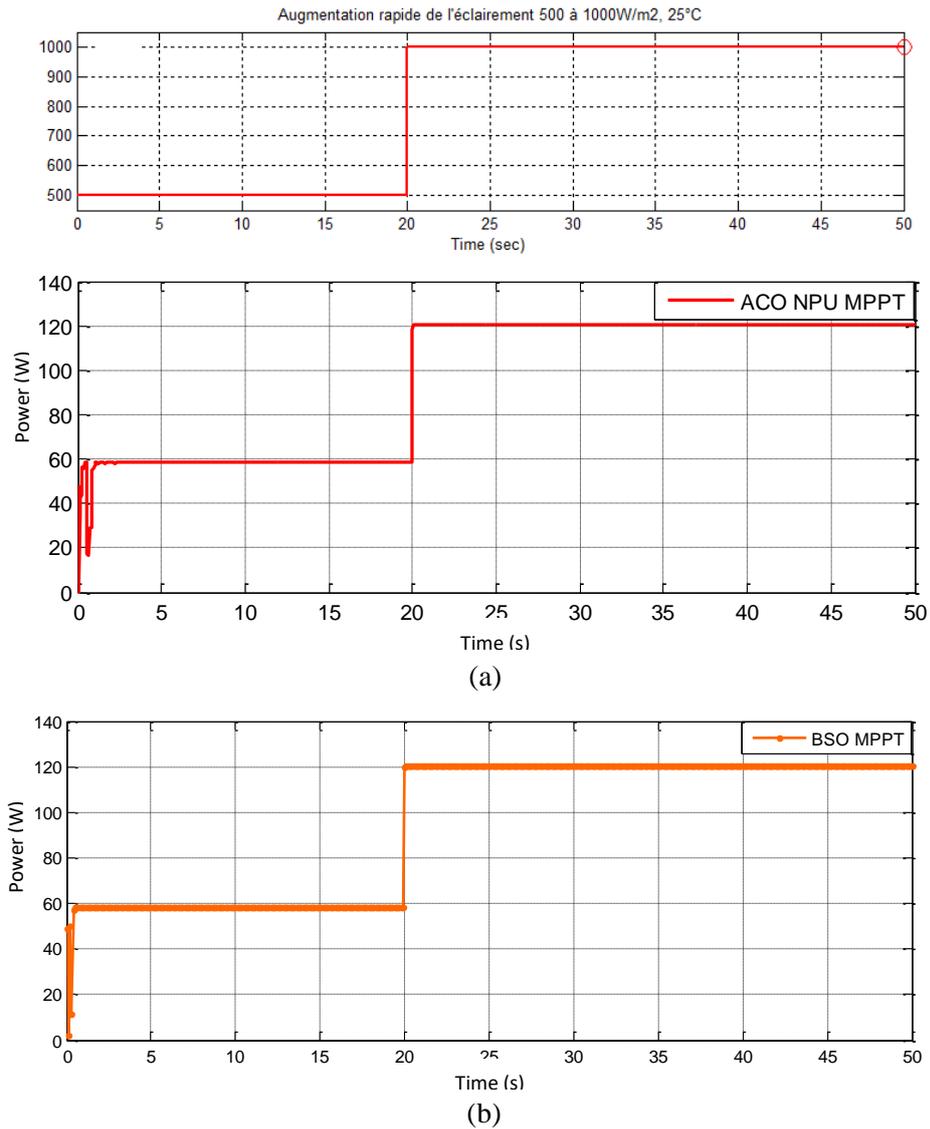


Figure.4.10 (a)-(b). Evolution de la puissance du contrôleur BSO, et *ACO_NPU* MPPT pour une augmentation rapide de l'éclairement de 500 à 1000W/m², 25°C.

A partir des résultats de simulation illustrés par les figures.4.10.(a) et (b), nous observons que lorsque le système PV est soumis à cet éclairement brusque de 500 à 1000W/m², une augmentation de la puissance générée par le PV se produit. A t=20s, les deux contrôleurs suivent parfaitement la variation brusque de l'éclairement de 500 à 1000W/m².

La figure.4.11 illustre une étude comparative de l'évolution de la puissance de sortie fournie à la charge pour les contrôleurs *ACO_NPU_MPPT*, *BSO_MPPT*, *GA_MPPT*, *ANFIS_MPPT*, *FL_MPPT*, *ANN_MPPT* et *P&O_MPPT* issus respectivement des méthodes bio inspirées, intelligentes et conventionnelles. Les signaux de puissances sont superposés afin de pouvoir comparer leurs performances.

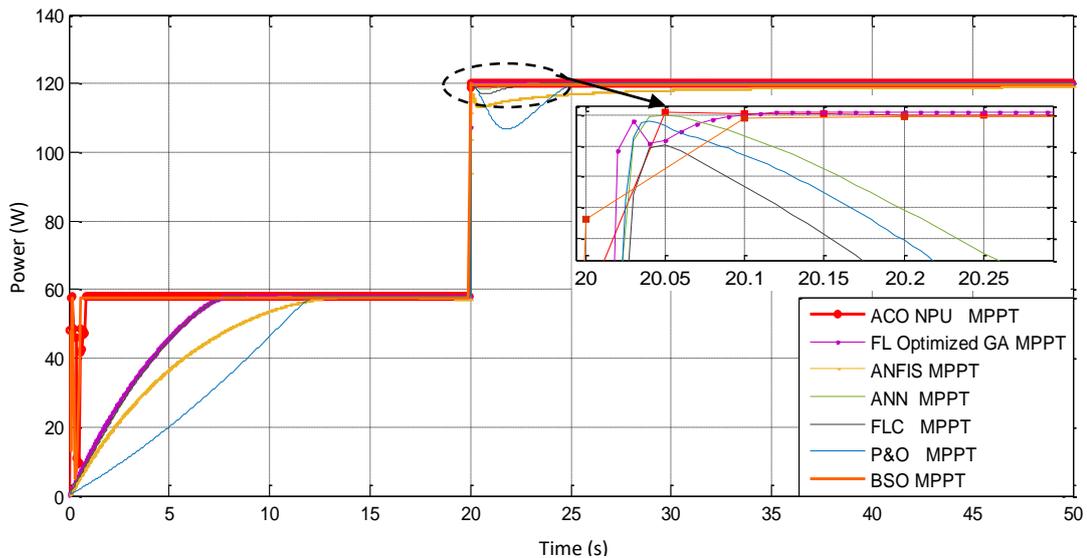


Figure.4.11 Evolution de la puissance du contrôleur *ACO_NPU_MPPT*, *BSO_MPPT*, *GA_MPPT*, *ANFIS_MPPT*, *FL_MPPT*, *ANN_MPPT* et *P&O_MPPT* pour une augmentation rapide de l'éclairement de 500 à 1000W/m², 25°C.

Discussion des résultats :

- A partir des résultats de simulation illustrés par la figure.11. nous observons que lorsque le système PV est soumis à un éclairement brusque de 500 à 1000W/m² à t=20s, une augmentation de la puissance générée par le PV se produit, tous les contrôleurs suivent la variation brusque de l'éclairement ;
- Le contrôleur *P&O_MPPT* appartenant à la catégorie des méthodes conventionnelles (CM) suit convenablement la variation à t=20s, mais les oscillations autour du PPM sont toujours présentes. Ces dernières ne se stabilisent pas, ce qui engendre des pertes de puissance ;
- Les contrôleurs issus des méthodes à base d'intelligence artificielle (AIM) à savoir, *ANFIS-MPPT*, *FLC-MPPT*, *ANN-MPPT*, *GA-FL-MPPT* suivent parfaitement la variation brusque à t=20s, cependant, ils présentent quelques oscillations, et mettent un temps avant de se stabiliser autour du PPM ;
- Concernant les contrôleurs proposés *BSO_MPPT* et *ACO_NPU_MPPT* issus des méthodes bio inspirées (BIM), ces derniers suivent le sens de la variation d'éclairement avec exactitude. Le PPM est atteint après seulement 0.05s du début de la variation brusque à t=20s pour le contrôleur *ACO_NPU_MPPT* et 0.1s pour le contrôleur *BSO_MPPT*. Ils ne présentent aucune oscillation autour du PPM et réagissent favorablement aux variations. Ils présentent une forte robustesse face à une augmentation rapide de l'éclairement.

4.2.3.2 Augmentation rapide de l'éclairage avec un temps transitoire

Au cours de ce test, une augmentation de l'éclairage de 500 à 1000 W/m² est observée à l'instant t=20s avec une période transitoire de 5s. La température de la cellule est maintenue à 25°C pendant toute la durée du test.

Le figure.4.12 illustre respectivement l'évolution de la puissance de sortie fournie à la charge pour les contrôleurs ACO_NPU_MPPT et BSO_MPPT respectivement

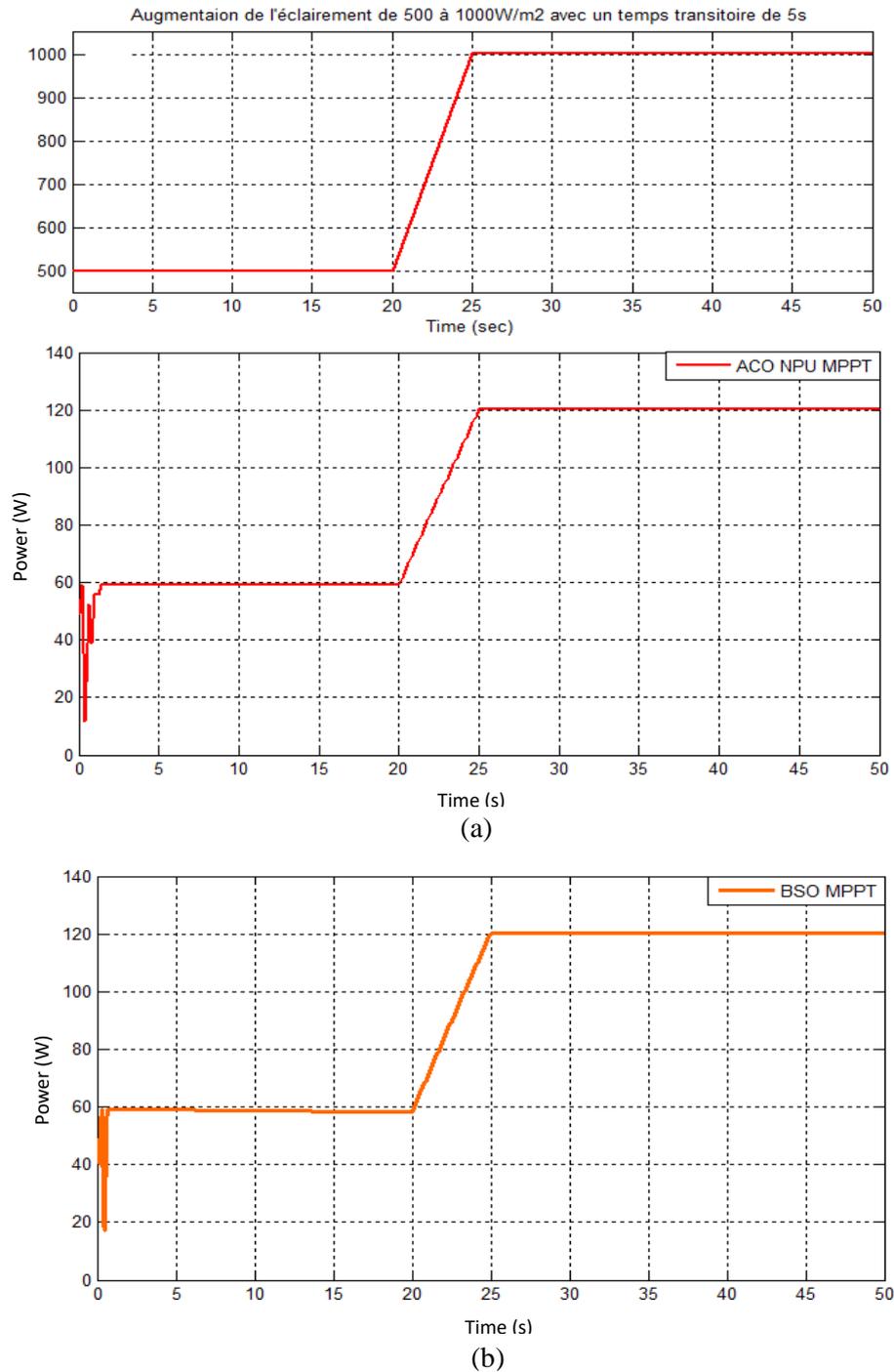


Figure.4.12 (a)-(b). Evolution de la puissance du contrôleur ACO_NPU et BSO MPPT pour une augmentation rapide de l'éclairage de 500 à 1000W/m², 25°C, avec une transition de 5s.

La figure.4.13 illustre une étude comparative de l'évolution de la puissance de sortie fournie à la charge pour les contrôleurs *ACO_NPU_MPPT*, *BSO_MPPT*, *GA_MPPT*, *ANFIS_MPPT*, *FL_MPPT*, *ANN_MPPT* et *P&O MPPT* issus respectivement des méthodes bio inspirées, intelligentes et conventionnelles.

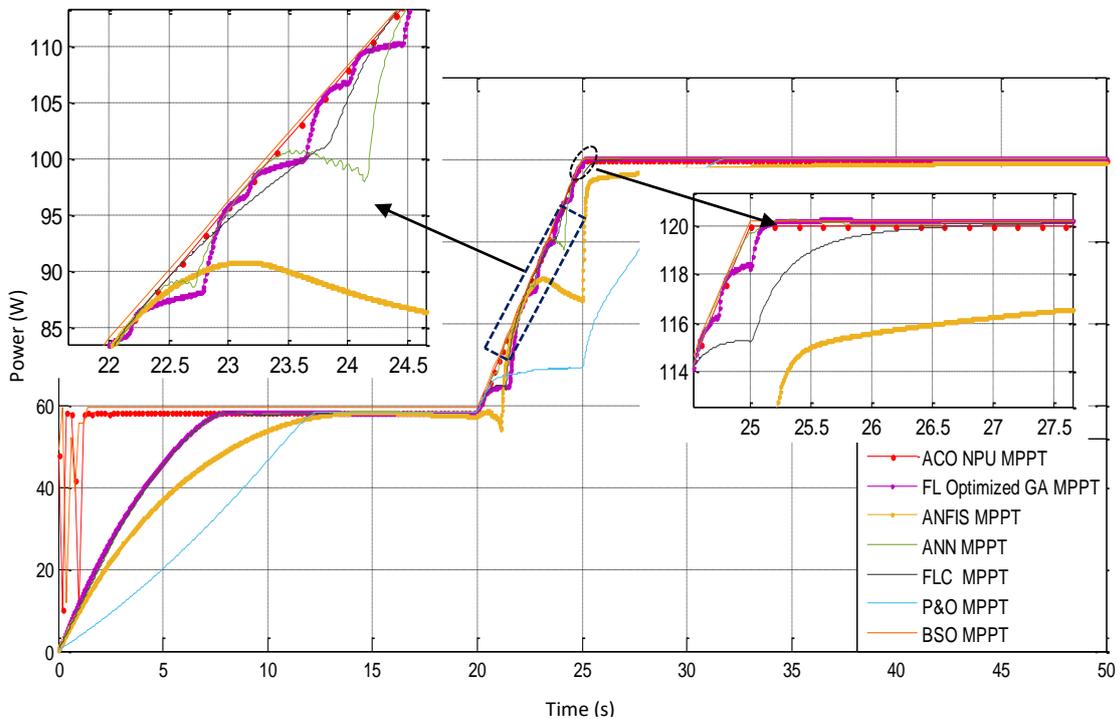


Figure.4.13 Evolution de la puissance du contrôleur *ACO_NPU_MPPT*, *BSO_MPPT*, *GA_MPPT*, *ANFIS_MPPT*, *FL_MPPT*, *ANN_MPPT* et *P&O MPPT* pour une augmentation rapide de l'éclairement de 500 à 1000W/m^2 , 25°C , avec une transition de 5s .

Discussion des résultats :

- Comme illustré dans la figure.4.13 le contrôleur *P&O_MPPT*, appartenant à la catégorie des méthodes conventionnelles (CM), réagit par rapport à cette augmentation comme si cette dernière est générée sous l'effet d'une précédente perturbation. Le fait que le contrôleur continue dans la même direction, il en résulte que le point de fonctionnement s'éloigne du vrai point de puissance maximale. Ainsi, le contrôleur ne revient au vrai PPM que lorsque l'éclairement se stabilise. Ceci induit un retard de réponse lors des changements soudains des conditions de fonctionnement et par voie de conséquence des pertes de puissance ;

- Les contrôleurs issus des méthodes à base d'intelligence artificielle (AIM) à savoir : ANFIS-MPPT, FLC-MPPT, ANN-MPPT et GA-FL-MPPT réagissent à la variation de l'éclairement ;
- Avant d'atteindre le PPM, le contrôleur ANFIS_MPPT présente quelques oscillations avant de se stabiliser autour de ce dernier. Cela génère une lenteur de convergence. Sur ce plan, les contrôleurs FLC_MPPT, ANN_MPPT et GA_FL_MPPT ont mieux réagi ;
- Concernant les contrôleurs proposés BSO_MPPT et ACO_NPU_MPPT issus des méthodes bio inspirées (BIM), ces derniers suivent le sens de la variation d'éclairement avec exactitude, comme illustré dans la figure.4.13.

Le PPM est atteint à $t=25s$ correspondant à la fin de la variation. Ils ne présentent aucune oscillation autour du PPM et réagissent favorablement aux variations.

Ceci nous amène à conclure que les deux contrôleurs présentent une forte robustesse face à une augmentation rapide de l'éclairement avec un temps transitoire.

4.2.3.2.3 Diminution rapide de l'éclairement

Au cours de ce test, l'éclairement est maintenu à 1000 W/m^2 de $t = 0$ à $t = 20s$, puis une diminution rapide de l'éclairement de 1000 W/m^2 à 500 W/m^2 se produit. La température de la cellule est maintenue à 25°C pendant toute la durée du test.

Les résultats de simulation sont illustrés respectivement dans la Figure.4.14 pour les contrôleurs ACO_NPU_MPPT et BSO_MPPT.

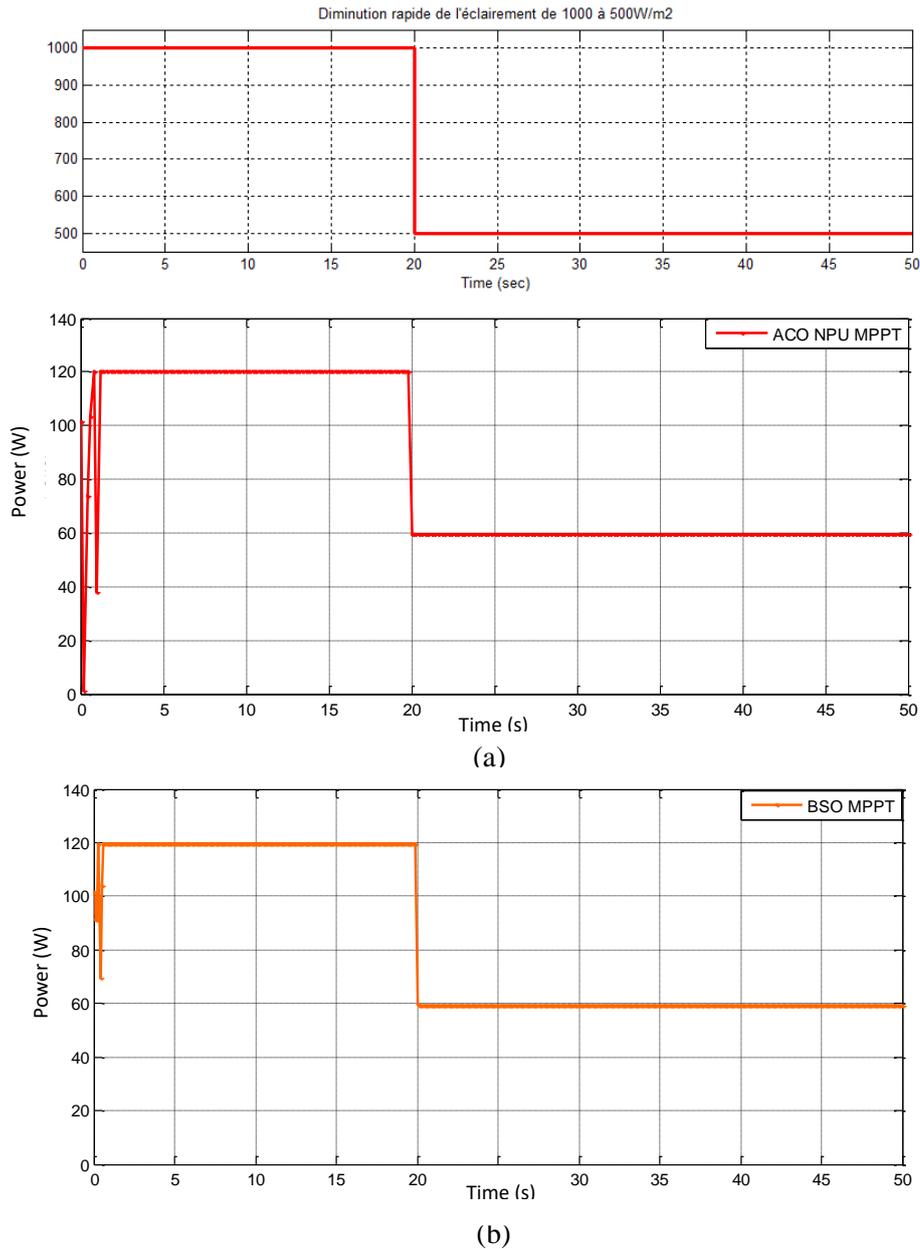


Figure.4.14 (a)-(b). Evolution de la puissance du contrôleur *ACO_NPU* et *BSO MPPT* pour une diminution rapide de l'éclairement de 1000 à 500W/m², 25°C.

La figure.4.15 illustre l'évolution de la puissance de sortie fournie à la charge pour les contrôleurs *ACO_NPU_MPPT*, *BSO_MPPT*, *GA_MPPT*, *ANFIS_MPPT*, *FL_MPPT*, *ANN_MPPT* et *P&O MPPT*. Les signaux de puissances sont superposés afin de pouvoir comparer leurs performances.

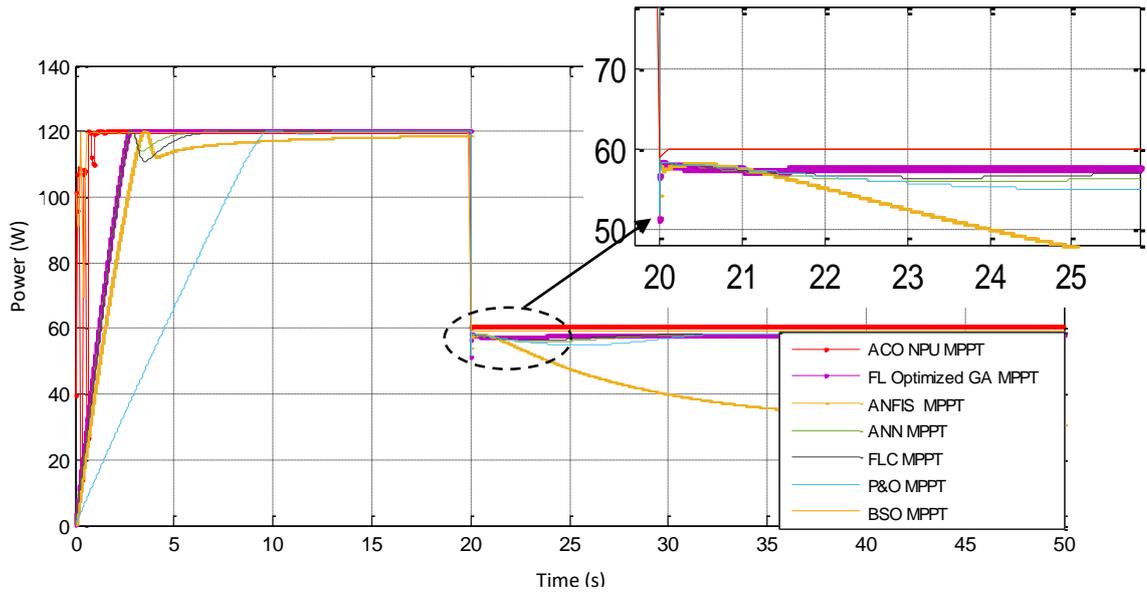


Figure.4.15 Evolution de la puissance du contrôleur *ACO_NPU_MPPT*, *BSO_MPPT*, *GA_MPPT*, *ANFIS_MPPT*, *FL_MPPT*, *ANN_MPPT* et *P&O_MPPT* pour une diminution rapide de l'éclairement de 1000 à 500W/m², 25°C.

Discussion des résultats :

Lorsque l'éclairement diminue à t =20s, les courbes P_V caractéristiques des différents contrôleurs sont quasi similaires lors de la variation brusque de l'éclairement.

- Cependant, le contrôleur *P&O_MPPT* appartenant à la catégorie des méthodes conventionnelles (CM) suit convenablement la variation de 1000 à 500W/m² à t=20s, mais les oscillations autour du PPM sont toujours présentes, ne se stabilisent pas ce qui engendre des pertes de puissance ;
- Concernant les contrôleurs issus des méthodes à base d'intelligence artificielle (AIM) à savoir : *ANFIS-MPPT*, *FLC-MPPT*, *ANN-MPPT*, *GA-FL-MPPT*, ils suivent parfaitement la variation à t =20s. Cependant, ces derniers présentent quelques oscillations avant de se stabiliser autour du PPM ;
- Les contrôleurs *BSO_MPPT* et *ACO_NPU_MPPT* proposés, issus des méthodes bio inspirées (BIM), suivent le sens de la variation d'éclairement à t=20s avec exactitude. Le PPM est atteint après seulement 0.2s de la fin de la variation à t=20s c'est-à-dire que le PPM est atteint à t=20.2s. Ils ne présentent aucune oscillation autour du PPM, et réagissent favorablement aux variations. Ceci nous permet de conclure qu'ils ont une forte robustesse face à une diminution rapide de l'éclairement.

4.2.3.2.4 Diminution rapide de l'éclairement avec un temps transitoire

Au cours de ce test, une diminution de l'éclairement de 1000 à 500 W/m² est observée à l'instant t=20s avec une période transitoire de 5s. La température de la cellule est maintenue à 25°C pendant toute la durée du test.

Le figure.4.16 illustre l'évolution de la puissance de sortie fournie à la charge pour les contrôleurs ACO_NPU_MPPT et BSO_MPPT respectivement.

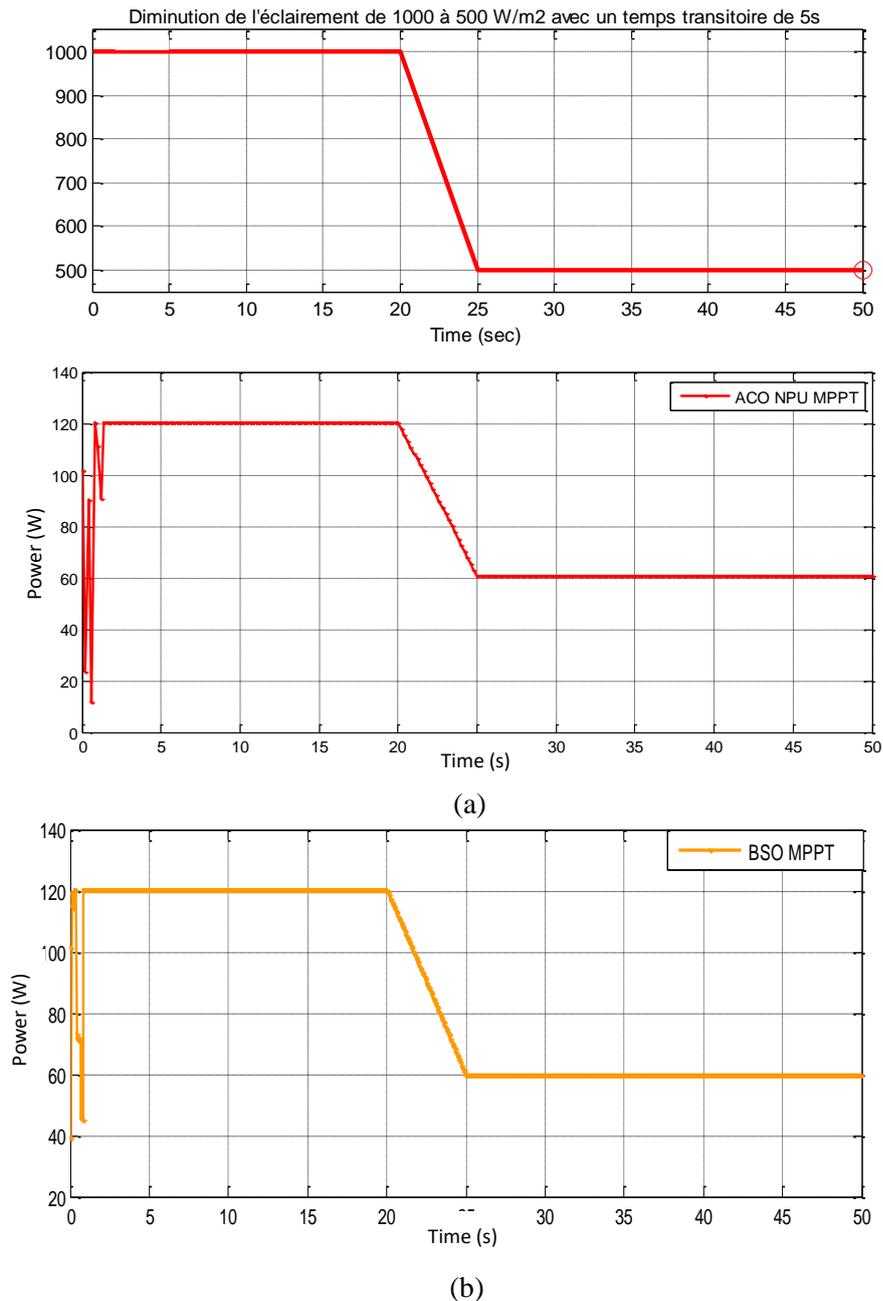


Figure.4.16 (a)-(b). Evolution de la puissance du contrôleur ACO_NPU et BSO MPPT pour une diminution rapide de l'éclairement de 1000 à 500W/m², 25°C, avec une transition de 5s.

La figure.4.17 illustre l'évolution de la puissance de sortie fournie à la charge pour les contrôleurs *ACO_NPU_MPPT*, *BSO_MPPT*, *GA_MPPT*, *ANFIS_MPPT*, *FL_MPPT*, *ANN_MPPT* et *P&O MPPT*. Les signaux de puissances sont superposés afin de pouvoir comparer leurs performances.

A partir des résultats de simulation illustrés par la figure.4.17. nous observons que lorsque le système PV est soumis à cet éclaircissement brusque de 1000 à 500W/m², une diminution de la puissance générée par le PV se produit.

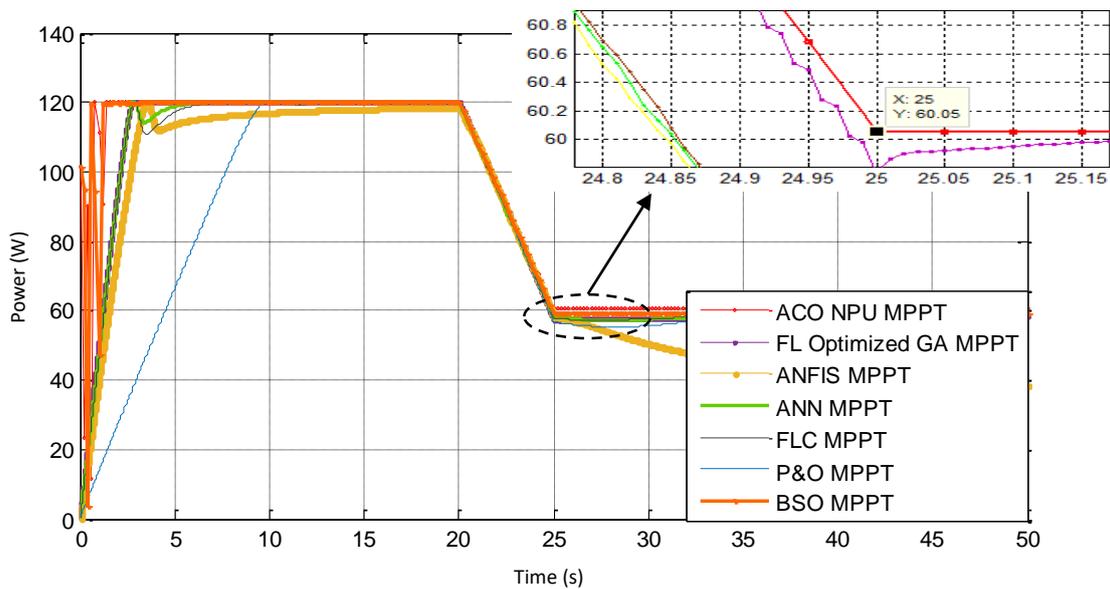


Figure.4.17 Evolution de la puissance du contrôleur *ACO_NPU_MPPT*, *BSO_MPPT*, *GA_MPPT*, *ANFIS_MPPT*, *FL_MPPT*, *ANN_MPPT* et *P&O MPPT* pour une diminution rapide de l'éclaircissement de 1000 à 500W/m², 25°C, avec une transition de 5s.

Discussion des résultats :

- Le contrôleur *P&O_MPPT* appartenant à la catégorie des méthodes conventionnelles (CM) suit convenablement la variation de 1000 à 500W/m² de t=20 à 25s, cependant, les oscillations autour du PPM sont toujours présentes, ne se stabilisent pas ce qui engendre des pertes de puissance ;
- Concernant les contrôleurs issus des méthodes à base d'intelligence artificielle (AIM) à savoir, *ANFIS-MPPT*, *FLC-MPPT*, *ANN-MPPT*, *GA-FL-MPPT*, ils suivent parfaitement la variation de t=20 à t=25s. Malgré cela, ces derniers présentent de petites oscillations avant de se stabiliser autour du PPM ;

- Les contrôleurs BSO_MPPT et ACO_NPU_MPPT que nous proposons, issus des méthodes bio inspirées (BIM), suivent le sens de la variation de l'éclairement de $t=20$ à $t=25$ s avec exactitude. Le PPM est atteint après seulement 25s correspondant à la fin de la variation.

Ils ne présentent aucune oscillation autour du PPM et réagissent favorablement aux variations. Ceci nous amène à conclure que les contrôleurs que nous proposons présentent une forte robustesse face à une diminution rapide de l'éclairement avec un temps transitoire.

4.2.3.2.5 Diminution rapide de la température

La température de la cellule est un facteur important qui affecte la courbe caractéristique du système photovoltaïque. Étant donné que la température au cours d'une journée ne change pas rapidement, ce test est effectué uniquement pour évaluer la performance des contrôleurs.

Au cours de ce test, l'éclairement est maintenu à 1000 W/m^2 et une diminution rapide de la température de 50° à 25°C est observée pendant 5s de $t=20$ à $t=25$ s.

La Figure.4.18 illustre les résultats de simulation.

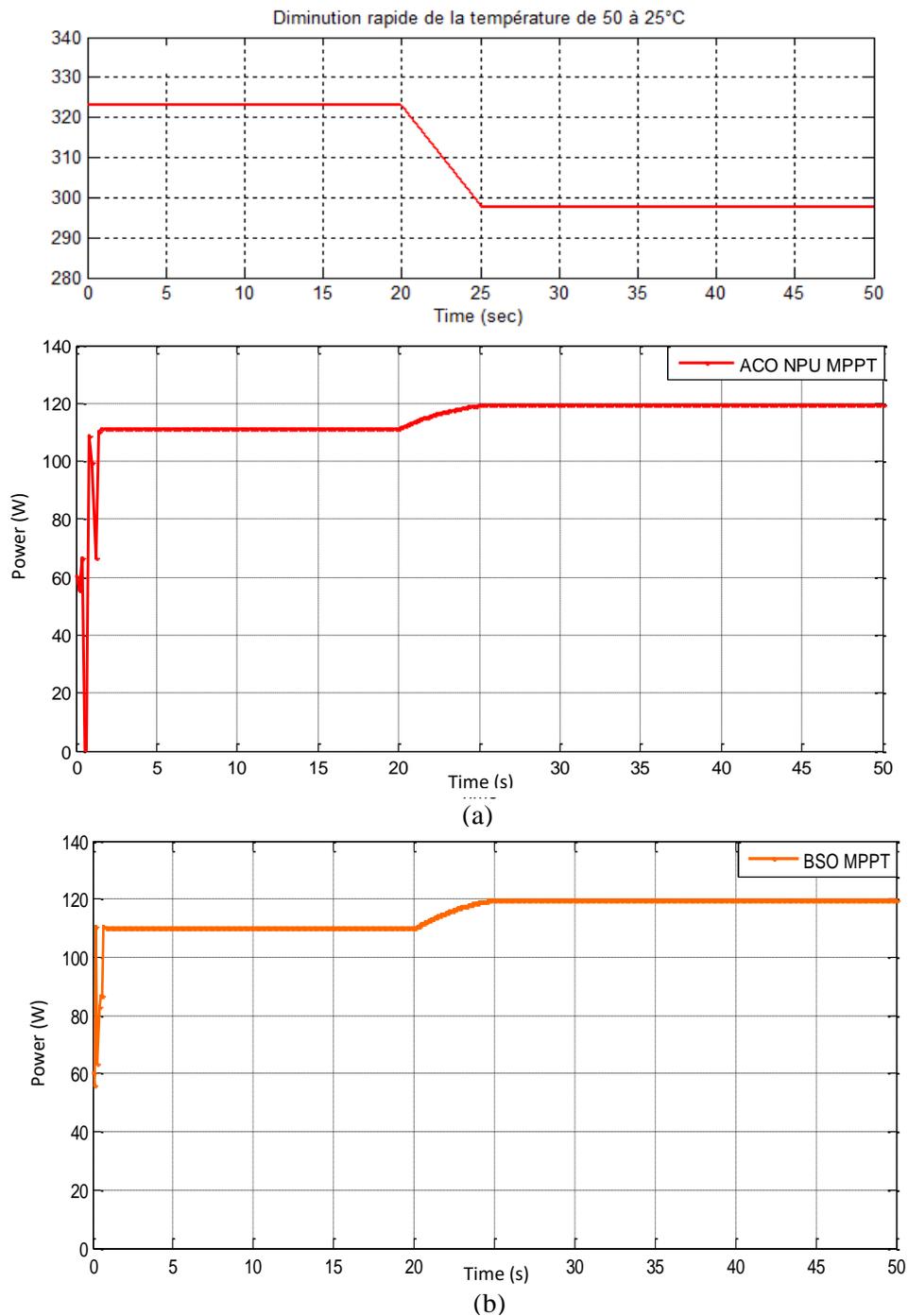


Figure.4.18 (a)-(b). Evolution de la puissance du contrôleur *ACO_NPU* et *BSO MPPT* pour une diminution rapide de la température de 50 à 25°C, 1000W/m² avec une transition de 5s.

La figure.4.19 illustre l'évolution de la puissance de sortie fournie à la charge pour les contrôleurs *ACO_NPU_MPPT*, *BSO_MPPT*, *GA_MPPT*, *ANFIS_MPPT*, *FL_MPPT*, *ANN_MPPT* et *P&O MPPT*. Les signaux de puissances sont superposés afin de pouvoir comparer leurs performances.

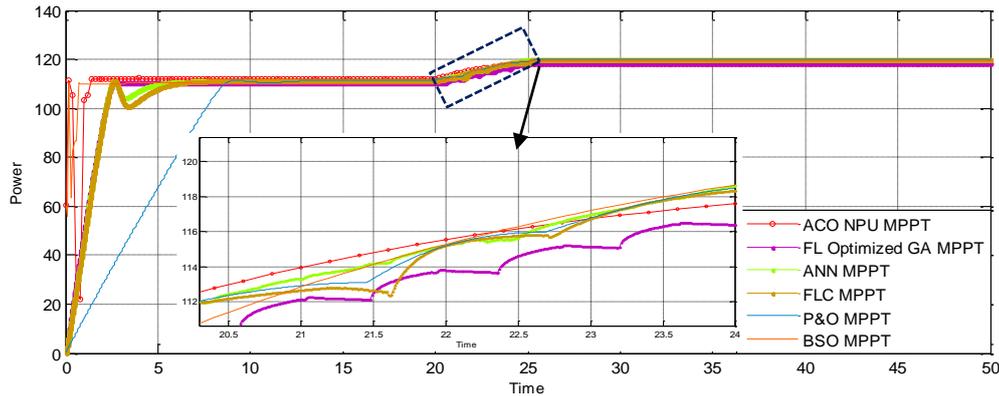


Figure.4.19 Evolution de la puissance du contrôleur *ACO_NPU_MPPT*, *BSO_MPPT*, *GA_MPPT*, *ANFIS_MPPT*, *FL_MPPT*, *ANN_MPPT* et *P&O MPPT* pour une diminution rapide de la température de 50 à 25°C, avec une transition de 5s.

Discussion des résultats :

- Lorsque la température baisse lors de la phase transitoire, tous les contrôleurs issus des différentes méthodes détectent la variation de température et suivent le PPM pour se stabiliser à la fin de la variation ;
- Cependant, comme illustré sur la figure.4.19.les contrôleurs issus des méthodes conventionnelles et de l'intelligence artificielle présentent des oscillations lors de la phase transitoire comparée aux méthodes bio inspirées qui réagissent parfaitement à la variation de la température.

4.2.3.2.6 Effet de l'ombrage partiel

Ce scénario a pour but d'observer les performances des méthodes bio inspirées face à l'effet de l'ombrage partiel sur le système PV et d'évaluer les performances de ces dernières.

Comme l'ombrage partiel est un phénomène stochastique, plusieurs scénarios peuvent se produire. Par conséquent, il est difficile de tester toutes les occurrences de l'ombre partiel.

De ce fait, nous avons opté pour la conception d'un modèle sous Simulink permettant de simuler le comportement du système PV. Ce dernier est composé de quatre modules PV. A une température fixée à la valeur 25°C, les quatre modules sont soumis à des éclaircements constants mais différents en terme de conditions d'ombrage partiel, comme énumérées dans le tableau ci-après :

Tableau.4.3 Différents scénarios d’ombrage.

	PV Panel 1 (W/m ²)	PV Panel 2 (W/m ²)	PV Panel 3 (W/m ²)	PV Panel 4 W/m ²)
Scénario d’ombrage 1 SP1	800	500	300	100
Scénario d’ombrage 2 SP2	800	200	200	800
Scénario d’ombrage 3 SP3	1000	800	700	600

Afin d’évaluer les performances des MBI, nous avons sélectionné le contrôleur ACO_NPU_MPPT pour effectuer les différents tests.

La figure ci-dessous illustre respectivement l’influence de l’ensoleillement non homogène sur les caractéristiques P-V du système PV pour différents patterns.

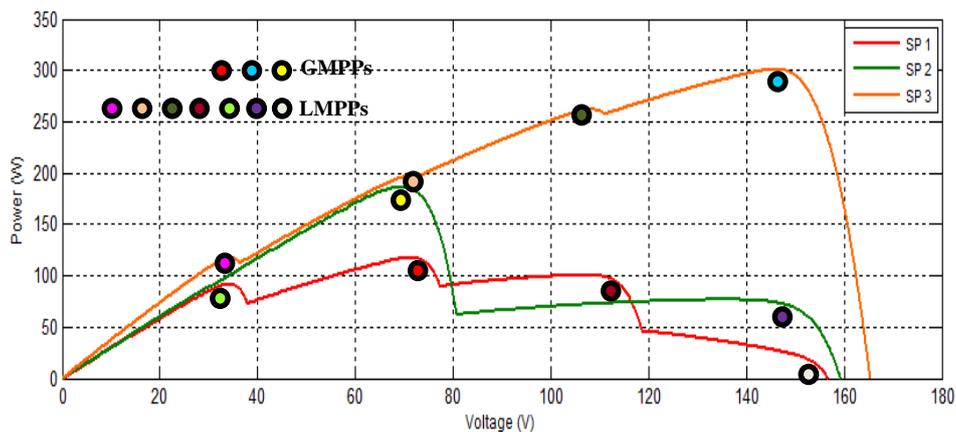


Figure.4.20 Caractéristique P-V pour différents pattern d’ombrage.

Comme l’illustre la figure.4.20, pour le SP1 et SP3, la chaîne reçoit quatre différentes insulations ; la courbe P_V présente quatre pics, alors que pour le SP2, elle reçoit deux pics.

La figure.4.21 illustre les résultats de simulation de la puissance extraite du contrôleur bio inspiré ACO_NPU_MPPT proposé.

Discussion des résultats :

- A partir des résultats de simulation obtenus, nous pouvons conclure que lorsque le phénomène d’ombrage se produit, le contrôleur proposé présente de bonnes performances pour atteindre le PPM sous différentes conditions d’ombrage partiel avec zéro oscillation à l’état d’équilibre ;

- Pour chaque pattern d'ombrage, le maximum global est atteint comme l'illustre la figure.4.21 correspondant aux différents cas d'ombrage partiel.

De ce fait, nous pouvons conclure, de par leur robustesse et leur aptitude d'auto adaptation, les méthodes Bio inspirées sont très efficace pour traiter le problème de l'ombrage partiel.

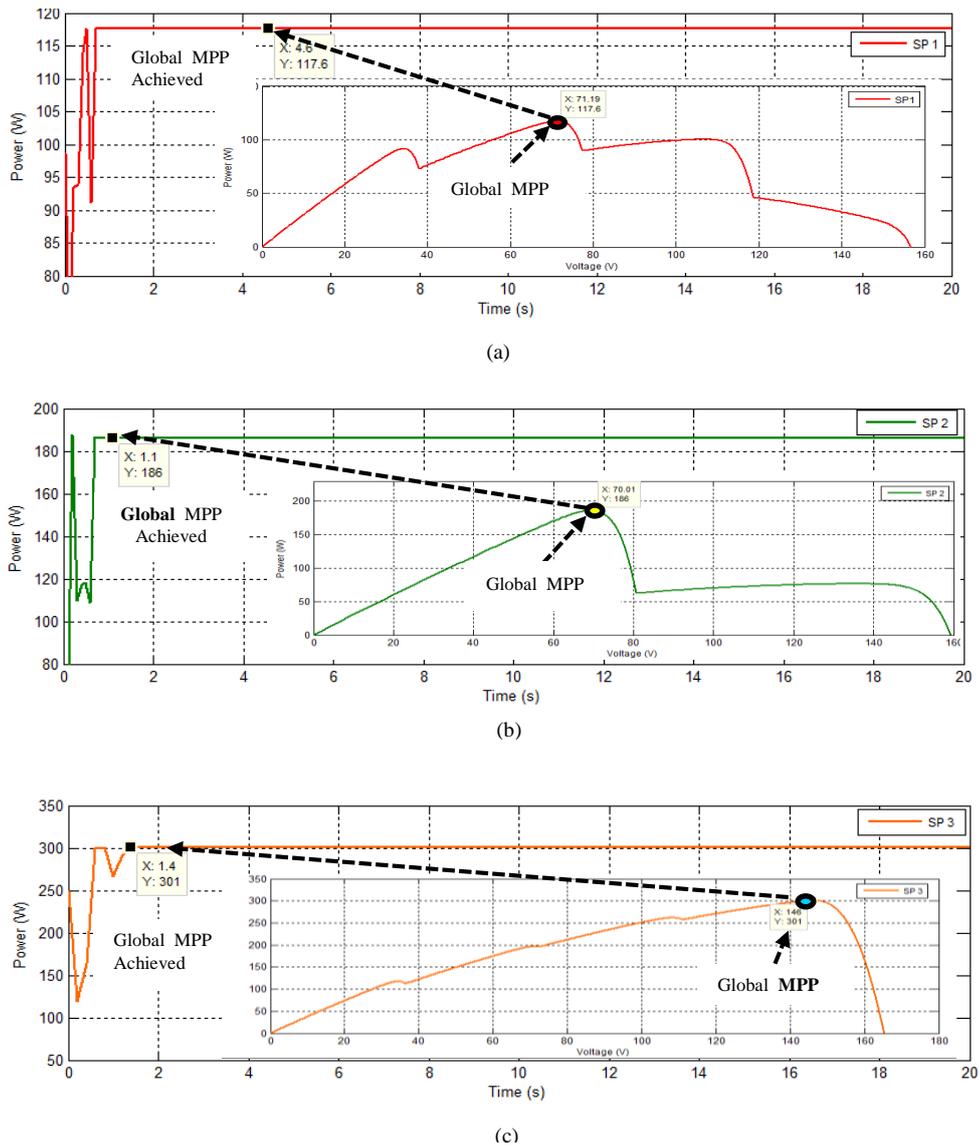


Figure.4.21 The extracted power during the selected shading patterns with the corresponding P_V characteristic curve, (a) SP1, (b) SP2, (c) SP3.

4.3 SYNTHÈSE DES DIFFÉRENTS CONTRÔLEURS MPPT

Le **Tableau.4.4** résume les caractéristiques des différents contrôleurs MPPT obtenus après simulation.

Il donne un aperçu de l'efficacité des différents contrôleurs pour les différents tests effectués (conditions standard et la variation des conditions atmosphériques). Nous avons adopté les critères de stabilité, vitesse de convergence, robustesse et complexité d'implémentation.

Il est clair que la liste des critères de performance n'est guère exhaustive. Ce choix volontaire se veut comme une contribution à une réflexion plus générale visant à mettre au point un contrôleur MPPT qui soit le plus performant possible dans une large plage de fonctionnement. L'estimation des performances et des limites des commandes MPPT ne pourrait se faire qu'en se basant sur ces critères.

Sur la base des résultats obtenus, il est prouvé que les contrôleurs MPPT proposés à savoir : ACO_NPU_MPPT et BSO_MPPT, appartenant aux méthodes bio inspirées, sont les plus performants sous toutes les conditions de fonctionnement.

Ils montrent un niveau élevé de *robustesse* face aux variations atmosphériques (Eclairage et/ou Température, Ombrage), permettant une adaptation dynamique face aux changements du fait que les contrôleurs proposés sont conçus à base de populations (fourmis, chauve-souris), définissant des interactions simples au niveau local et permettant l'émergence d'un comportement complexe au niveau global.

Tableau.4.4 Caractéristiques des différents contrôleurs MPPT

Contrôleurs MPPT		Efficacité η (%)	Stabilité (oscillation)	Convergence	Robustesse	Complexité d'implémentation
MBI	ACO_NPU_MPPT	Très haute	Zero	Très rapide	Très haute	Moyenne
	BSO_MPPT	Très haute	Zero	Très rapide	Très haute	Moyenne
	ACO_MPPT [112]	Haute	Négligeable	Rapide	Très haute	Moyenne
MIA	ANN_MPPT	Haute	Négligeable	Rapide	Haute	Haute
	FLC_MPPT	Haute	Négligeable	Rapide	Haute	Haute
	FL_GA_MPPT	Haute	Négligeable	Rapide	Haute	Haute
	ANFIS_MPPT	Haute	Négligeable	Rapide	Haute	Haute
MC	P&O_MPPT	Faible	Négligeable	Lent	Lent	Simple

4.4 CONCLUSION

Les simulations décrites dans ce chapitre sont purement théoriques car pour évaluer les performances d'une commande numérique avant d'établir un prototype réel, il est nécessaire d'effectuer d'abord des simulations.

Ces dernières demeurent, à l'heure actuelle, la meilleure approche pour évaluer les diverses commandes MPPT et d'analyser leurs comportements sous les mêmes conditions de fonctionnement. Pour cela, il a fallu prendre en considération plusieurs critères pour les évaluer.

Les résultats obtenus des simulations ont démontré les différences existantes dans l'efficacité des contrôleurs MPPT. Pour cela plusieurs scénarios ont été élaborés pour mettre les contrôleurs à l'épreuve des variations atmosphériques : éclairement et/ ou température. Par ailleurs, le problème tout aussi réaliste de l'ombrage partiel a été traité. Ce qui a donné naissance à un modèle permettant la simulation de ce phénomène. En effet, dans ce cas précis, plusieurs maxima locaux peuvent surgir ce qui induit le contrôleur en erreur et impliquant ainsi une perte de puissance non négligeable.

Le fait de comparer les différentes commandes MPPT, nous permet de sélectionner celle qui soit la plus appropriée pour une application déterminée. Cela nous permet aussi de prévoir le comportement réel et les performances de chaque commande MPPT.

En conclusion, nous pouvons dire que les contrôleurs issus des méthodes Bio inspirées sont les plus robustes, stables et précis.

Dans le prochain chapitre, il sera question de la proposition d'une nouvelle plateforme software pour le prototypage rapide des contrôleurs MPPT.

Chapitre 5

**PROPOSITION D'UNE NOUVELLE
PLATEFORME POUR LE PROTOTYPAGE
RAPIDE DES CONTRÔLEURS MPPT**

5.1 INTRODUCTION

Au cours de ce chapitre, nous proposons une nouvelle plateforme baptisée **PVSP** pour « *PhotoVoltaic Simulation Platform* », développée pour le prototypage rapide des contrôleurs MPPT. Cette dernière, intègre, en un seul outil, toutes les phases nécessaires pour le développement et la simulation d'un contrôleur MPPT soumis à diverses contraintes environnementales (éclairage, température, ombrage partiel...).

Afin de situer notre contribution, un état de l'art sur les plateformes d'énergies renouvelables (EnR) existantes sera donné.

Suite à cela, une présentation de la plateforme **PVSP**, de son architecture, ainsi que son fonctionnement, permettant l'implémentation et le test des contrôleurs MPPT, sera exposée.

5.2 ÉTAT DE L'ART SUR LES PLATEFORMES POUR LES EnR

La littérature fait référence à un certain nombre de plateformes pour les EnR, développées par différentes institutions, permettant de réaliser des dimensionnements, des simulations et l'analyse des systèmes d'énergies renouvelables. La majorité de ces plateformes de simulation disponibles sur le marché ne sont pas gratuites, de plus, elles ne sont pas destinées à l'évaluation des performances des systèmes PV [136]. Pour exemple :

 PV DesignPro	La plateforme PV Design Pro [137] développée par <i>Maui Solar Energy Software Corporation</i> , est une plateforme permettant de simuler le fonctionnement des systèmes photovoltaïques. Ils peuvent être dans des sites isolé (avec stockage) ou connectés au réseau électrique (sans stockage) ;
 PVsyst	La plateforme PVsyst [138], développée par <i>PVsyst SA</i> , est une plateforme permettant de simuler le fonctionnement des systèmes photovoltaïques ainsi que le dimensionnement de ces systèmes ;
 archelios TM CALC	La plateforme ARCHELIOS CALC [139] permet de concevoir, simuler dimensionner et optimiser une installation photovoltaïque raccordée au réseau.
 HOMER	La plateforme Homer [140], développée par <i>HOMER Energy LLC</i> , est une plateforme dédiée au design et à l'étude des systèmes énergétique contenant un mix de productions conventionnelles et renouvelables, utilisée tout aussi bien pour les applications en sites connectés au réseau, ou isolés.
 energyPro	La plateforme energyPRO [141], développée par <i>EMD International</i> , est une plateforme dédiée au design, à l'analyse et à l'optimisation de systèmes énergétiques complexes.

La liste des plateformes ci-dessus n'est pas exhaustive. Nous n'avons pu toutes les énumérer, faute de les avoir toutes identifiées.

Cependant, les plateformes existantes tiennent compte plus des aspects économiques liés au design, au dimensionnement, à la simulation des systèmes hybrides, ...etc. A l'heure actuelle, il n'existe pas de consensus concernant le développement de plateformes qui prennent en considérations les aspects techniques liés aux tests et à la simulation des différents blocs constituant un système photovoltaïque, tel que les contrôleurs MPPT, les onduleurs, les hacheurs, ... etc.

De ce fait, le besoin de concevoir une telle plateforme est motivé par le fait que la plupart des tests sur les contrôleurs MPPT proposés sont généralement effectués dans des environnements et avec des outils différents, ce qui rend la comparaison entre ces contrôleurs difficile et complexe. De plus, cela n'offre pas à l'utilisateur/concepteur le choix entre plusieurs contrôleurs MPPT, ou la possibilité d'insérer de nouveaux.

C'est dans cette optique, que nous avons jugé utile de proposer une première version d'une plateforme expérimentale didactique « *Photovoltaic Simulation Platform (PVSP)* ». Par son caractère modulaire, cette plateforme peut accueillir d'autres composants, qui, une fois validés (développement et tests) pourront être stockés dans la bibliothèque. En effet, cette dernière permet à l'utilisateur de tester les contrôleurs MPPT, sous différentes conditions atmosphériques (Eclairage/Température/Ombre), pour différents modèles de panneaux photovoltaïques et enfin de les comparer à d'autres contrôleurs MPPT, dans le but de sélectionner le meilleur, optimisant ainsi la gestion de puissance des systèmes photovoltaïques.

De plus, la plateforme développée offre ainsi une double fonctionnalité : elle constitue, d'une part, un outil de développement et de prototypage rapide de contrôleur MPPT pour le test et la validation des systèmes PV ; et d'autre part, elle servira comme outil pédagogique déployé au niveau des universités afin de vulgariser l'enseignement des énergies renouvelables et plus particulièrement le solaire photovoltaïque.

Dans les prochaines sections, une présentation de ladite plateforme sera donnée.

5.3 PRÉSENTATION DE LA PLATEFORME *PVSP*

La plateforme *PVSP* propose, par sa structure, un cadre méthodologique de développement, de simulation et de test de contrôleurs MPPT pour l'optimisation de puissance des systèmes PV (Figure.5.1).

Elle permet à l'utilisateur de fixer un certain nombre de paramètres de configuration à savoir : le *modèle du panneau photovoltaïque*, l'*architecture du générateur photovoltaïque* (nombre de panneaux série-parallèle), les données relatives aux *conditions climatiques* de fonctionnement (température, irradiation, ombrage) ainsi que le *choix des paramètres* de l'algorithme constituant le contrôleur MPPT.

Aussi, elle offre à l'utilisateur le choix entre le lancement d'une simulation de manière continue, d'une exécution pas à pas, de la mise en pause ou de la reprise de l'exécution à tout moment, ou carrément de stopper la simulation.

La plateforme permet le suivi et l'étude de l'évolution des algorithmes grâce à un suivi de l'historique d'exécution avec la possibilité de sauvegarder cet historique ou de l'utiliser ultérieurement pour la comparaison avec d'autres algorithmes.

Cette dernière se veut être une plateforme utile d'une part pour la communauté scientifique par le développement de nouveaux contrôleurs MPPT, facilitant ainsi les tests et simulations avec un prototypage rapide de ces derniers permettant une comparaison direct avec des contrôleurs existants déjà dans la base de données de la bibliothèque de la plateforme.

Ceci permettra ainsi aux chercheurs de se concentrer seulement sur le développement et le test de nouveaux contrôleurs MPPT pour le suivi du PPM, rendant ainsi la simulation et le test facile et accessible à tous. De ce fait, la plateforme acquiert ainsi un caractère évolutif (ajout de nouveaux panneaux PV, ajout de nouveaux contrôleurs dans la bibliothèque *PVSP*, après leur validation, etc...). La **figure.5.1** illustre la structure de la plateforme *PVSP*.

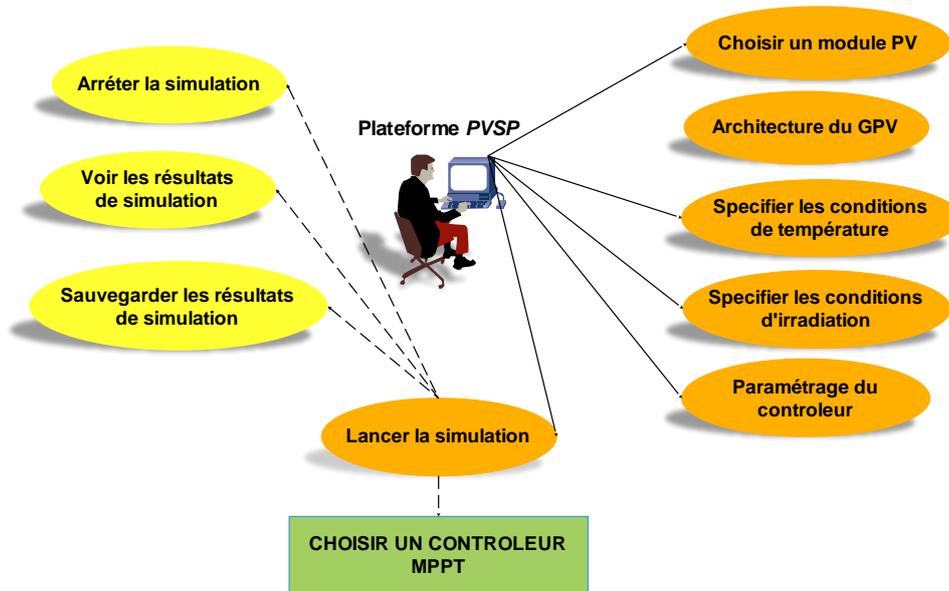


Figure.5.1 Structure de la plateforme PVSP.

Avant de se lancer dans la conception et l'élaboration de la plateforme, un certain nombre de points ont été pris en considération, à savoir :

- le choix de l'architecture du système photovoltaïque à simuler

Nous avons choisi une architecture basique (Figure.1.13.), constituée d'un générateur photovoltaïque GPV, d'un *convertisseur statique de type Boost*, d'une *charge résistive* et d'un bloc *contrôleur MPPT* contenant l'algorithme à étudier. L'architecture du GPV est constituée d'un certain nombre de panneaux PV avec une configuration en série-parallèle dont le nombre est à configurer par l'utilisateur ;

- Le choix du modèle électrique du panneau solaire

Pour plus de précision, le modèle électrique à deux diodes présenté au *chapitre.1.* a été sélectionné ;

- Le choix des scénarios des conditions atmosphériques pour la simulation

L'architecture du GPV peut contenir plusieurs configurations (nombre de panneaux en série-parallèle) à sélectionner par l'utilisateur afin de prendre en compte les différents cas des variations atmosphériques et de ce fait, tester plusieurs scénarios, en attribuant à chaque groupe différents niveaux de variations (irradiation, température et ombrage);

- Le caractère évolutif de la plateforme

Afin d'acquiescer un caractère évolutif, la plateforme dispose d'une bibliothèque permettant à l'utilisateur d'insérer de nouveaux contrôleurs, après test et validation de ces derniers. De plus, l'ajout de nouveaux modèles de panneaux PV est possible, permettant ainsi d'élargir le choix pour la sélection de nouveau système PV.

5.4 ARCHITECTURE DE LA PLATEFORME PVSP

Comme mentionné précédemment, l'architecture de notre plateforme a été conçue en nous inspirant de l'architecture typique du système photovoltaïque illustrée par la **figure.1.13**. Cette dernière, comme illustrée par la Figure.5.2, est constituée des modules : *Générateur PV*, *Algorithme MPPT*, *Contrôleur MPPT* et *Afficheur*, interconnectés entre eux pour former le système PV global.

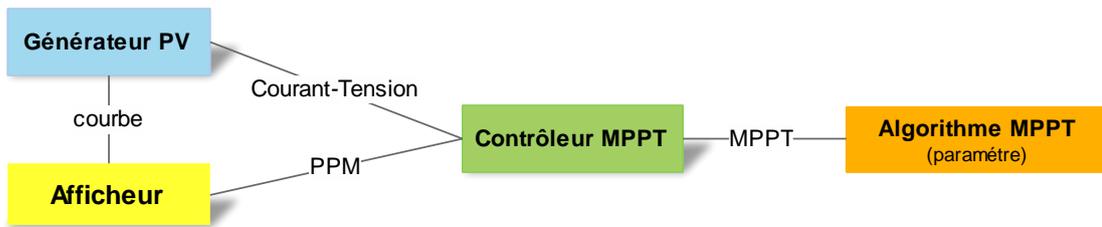


Figure.5.2 Les modules constituant le système PV global de la plateforme.

Chaque module de la plateforme est autonome et est appelé à réaliser une fonction bien précise.

5.4.1 Le module "GPV"

Le rôle de ce module est de simuler le comportement du générateur photovoltaïque, qui consiste principalement en la génération d'un courant et d'une tension tout en considérant les niveaux des conditions climatiques durant la simulation. Ce module comprend aussi les mécanismes utilisés pour simuler le comportement des conditions environnementales telles que le niveau d'éclairage, la température, ainsi que les changements de ces paramètres en vue de produire des situations de l'ombrage partiel.

5.4.2 Le module "Contrôleur MPPT"

Ce module permet d'assurer la recherche du PPM, il joue le rôle d'intermédiaire entre l'élément « GPV » et l'élément qui contient l'algorithme de contrôle. Il permet de :

- Capturer le courant et la tension générés par le module « *GPV* » périodiquement, ces paramètres sont passés par la suite au module contenant l'algorithme MPPT ;
- Forcer le module « *GPV* » à travailler sur un couple tension/courant donné ;
- Fournir le PPM calculé à l'élément « *Afficheur* » afin de permettre l'affichage de ce point en temps réel.

Ce module est complètement indépendant de l'algorithme de contrôle, ce qui permet d'avoir une plateforme évolutive, avec l'implémentation et l'insertion de nouveaux algorithmes dans le module « *Algorithme MPPT* ». Cela permet également au concepteur de l'algorithme de se focaliser sur la conception de l'algorithme sans se soucier des détails du générateur photovoltaïque et le module d'affichage des résultats.

5.4.3 Le module "*Algorithme MPPT*"

C'est une bibliothèque contenant un ensemble d'algorithmes MPPT testés et validés. Elle peut également accueillir de nouveaux algorithmes que le concepteur pourrait introduire, en vue d'une comparaison avec les contrôleurs existants.

5.4.4 Le module "*Afficheur*"

Il est utilisé pour illustrer (tracer) les différentes courbes de caractéristiques I-V et P-V, ainsi que pour le suivi de la trace d'évolution du PPM en temps réel.

5.5 DÉVELOPPEMENT DE LA PLATEFORME PVSP

Pour le développement de la plateforme, nous avons opté pour un ensemble d'outils qui repose sur la bibliothèque Qt version 5.4.1 [141]. C'est un *Framework* orienté objet, développé en C++. Ce dernier offre des composantes d'interface graphique (GUI), qui repose sur la bibliothèque Qwt version 6.1.2 [142], utile pour l'affichage des courbes I-V et P-V et des tableaux.

Comme présenté dans la **section 5.4**, la plateforme est structurée en plusieurs éléments comme l'illustre la figure ci-dessous.

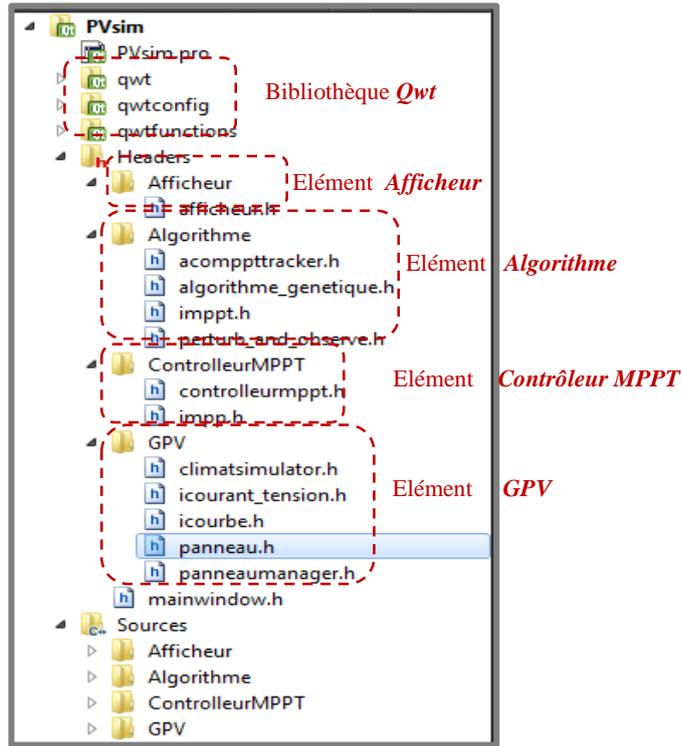


Figure.5.3 Structure software de la plateforme.

La plateforme *PVSP* développée, illustrée en figure.5.4 est dotée d'une interface utilisateur graphique conviviale. Elle est constituée de deux onglets générant deux interfaces principales : *Simulation* et *Comparaison*.

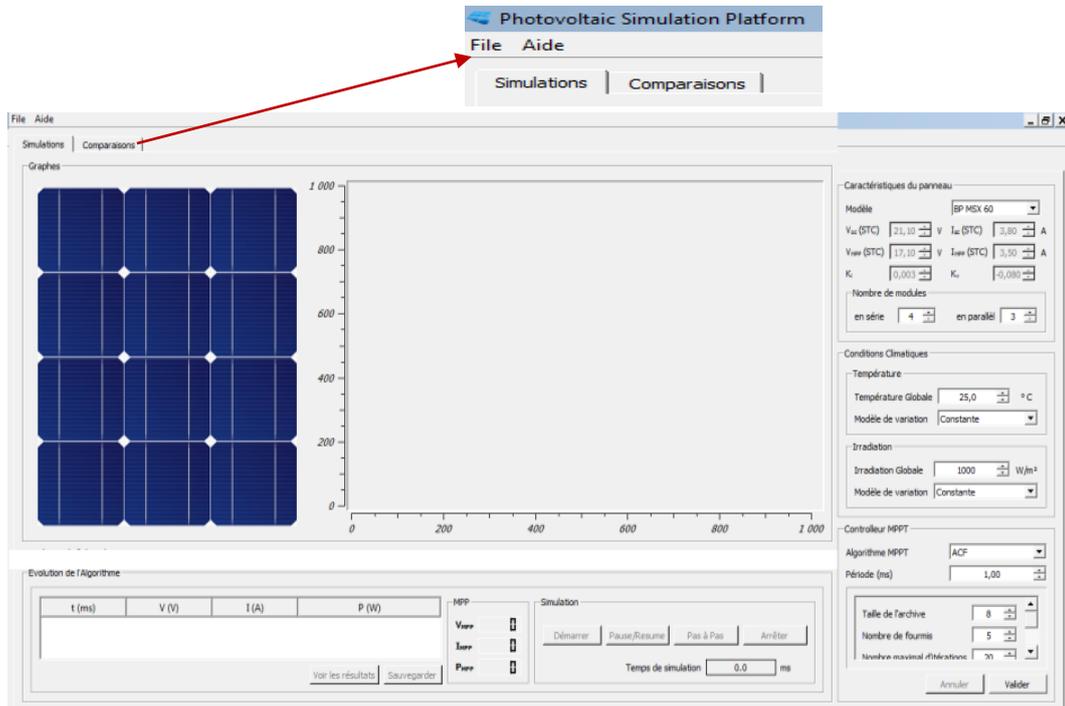


Figure.5.4 Interface graphique principale de la plateforme

5.5.1 L'onglet "Simulation"

La figure.5.5 illustre l'interface principale de *PVSP* lors de l'activation de l'onglet *Simulation*. Ce dernier permet d'activer l'interface de configuration, de test et de simulation des différents contrôleurs MPPT.

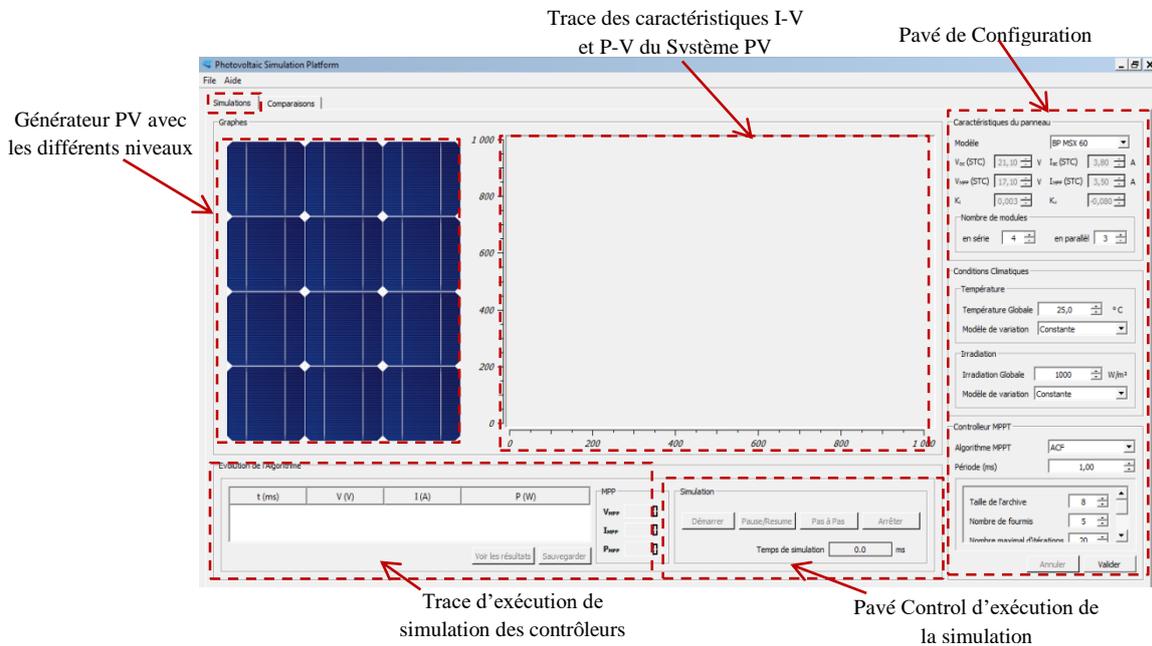


Figure.5.5 Interface graphique principale de la plateforme avec activation de l'onglet *Simulation*.

Comme l'illustre la Figure.5.5., l'interface configuration, test et simulation est composée des éléments suivants :

- Le *pavé de configuration* qui permet de configurer le générateur photovoltaïque (type du panneau, nombre de module, etc...), les conditions climatiques (Eclaircement, Température, taux de l'ombrage) et les paramètres de l'algorithme MPPT à tester ;
- Une visualisation du générateur photovoltaïque montrant le niveau d'irradiation appliquée sur chaque groupe de panneaux solaires ;
- Le graphe montrant les courbes caractéristiques de courant en fonction de la tension I-V et de puissance en fonction de la tension P-V. Il est possible également de sauvegarder les mesures de tension, de courant et puissance correspondantes à ces caractéristiques à partir du menu principal ;
- Le pavé de contrôle de la simulation permettant de démarrer, mettre en pause ou d'arrêter la simulation ;
- La trace d'exécution de l'algorithme MPPT montrant l'évolution des valeurs de courant, de tension et de puissance en fonction de temps.

5.5.2 L'onglet "Comparaison"

La figure.5.6 illustre l'interface principale de PVSP lors de l'activation de l'onglet *Comparaison*. Ce dernier, permet, suite à une simulation, d'effectuer une comparaison entre les traces d'exécution des différents algorithmes MPPT.

L'interface est composée des éléments suivants :

- *Trace d'exécution* : Pour chaque contrôleur, l'historique des différentes simulations est représenté, à la fois, sous forme de tableau ainsi que sous la forme d'un graphe évoluant dans le temps ;
- *Tableau récapitulatif* : C'est un tableau regroupant une liste de profils de simulation indiquant, pour chaque exécution d'un contrôleur, les différents paramètres et configurations utilisés lors de la simulation, ainsi qu'un résumé des résultats obtenus. Cette fonctionnalité permet de comparer plusieurs simulations en termes de temps d'exécution, de stabilité et d'efficacité de l'algorithme.

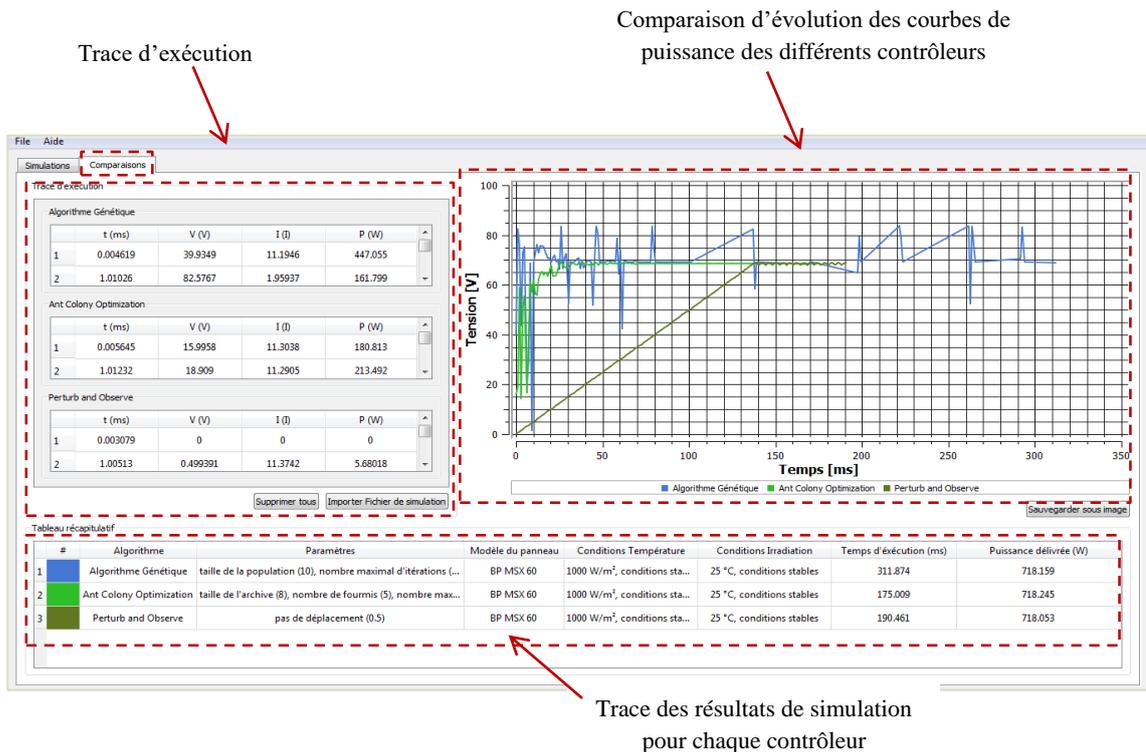


Figure.5.6 Interface graphique principale de la plateforme avec activation de l'onglet *Comparaison*.

La prochaine section détaillera les fonctionnalités relatives à chaque onglet de la plateforme.

5.6 FONCTIONNALITÉS DE LA PLATEFORME PVSP

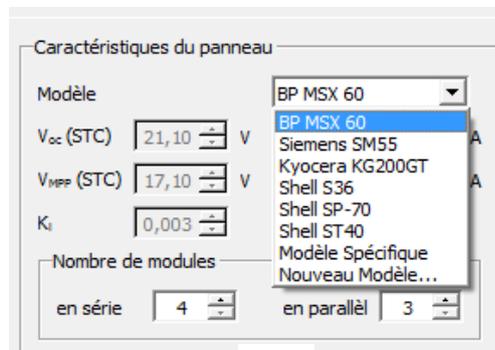
Comme mentionné dans la *Section 5.3*, PVSP propose, par sa structure, un cadre méthodologique de développement, simulation et test des contrôleurs MPPT.

Lors de l'élaboration d'un contrôleur MPPT (test et simulations), le concepteur devra tenir compte d'un certain nombre de configurations. Dans ce qui suit, nous allons expliciter chacune de ces étapes de configuration.

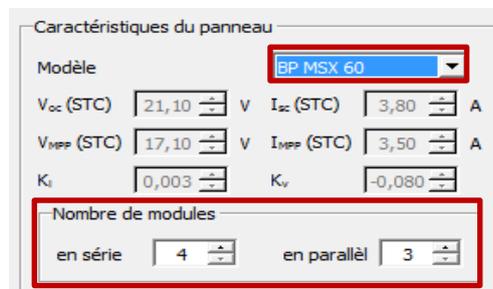
5.6.1 La sélection du modèle du panneau PV et élaboration de l'architecture du GPV

La configuration du générateur implique le choix du type du panneau PV ainsi que les dimensions du générateur PV. Cette fonctionnalité permet à l'utilisateur de choisir parmi plusieurs types de panneaux PV qui existent sur le marché et dont les caractéristiques sont prédéfinies. Cette même fonctionnalité permet aussi de définir des modèles spécifiques ou de créer carrément de nouveaux modèles et les sauvegarder dans une bibliothèque de modèles.

La figure.5.7 illustre respectivement la sélection du modèle du panneau PV et le dimensionnement du générateur PV (nombre de modules série/parallèle).



(a)



(b)

Figure.5.7 (a) Sélection du modèle du panneau PV,
(b) Dimensionnement du générateur PV (nombre de modules S/P).

5.6.2 Spécification des conditions atmosphériques

Lors de cette phase, le concepteur introduit les données relatives aux conditions climatiques (Température, Irradiation, Taux d'ombrage) permettant ainsi de parcourir lors des différentes simulations plusieurs cas, mettant ainsi en exergue la robustesse du contrôleur sélectionné ; Comme l'illustre la figure ci-dessous, la configuration de ces conditions inclut le choix des valeurs de température et d'irradiation solaire et le choix du modèle de variation de ces conditions.

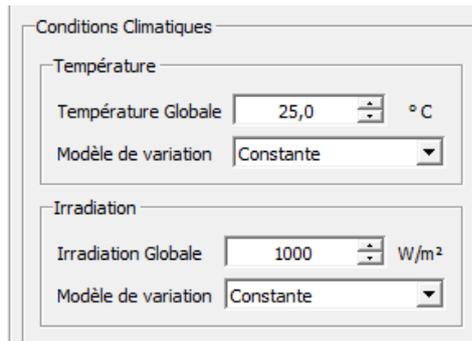


Figure.5.8 Configuration des conditions climatiques.

Pour ce faire, il est possible de choisir entre trois (3) modèles de variations atmosphériques :

- Le premier modèle permet de spécifier des valeurs constantes de température et/ou d'irradiation pour tous les panneaux du système PV ;

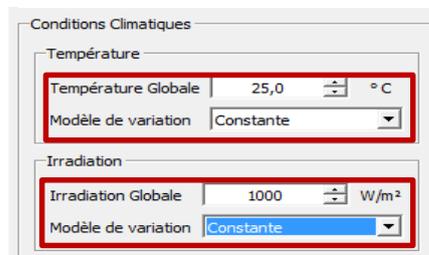


Figure.5.9 Configuration des conditions climatiques : variation stable en Température et Irradiation.

- Le deuxième modèle permet de spécifier des valeurs qui varient rapidement dans le temps ;

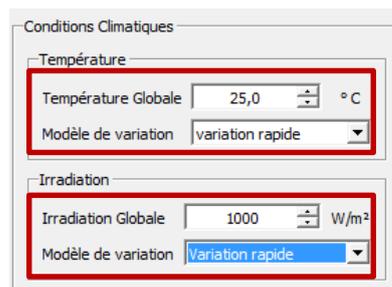
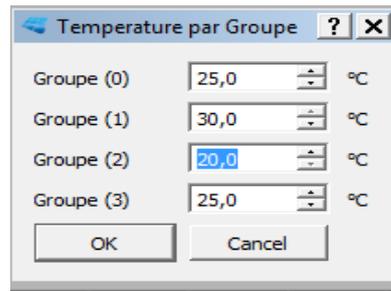
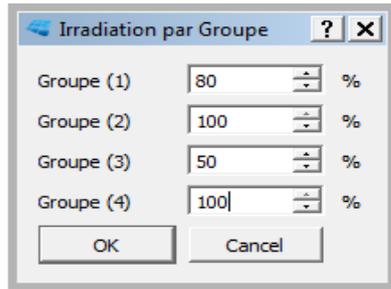


Figure.5.10 Configuration des conditions climatiques : variation rapide en Température et Irradiation.

- Le troisième modèle permet de spécifier des valeurs pour chaque groupe de panneaux solaires.



(a)



(b)

Figure.5.11 Configuration des conditions climatiques : variation spécifique pour chaque panneau (a) en Température, (b) en Irradiation.

- Enfin, pour l'irradiation solaire, un quatrième modèle permet de spécifier un changement dans le niveau d'irradiation « ombrage partiel » de chaque groupe qui se déclenche après un temps spécifié par le concepteur.

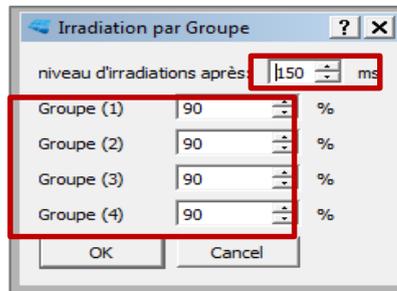


Figure.5.12 Configuration des conditions climatiques : ombrage partiel déclenché avec temporisateur.

5.6.3 Sélection du contrôleur MPPT

Lors de cette phase, le concepteur devra sélectionner dans la bibliothèque un contrôleur. Parmi les contrôleurs développés et stockés dans la bibliothèque PVSP, nous pouvons citer : *Perturbe and Observe* (P&O), *Incremental Conductance* (IC), les *Algorithmes Genetic* (AG) et les *Essaims de Particules* (PSO). Ces derniers appartiennent respectivement aux méthodes conventionnelles (MC), d'intelligence artificielle (MIA) et les méthodes Bio inspirées (MBI).

La Figure.5.13 illustre l'interface permettant de sélectionner de la bibliothèque respectivement : les contrôleurs P&O et PSO, de paramétrer ces derniers.

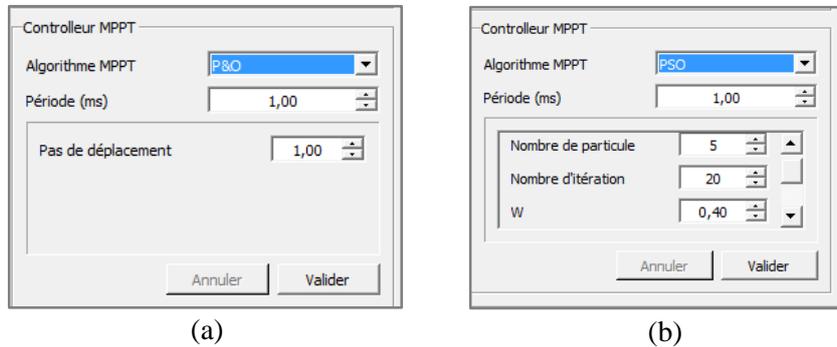


Figure.5.13 Sélection et paramétrage des contrôleurs, (a) P&O, (b) PSO.

Il est utile de rappeler que la plateforme *PVSP* est une plateforme évolutive : après chaque test et validation d'un nouveau contrôleur, le concepteur pourra enrichir la bibliothèque en intégrant le contrôleur ainsi développé et validé dans la base de données (*bibliothèque*) de cette dernière.

5.6.4 Simulation du contrôleur MPPT

Lorsque toutes les étapes de la configuration du générateur photovoltaïque, des conditions climatiques et le choix de l'algorithme MPPT accomplies et validées, vient alors l'étape de simulation du contrôleur MPPT ainsi sélectionné et paramétré. L'onglet *démarrer simulation* est alors activé (Figure.5.14).

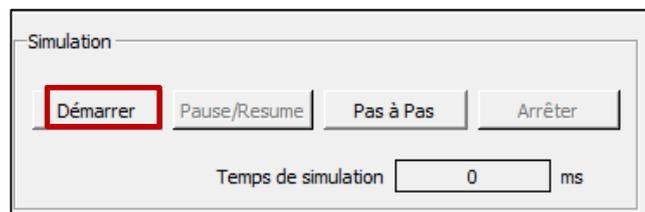


Figure.5.14 Lancement de simulation.

Suite à cette phase de simulation, une trace de l'évolution de la puissance (Figure.5.15), est illustrée par le déplacement du point PPM sur la courbe caractéristique de puissance et par la trace d'évolution des valeurs de courant, de tension et de puissance au cours du temps.

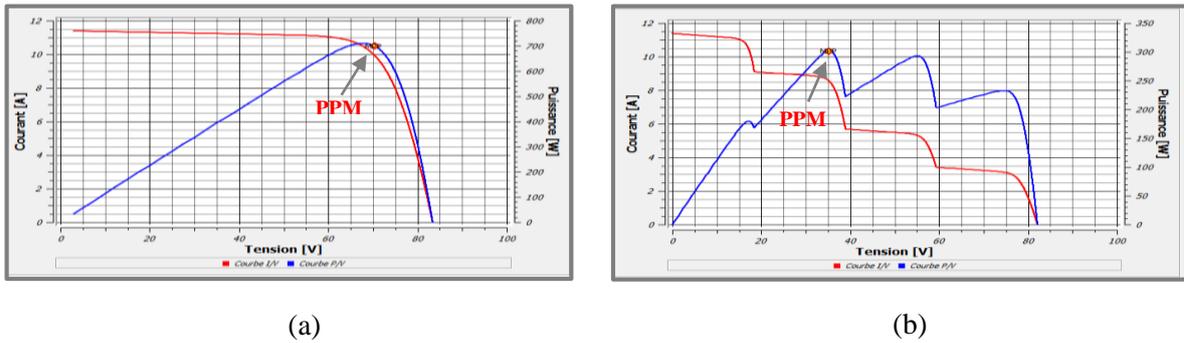


Figure.5.15 Trace de l'évolution I-V et P-V et visualisation du PPM,
(a) condition standard, (b) sous ombrage partiel.

5.6.5 Comparaison des résultats de simulations

Une fois que les différents scénarios relatifs aux différentes simulations sont exécutés, la plateforme permet alors de faire une comparaison entre les différentes traces d'exécutions relatives :

- au choix de sélection des paramètres adéquats pour l'élaboration d'un contrôleur efficace ;
- aux différents algorithmes MPPT sélectionnés en vue d'une comparaison en termes de performances pour la sélection d'un contrôleur adéquat.

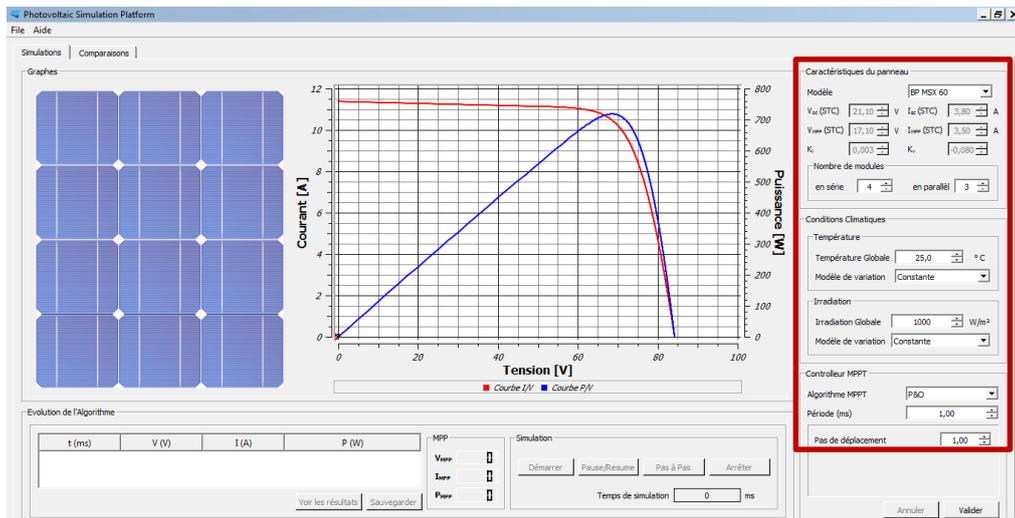
Dans ce qui suit et afin de valider la fonctionnalité et démontrer les performances de la plateforme développée, nous allons expliciter chacune de ces différentes étapes de configuration.

Pour cela, nous avons sélectionné le contrôleur *P&O_MPPT* de la bibliothèque. Pour la simulation, nous avons sélectionné le panneau *BMX60*, avec une configuration de 4 modules en série et 3 en parallèle.

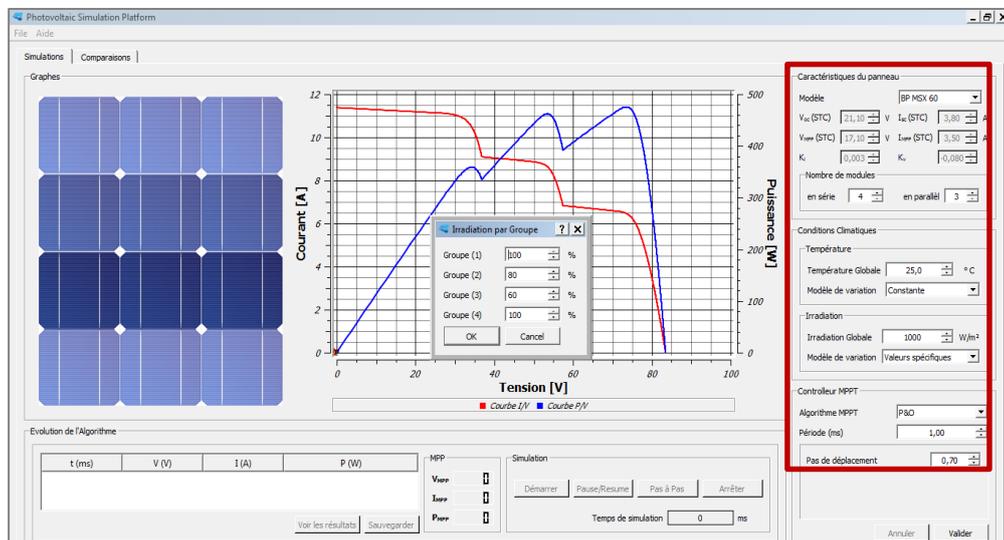
La simulation est faite en premier lieu sous les mêmes conditions atmosphériques, avec les mêmes configurations du système PV, cependant le pas d'exécution de l'algorithme est modifié lors de chaque simulation, permettant ainsi de suivre la trace de l'évolution de ce dernier et de faire une comparaison.

En second lieu, nous avons gardé la même configuration du SPV, toutefois, le contrôleur P&O est testé dans le cas où le SPV est partiellement ombragé, avec un éclairement de 100% pour le premier et le quatrième groupe, 80% pour le second et 60% pour le troisième.

La figure.5.16 illustre la configuration de la plateforme pour la simulation des deux scénarios.



(a)



(b)

Figure.5.16 Configuration de la plateforme pour la simulation du contrôleur P&O, (a) condition standard, (b) sous ombrage partiel.

Les résultats de simulation sont illustrés par les Figures.5.17 à 5.22 où l'évolution des courbes de puissance dans le temps et une trace d'exécution de la simulation sont présentés.

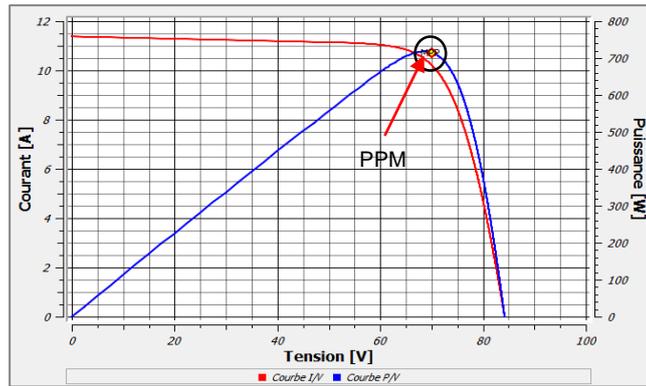


Figure.5.17 Caractéristique P-V et I-V sous condition standard.

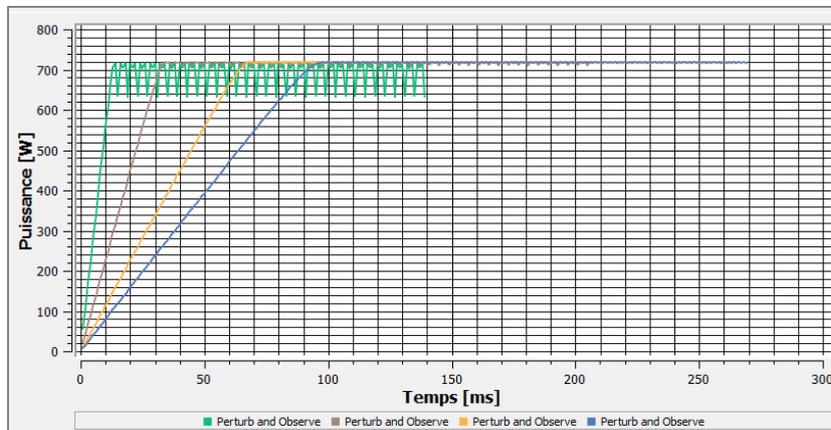


Figure.5.18 Trace de l'évolution de la puissance dans le temps sous condition standard.

Tableau récapitulatif							
#	Algorithme	Paramètres	Modèle du panneau	Conditions Température	Conditions Irradiation	Temps d'exécution (ms)	Puissance délivrée (W)
1	Perturb and Observe	pas de déplacement (5)	BP MSX 60	1000 W/m ² , conditions sta...	25 °C, conditions stables	139.001	633.031
2	Perturb and Observe	pas de déplacement (2)	BP MSX 60	1000 W/m ² , conditions sta...	25 °C, conditions stables	207.001	715.296
3	Perturb and Observe	pas de déplacement (1)	BP MSX 60	1000 W/m ² , conditions sta...	25 °C, conditions stables	265.001	718.007
4	Perturb and Observe	pas de déplacement (0.7)	BP MSX 60	1000 W/m ² , conditions sta...	25 °C, conditions stables	269	717.66

Figure.5.19 Trace d'exécution du contrôleur P&O.

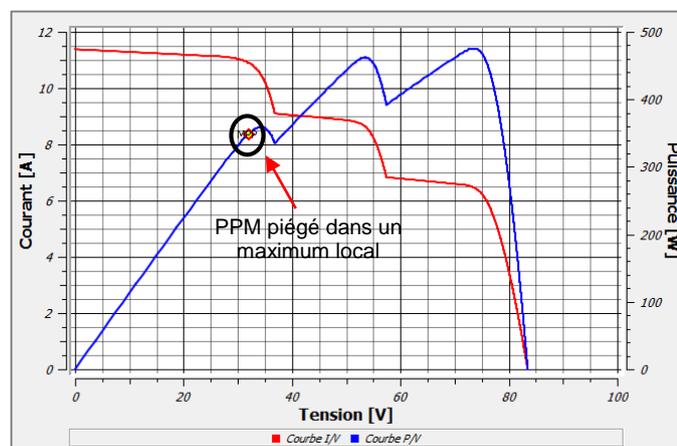


Figure.5.20 Caractéristique P-V sous condition d'ombrage partiel.

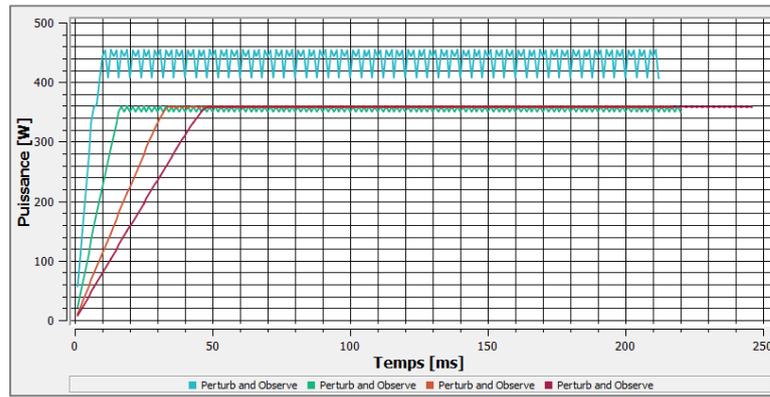


Figure.5.21 Trace de l'évolution de la puissance dans le temps sous ombrage partiel.

Tableau récapitulatif							
#	Algorithme	Paramètres	Modèle du panneau	Conditions Température	Conditions Irradiation	Temps d'exécution (ms)	Puissance délivrée (W)
1	Perturb and Observe	pas de déplacement (5)	BP MSX 60	1000 W/m ² , conditions om...	25 °C, valeurs différentes	212.001	406.806
2	Perturb and Observe	pas de déplacement (2)	BP MSX 60	1000 W/m ² , conditions om...	25 °C, valeurs différentes	220.001	348.893
3	Perturb and Observe	pas de déplacement (1)	BP MSX 60	1000 W/m ² , conditions om...	25 °C, valeurs différentes	217.001	355.385
4	Perturb and Observe	pas de déplacement (0.7)	BP MSX 60	1000 W/m ² , conditions om...	25 °C, valeurs différentes	246.001	357.629

Figure.5.22 Trace d'exécution du contrôleur P&O sous ombrage partiel.

Discussion des résultats

Les résultats de simulations relatives aux deux scénarios (conditions standard et sous ombrage partiel) sont illustrés par les figures.5.17 à 5.22.

Ces dernières, montrent respectivement, la caractéristique P_V et I-V sous conditions standard et ombrage exhibant le PPM atteint ; une trace de l'évolution de la puissance en fonction du temps pour différentes valeurs de pas d'exécution et une trace d'exécution du contrôleur.

Au vu des résultats obtenus, nous constatons que le PPM est atteint, toutefois, sous condition standard, lorsque le pas d'exécution est grand, le PPM est atteint mais les oscillations sont très importantes. En revanche, lorsque le pas d'exécution est petit, le contrôleur met du temps pour atteindre le PPM et les oscillations sont toujours présentes mais plus faibles.

Dans le cas de l'ombrage partiel, le PPM est piégé dans un maximum local engendrant une perte de puissance importante. Toutefois, lorsque le pas d'exécution est grand, le contrôleur parvient à surmonter le premier maximum local pour se voir piégé dans le second comme l'illustre la figure.5.23.

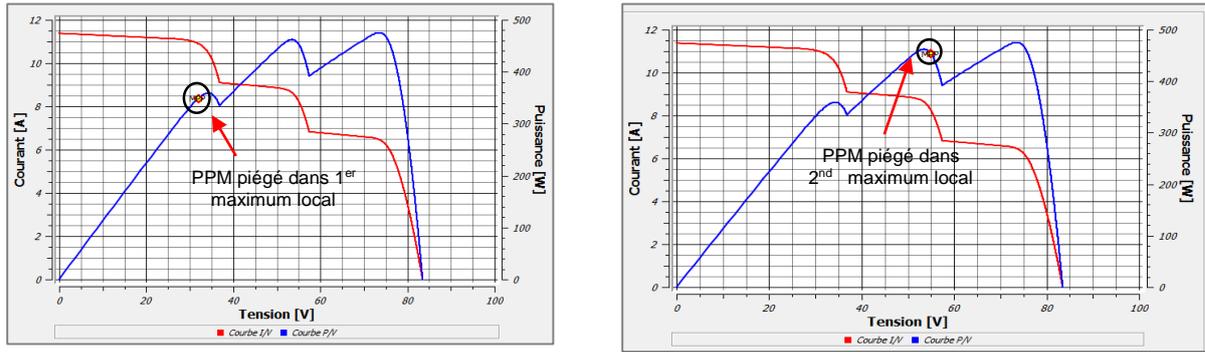


Figure.5.23 (a)-(b) Caractéristique P-V sous condition d'ombrage partiel.

5.7 CONCLUSION

Au cours de ce chapitre, nous avons exposé un état de l'art sur les plateformes des EnRs existantes, puis nous avons présenté l'architecture de la plateforme *PVSP*, qui intègre en un seul outil toutes les phases nécessaires pour le développement et la simulation d'un contrôleur MPPT, soumis à diverses contraintes environnementales (éclairage, température, ombrage partiel...), ainsi que ses fonctionnalités.

Il est intéressant de noter que cette plateforme proposée est évolutive, puisqu'elle contient une liste non exhaustive de contrôleurs MPPT que les concepteurs pourraient enrichir en intégrant de nouveaux, après tests et validation de ces derniers.

En facilitant les simulations avec un prototypage rapide, la plateforme *PVSP* constitue une plateforme utile pour la communauté scientifique dont les travaux de recherche portent sur le développement de nouveaux contrôleurs MPPT.

De plus, cette plateforme peut être perçue comme un outil pédagogique innovant qui pourrait être destiné à l'enseignement des connaissances de base pour l'optimisation de puissance dans les systèmes et les applications photovoltaïques.

PARTIE III

**APPROCHE HARDWARE POUR
L'IMPLÉMENTATION DES CONTROLEURS
MPPT APPLIQUEE AUX ALGORITHMES
ACO_NPU ET BSO**

Chapitre 6

**ÉTAT DE L'ART SUR L'IMPLÉMENTATION
DES CONTROLEURS MPPT
SUR CIRCUIT FPGA**

6.1 INTRODUCTION

La Partie II s'est soldée par la conception et la mise en œuvre des deux contrôleurs bio inspirés ACO-NPU_MPPT et BSO_MPPT. Afin de mettre en exergue le choix de ces deux contrôleurs, il est apparu judicieux de cibler une méthode d'implémentation hardware pertinente afin de conforter davantage les résultats obtenus grâce à ce choix.

Avec l'avancement technologique dans le domaine de la microélectronique, de nouvelles solutions numériques telles que les FPGA (Field Programmable Gate Array) ou les ASIC (Application Specific Integrated Circuit) sont disponibles et peuvent être utilisées comme cibles numériques pour l'implantation des algorithmes pour le control.

Le parallélisme inhérent de ces nouvelles solutions ainsi que leurs grandes capacités de calcul font que les délais de temps de calcul sont réduits malgré la complexité des algorithmes à implanter. En effet, outre l'amélioration des performances à travers la réduction des temps de calcul, le parallélisme des solutions matérielles permet d'intégrer, sur un seul et unique circuit, plusieurs algorithmes qui assurent différentes fonctionnalités et qui peuvent travailler indépendamment les uns des autres.

Au vu des avantages avérés des circuits FPGA, ces derniers sont devenus au fil des temps incontournables dans la conception des systèmes numériques, et ce, dans les multiples domaines où ces systèmes sont utilisés. Le domaine lié aux énergies renouvelables n'est pas en reste. En effet, l'utilisation des contrôleurs MPPT pour l'optimisation de puissance dans les systèmes PV s'est imposée comme solution technologique incontournable.

C'est dans cette optique, et dans le cadre de nos travaux de recherche, que nous avons ciblé les circuits FPGA pour l'implémentation hardware des deux contrôleurs bio inspirés.

Au cours de ce chapitre, nous présenterons un état de l'art sur l'implémentation des contrôleurs MPPT sur circuits FPGA. Aussi, nous aborderons, dans un premier temps, les circuits programmables « **Field Programmable Gate Array** » (**FPGA**), avec leurs principales caractéristiques, leur architecture, leur évolution, ...etc.

Puis dans un second temps, nous examinerons l'apport des circuits FPGA pour le control MPPT, où une synthèse des différents travaux existants dans la littérature sera présentée.

6.2 LES CIRCUITS “*FIELD PROGRAMMABLE GATE ARRAY*” (FPGA)

6.2.1 Historique

Les circuits logiques programmables sont apparus au début des années 80. Les développeurs disposaient déjà de circuits de type « **P**rogrammable **L**ogic **D**evice » (PLD), facilement configurables. Seulement, ces derniers ne pouvaient pas contenir des designs trop complexes.

Les circuits de type « **A**pply Specific **I**ntegrated **C**ircuits » (ASICs), quant à eux, supportaient des systèmes de grande complexité, mais en revanche, n'avaient pas les propriétés de configuration des PLDs. Ils représentaient, de ce fait, une tâche coûteuse en temps et en argent. La réalisation d'un prototype nécessitait la création d'un ou de plusieurs masques très coûteux dont une simple erreur dans le système impliquait la création d'un nouveau(x) masque(s). Cela impliquait donc un investissement en temps de développement important, ce qui n'est pas tolérable pour des produits devant être rapidement mis sur le marché.

Cet état de fait a conduit la communauté scientifique à orienter leurs travaux de recherches vers la conception d'un circuit, qui, à la fois, permet d'embarquer des systèmes de plus en plus complexes, tout en consentant la reconfiguration du circuit de manière rapide avec des frais de mise en œuvre peu coûteux.

C'est pourquoi qu'en 1984, Ross Freeman, Bernie Vonderschmitt et Jim Barnett fondent la compagnie Xilinx. En 1985, ils introduisent sur le marché le XC2064, le premier circuit **F**ield **P**rogrammable **G**ate **A**rray (FPGA), qui est un circuit intégré reprogrammable, offrant une alternative aux précédentes approches.

C'est ainsi que les FPGA sont devenus des éléments incontournables des flots de conception de circuits numériques complexes. Leur apport au niveau de la phase de prototypage des systèmes en cours de conception est indéniable.

De plus, l'évolution rapide des techniques de fabrication et des technologies ont permis d'avoir des circuits FPGA de plus en plus performants et de plus en plus complexes. Nous trouvons ainsi dans les FPGAs de dernière génération des processeurs embarqués, des liens de communications ultrarapides et bien d'autres fonctionnalités. Ils sont aussi de plus en plus denses (jusqu'à 1 milliard de portes logiques équivalentes) et de plus en plus rapides (jusqu'à

500 MHz). Tout cela permet la réalisation de systèmes très complexes et très performants que ceux réalisés jusqu'alors avec les circuits ASICs, tout en ayant le grand avantage de pouvoir être programmés électriquement.

6.2.2 Les principales caractéristiques des circuits FPGA

Parmi les principales caractéristiques des circuits FPGA, nous pouvons citer [143] :

6.2.2.1 Les performances

Le mode d'exécution parallèle permet aux FPGAs d'avoir une puissance de calcul bien plus importante que celle des processeurs à signaux numériques (DSP) car ce mode parallèle, comparativement au mode séquentiel, permet d'exécuter plusieurs opérations par cycle d'horloge ce qui réduit la durée d'exécution de manière considérable.

Selon le *BDTI (Berkley Design Technology Inc)*, les circuits FPGAs peuvent offrir une puissance de calcul (traitement/dollar) bien plus supérieure que ce que permettrait d'atteindre une solution DSP pour le cas de certaines applications [144]. En effet, contrôler les entrées et sorties (E/S) au niveau matériel (hardware) permet d'atteindre des temps de réponse bien plus courts. Cela est d'autant plus intéressant lorsqu'il s'agit d'implémenter des fonctionnalités spécifiques afin de répondre à des besoins précis de l'application.

6.2.2.2 Le temps de mise sur le marché

Afin de répondre aux considérations économiques notamment celles inhérentes à réduire le temps de mise sur le marché (Time To Market), la technologie FPGA, grâce à ses capacités de prototypage rapide, constitue en soi, une solution souple et rapide et permet la réalisation de systèmes relativement complexes.

En effet, avec cette technologie, on peut, aisément, à partir d'une simple idée ou concept, l'élaborer, le vérifier sur un banc d'essai matériel, tout cela sans avoir à supporter le long processus de fabrication d'un ASIC personnalisé. A ce stade du processus, on pourra imaginer toutes sortes de modifications, de transformations et tout ce qui est nécessaire pour notre FPGA avant de finaliser la conception, qui pourra se faire en quelques heures au lieu de prévoir des jours, voire des semaines.

6.2.2.3 Le coût

Il est tout à fait clair que les coûts d'ingénierie non récurrents (NRE) des ASIC personnalisés sont bien plus importants que ceux des solutions matérielles basées sur le FPGA. Seulement, il ne faudrait pas perdre de vue que l'investissement de départ que requièrent les ASIC peut être justifié pour les OEM (Original Equipment Manufacturer). En effet, lorsqu'un utilisateur opte pour cette solution, ce n'est pas consenti pour la réalisation d'un système unique, mais bien au contraire, cela devient intéressant pour le développement de dizaines voire de centaines de systèmes personnalisés.

De par leur nature, la réalisation de circuits programmables ne nécessite pas des coûts de fabrication élevés, ni de longs délais d'assemblage. Dans la plupart des systèmes, les besoins de ces derniers évoluent avec le temps. C'est pour cela que la modification évolutive d'un circuit FPGA nécessite un coût négligeable si on le compare au coût qu'aurait généré la conception d'un nouvel ASIC.

6.2.2.4 La fiabilité

Le tandem Hardware/Software, illustré par la paire FPGA/outils logiciels, fait que les FPGA sont une véritable implémentation matérielle pour l'exécution des outils logiciels composant l'environnement de programmation. L'une des particularités des systèmes basés processeur est qu'ils présentent plusieurs couches d'abstraction afin de permettre la planification des tâches et la répartition des ressources entre les processus.

C'est la couche de driver qui contrôle les ressources matérielles et c'est le système d'exploitation qui gère la mémoire et la bande passante du processeur. Chaque cœur du processeur exécute une seule instruction à la fois. A l'inverse des systèmes basés processeur, les FPGA n'utilisent pas de système d'exploitation ce qui minimise considérablement ces problèmes de fiabilité vus précédemment en assurant une exécution parallèle et un matériel déterministe dédié à chaque tâche.

6.2.2.5 La maintenance à long terme

Les circuits FPGAs sont évolutifs et permettent un gain de temps et de coût important qu'impliquerait la conception à nouveau des ASIC. Si l'on considère, à titre d'exemple, les spécifications des protocoles de communication numériques, il est clair que ces derniers évoluent avec le temps. Il s'avère alors que les interfaces basées sur ASIC peuvent présenter

des soucis de maintenance et de compatibilité. Seulement, comme les circuits FPGAs sont reconfigurables, ils sont donc capables de s'adapter aux modifications. C'est ainsi que l'on peut dire, qu'à mesure qu'un système évolue, toutes les améliorations fonctionnelles peuvent y être apportées sans pour autant perdre du temps à chercher à reconcevoir le circuit.

Tous ces avantages qu'offrent les circuits FPGAs expliquent en partie le succès qu'ils ont connu dans tous les secteurs d'activités. Ceci pourrait s'expliquer simplement par le fait qu'ils présentent à la fois les atouts des ASIC ainsi que ceux des systèmes basés processeur. Cela leur permet d'assurer, à la fois, vitesse et fiabilité tout en restant plus rentables économiquement parlant par rapport aux ASIC.

Par ailleurs, les circuits logiques reprogrammables jouissent de la même souplesse d'exécution logicielle qu'un système basé processeur. Les FPGAs, contrairement aux processeurs, sont vraiment parallèles par nature étant donné que plusieurs opérations de traitement ne se trouvent pas en concurrence pour l'allocation des ressources. Cela est rendu possible car chaque tâche de traitement indépendante est affectée à une section spécifique du circuit de manière à s'exécuter en toute autonomie sans dépendre des autres blocs logiques. Ainsi, grâce à ce mode d'exécution parallèle, le volume de traitement effectué est démultiplié sans que les performances d'une partie de l'application n'en soient affectées.

6.2.3 Le marché des circuits FPGA

Le marché des FPGAs connaît un grand essor avec des dizaines de fabricants, qui utilisent des technologies plus ou moins similaires pour la fabrication des FPGA. Nous pouvons citer à titre d'exemple : Actel, Abound Logic, Atmel, Cypress, Lattice Semi-conductor, Xilinx, Altera, ...etc. Parmi tous ces constructeurs, seul deux principales firmes détiennent le monopole du marché des circuits FPGA à savoir XILINX [145] et ALTERA [146].

Ces derniers ont traversé ces dernières années en éternels concurrents, assumant pleinement leur statut incontesté de leaders du domaine.

La figure.6.1 illustre une proportion des parts de marché pour chaque firme.

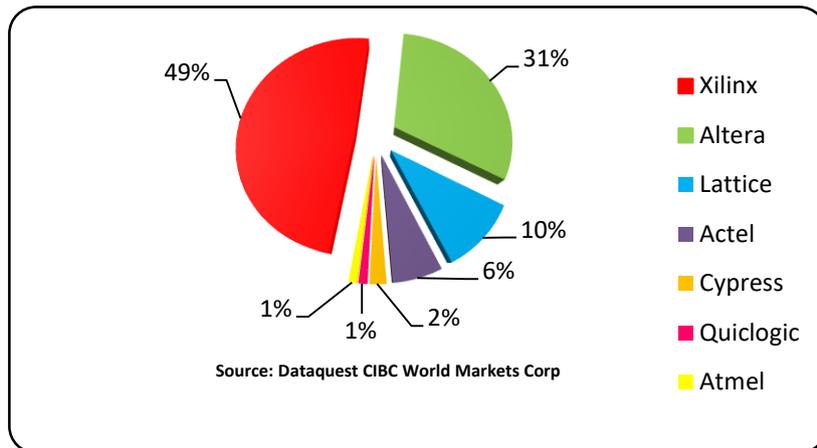


Figure.6.1. Les parts de marché des circuits FPGA pour chaque firme.

Dans ce qui suit, nous nous intéresserons aux circuits FPGA de la famille XILINX.

6.2.4 Architecture générale des circuits FPGA de XILINX

Les détails de conception des circuits FPGA varient d'un constructeur à un autre, néanmoins l'architecture générale consiste en un réseau de blocs logiques programmables, d'une matrice d'interconnexion programmable qui entoure les blocs logiques et de blocs d'entrées/sorties configurables et entoure le tous.

Cette architecture est illustrée par la Figure ci-dessous.

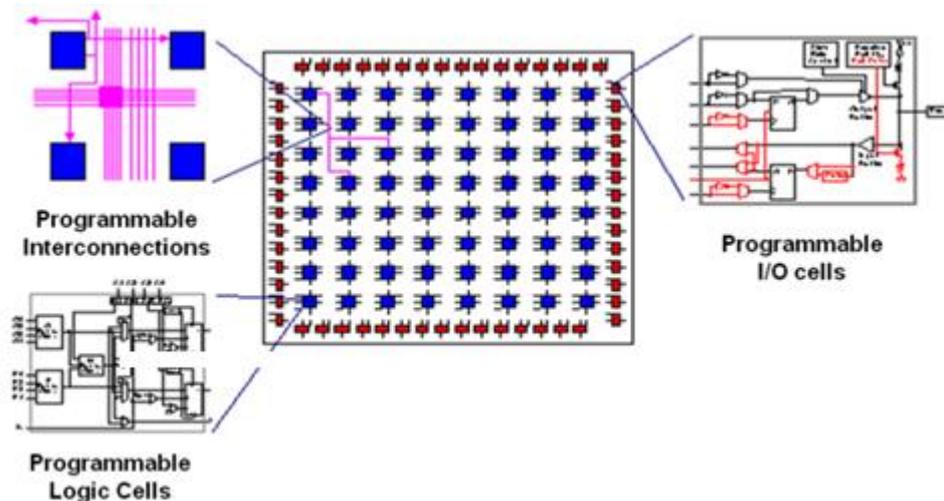


Figure.6.2. Architecture générale d'un circuit FPGA.

Ainsi, comme illustré en Figure.6.2, un circuit logique peut être implémenté dans un circuit FPGA en partitionnant la logique de ce dernier dans des blocs CLB (Programmable Logic Cells), interconnectant ces blocs à travers des switches nécessaires (Programmable

Interconnexions) et enfin établir une communication avec l'extérieur par les IOBs (Programmable I/O cells). Ces trois principaux éléments caractérisent un FPGA. Cependant, ces éléments différents (en taille et nombre d'entrée/sortie) selon les constructeurs et les types de circuit FPGA.

6.2.4.1 Structure des Bloc Logique (CLB)

Les blocs logiques configurables (CLB) illustré en Figure.6.3, sont les principaux éléments d'un FPGA. Ces cellules sont constituées d'éléments logiques programmables disposés sous forme matricielle et sont principalement composés de :

- table de transcodage LUT (Look-up Table) à K entrées qui permettent de synthétiser des fonctions combinatoires élémentaires ;
- bascule DFFs (D Flip-Flop) qui permettent la mémorisation et la composition de fonctions séquentielles ;
- Multiplexeurs qui permettent la sélection.

Chaque bloc logique est identique aux autres et peut être relié à ses voisins par le biais d'un réseau d'interconnexion.

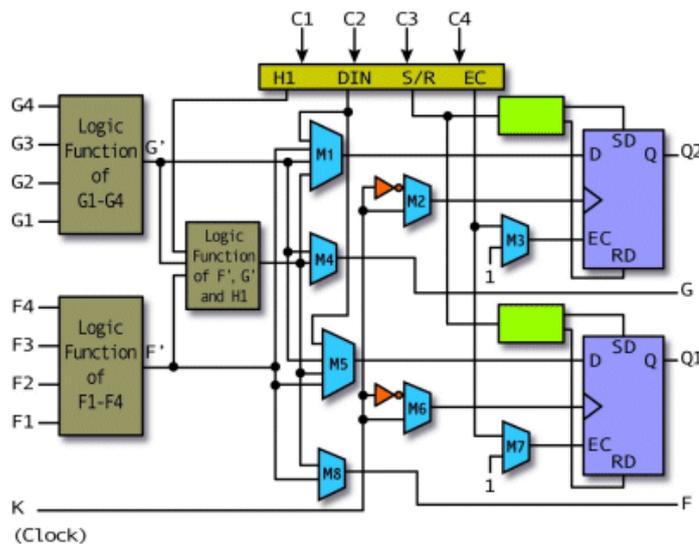


Figure.6.3. Architecture interne d'un CLB.

6.2.4.2 Les ressources d'interconnexions

Les connexions internes dans les circuits FPGAs sont composées de segments métallisés. Ces derniers sont utilisés pour connecter des blocs logiques définissant ainsi la topologie du circuit FPGA. Les lignes d'interconnexion, entourant les blocs logiques, sont organisées en lignes et en colonnes, formant ainsi des canaux de routage horizontaux et verticaux.

L'architecture de routage, comme illustré en Figure.6.4, est composée de trois éléments essentiels :

- Les boîtes de connexions (*Connection Box*), utilisées pour connecter les entrées sorties d'un bloc logique aux canaux de routage via des connexions programmables;
- Les boîtes de commutation (*Switch Box*), permettent de connecter des fils de routage appartenant à deux canaux de routage adjacents ;
- Les canaux de routage (*Horizontal & Vertical Chanel*), formés d'un certain nombre de fils de routage dans chaque canal. C'est à l'outil de routage de s'assurer que le nombre de signaux routés dans chaque canal ne dépasse pas la largeur du canal. De ce fait, si la largeur du canal est dépassée, alors le circuit à implémenter est dit non routable. Il faut alors sélectionner un circuit ayant plus de ressources.

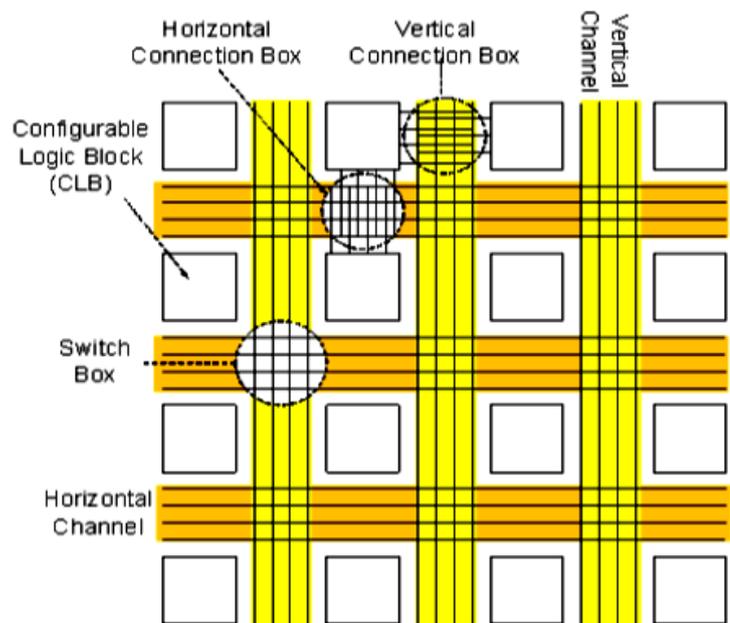


Figure.6.4. Ressources d'interconnexion.

6.2.4.3 Les blocs d'Entrée/Sortie (E/S)

Les blocs d'entrée-sortie permettent l'interconnexion de la logique interne aux ports d'entrées et de sorties du FPGA. Ces blocs sont présents sur toute la périphérie du circuit FPGA. Chaque bloc IOB contrôle une broche du composant et il peut être défini en entrée, en sortie, en signaux bidirectionnels.

Ainsi, tous les éléments suscités constituent l'architecture interne du circuit FPGA. Ces derniers diffèrent en taille et en nombre d'entrée/sortie selon le type du circuit utilisé. Il est à remarquer que l'architecture interne des circuits FPGA a beaucoup évolué ces dernières années.

6.2.5 Evolution des circuits FPGA de XILINX

Depuis leurs apparitions au milieu des années 80, les possibilités offertes par les circuits FPGAs de XILINX n'ont cessé d'augmenter traduisant une évolution et une révolution spectaculaires.

L'évolution d'abord, concerne l'amélioration du niveau de densité d'intégration qui est liée à l'évolution de la technologie microélectronique car nous sommes passés d'un niveau d'intégration de 2 μ m dans les premières familles XC2000 à 16nm pour les dernières familles de la série UltraScale (*KINTEX, VIRTEX, ZYNQ*).

Quant à la révolution, celle-ci concerne l'intégration des mémoires, des multiplieurs, des processeurs, des circuits DSP dans un seul circuit FPGA. Cette évolution dans la structure des FPGAs a bien évidemment été accompagnée par une évolution des outils de développement.

Il fallait en effet des outils de plus en plus performants pour tirer profit de ces structures qui devenaient à la fois de plus en plus grandes, mais aussi de plus en plus hétérogènes.

La firme Xilinx propose un large éventail de circuits FPGA pour diverses applications.

L'évolution technologique de ces circuits est illustrée dans le *Tableau.6.1*.

Tableau.6.1. Evolution des circuits FPGA de Xilinx.

Gamme(Génération)	Année	Taux d'intégration (Porte logique)	Caractéristiques
XC 2000	1985	600 ~ 1500 (2 µm)	
XC3000,	1987	1000 ~ 6000 (1.2 et 0.8 µm)	
XC4000	1991	2000 ~500000 (0,8 µm)	
XC5200	1995	6000~18000 (0,5 µm)	
XC6000	1996	10.000 ~100.000	
SPARTAN	1998		
VIRTEX	1998	0.5 ~1 millions	Blocs RAMs, horloges
VIRTEX-E	1999	0.5~4 millions	RAMs, multiplieurs
VIRTEX-II	2000	0.4~ 8 millions	plus de RAMs, Multiplieurs
VIRTEX-II-PRO	2002		RAM, multiplieurs, processeurs Microblaze, POWERPC et IBM
VIRTEX-4	2004	90nm	RAM, multiplieur, DSP, LUT 4 entrée
VIRTEX-5	2006	65nm	LUT 6 entrées
VIRTEX-6	2009	40nm	LUT 6 entrées
SERIE-7 (ARTIX, KINTEX,VIRTEX-7)	2010	28nm	processeur ARM CORTEX-9
SERIE UltraScale et UltraScale+(KINTEX, VIRTEX ,ZYNQ)	2013- 2016	16nm et 20nm	Consommation d'énergies faible

Ainsi, comme l'illustre le Tableau.6.1, au jour d'aujourd'hui, avec l'avancée technologique, les circuits FPGA de la *Série 7*, tel que *Artix*, *Kintex* et *Virtex7*, ont vu le jour pour remédier au problème des limitations technologiques [147]. Ces derniers, issus respectivement d'une technologie 28, 20 et 16 nm basse consommation, offrent un compromis entre la haute performance et la grande capacité d'intégration.

La figure.6.5, illustre une coupe transversale de l'architecture des circuits de la *Série7*. Cette dernière englobe quatre « dies » déposés sur un interposeur de silicium passif (*Passif silicon interposer*). Chacun d'eux permet l'implémentation de 500.000 blocs logiques ainsi que d'autres ressources telles que les RAM, les DSP et les I/Os.

L'interposeur fournit des dizaines de milliers de connexions entre die (die à die) permettant une interconnexion ultra haute avec une faible consommation d'énergie. Ces connexions sont assurées au travers de micro billes (Microbumps) issus des différents dies. Des TSV (Through Silicon Via) sont combinés avec des perles de soudure (C4 Bumps) permettant de monter le circuit FPGA sur un substrat haute performance [148].

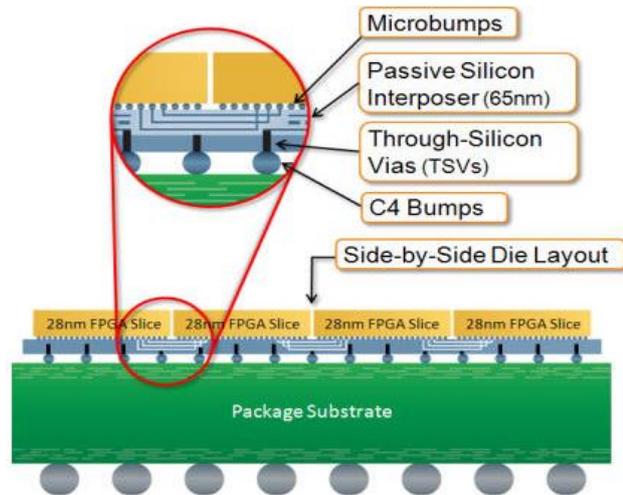


Figure.6.5. Vue transversale de l'architecture du circuit *Virtex7* [148].

Comme illustré en Figure.6.6, la nouvelle architecture interne des circuits FPGA de la *Série 7* de Xilinx est organisée en colonne contrairement aux précédentes architectures, qui elles étaient organisées en matrice (Figure.6.2).

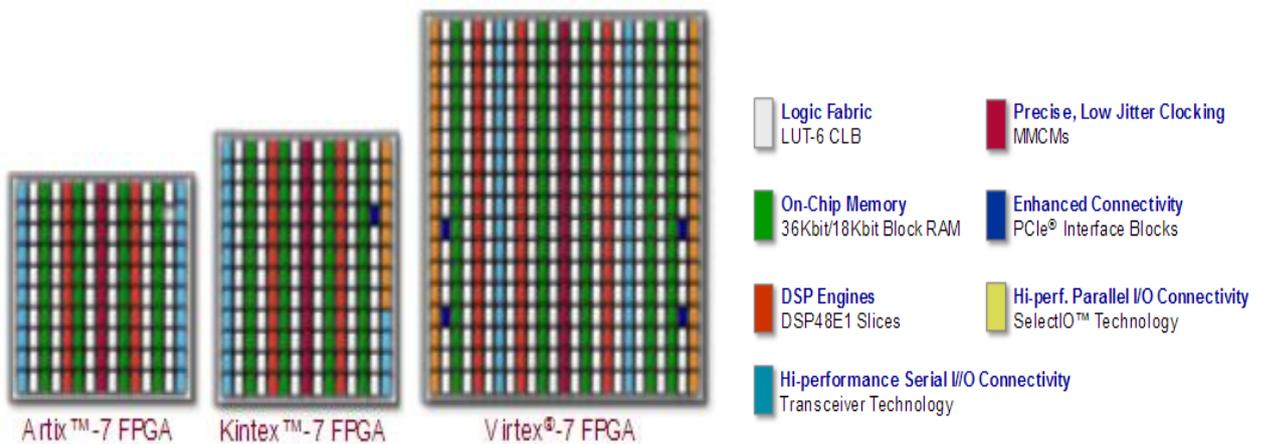


Figure.6.6. Architecture en colonne des circuits FPGA *Série7* de Xilinx [149].

L'architecture interne d'un Bloc Logique Configurable (*CLB*) du circuit FPGA *Virtex Série7* est illustrée par la Figure.6.7.

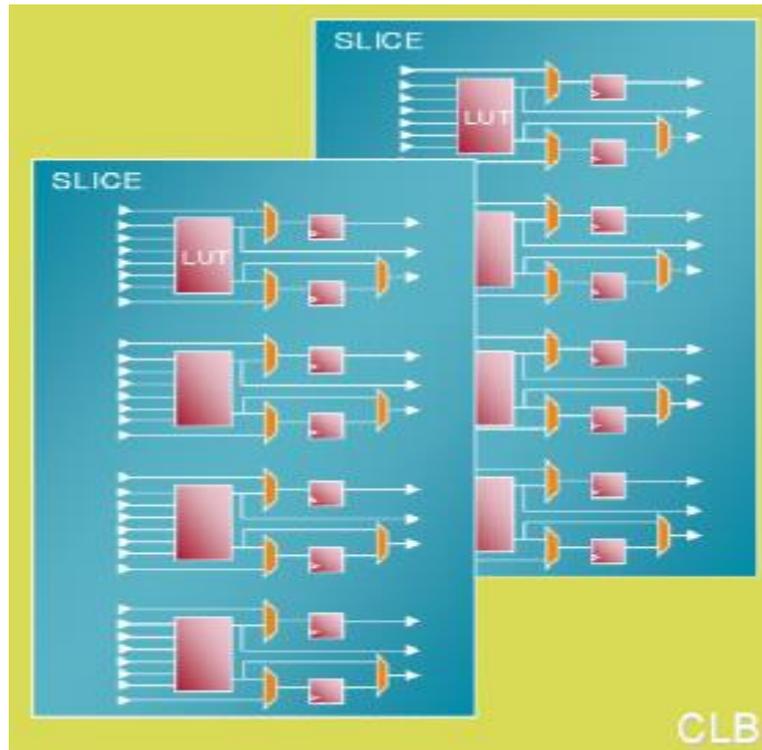


Figure.6.7. Architecture interne d'un CLB du circuit FPGA *Virtex Série7* de Xilinx [150].

Les CLBs se composent de deux parties : des bascules (flip-flop) et des LUT (Look Up Table). Chaque CLB englobe :

- Quatre LUTs à six entrées par slice ;
- Deux flip-flops par LUT ;
- Deux slices par CLB.

De plus, l'architecture des circuits de la *Série 7* contient :

- Des cœurs de processeur DSP appartenant à la famille DSP48E1. Ces derniers sont configurables, performants et de petites tailles permettant de garder la flexibilité du design. Ils intègrent entre autres, un bloc multiplieur dédié à deux entrées 25*18 bits qui opèrent à des fréquences relativement élevées, un bloc MAC multiplieur accumulateur de 48 bits, une ALU (Arithmetic Logic Unit). La figure.6.8 illustre l'architecture du DSP de la série 7 ;
- Des blocs de mémoire BlockRAM qui varient selon l'architecture, configurables, permettant des accès en lecture et écriture. Ces derniers sont distribués sur tout le circuit.

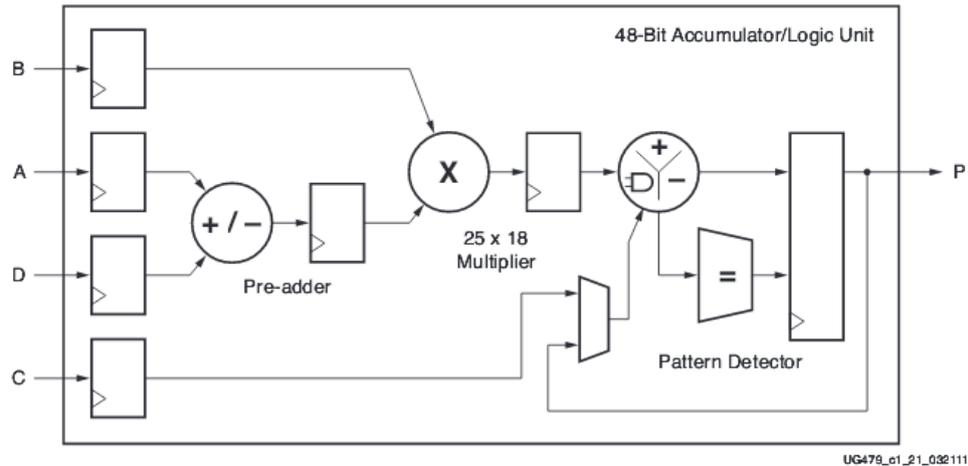


Figure.6.8. Architecture interne du DSP48E1 du circuit FPGA Virtex Série7 de Xilinx [151].

Ainsi, dans le cadre de nos travaux de recherche, pour l'implémentation des contrôleurs MPPT bio inspirés, nous avons ciblé la carte de prototypage *Virtex-7 FPGA VC707 Evaluation Kit* [152] conçu autour du circuit *Virtex7 xc7vx485t-2ffg1761c*.

6.3 IMPLÉMENTATION DES CONTROLEURS MPPT SUR CIRCUIT FPGA : ÉTAT DE L'ART

Plusieurs travaux liés à l'implémentation hardware des contrôleurs MPPT ont été proposés dans la littérature. Cependant, ces contrôleurs MPPT étaient implémentés traditionnellement, sur des microcontrôleurs et des DPSs [153-155]. Combien même ces contrôleurs MPPT ainsi implémentés atteignaient des performances raisonnables, il n'en demeure pas moins que l'implémentation de ces derniers sur des circuits logiques programmables tel que les FPGA offrent, de meilleurs performances [156].

De plus, comme le domaine de la microélectronique n'a cessé d'évoluer, et ce, de manière continue, particulièrement ces dernières années, les performances qu'offrent les FPGA sont en perpétuelles évolution (Tableau.6.1)

De ce fait, il est à noter que l'atout majeur des implémentations sur circuits FPGA par rapport à celle sur DSP, qui, elles, s'exécutent de manière séquentielle, réside justement dans le fait que le circuit FPGA, s'exécute en mode parallèle ce qui explique l'importance de leurs capacités.

En outre, en tenant compte du volet économique, les implémentations sur circuit FPGA sont peu coûteuses étant donné que l'ensemble des fonctions électroniques peut être implémenté sur une même puce.

Par conséquent, de par les avantages qu'offrent les circuits FPGA de nos jours, ces derniers sont devenus des éléments incontournables pour la conception de circuits numériques [157]. Leur apport au niveau de la phase de prototypage des systèmes en cours de conception est indéniable [158]. Ces derniers sont utilisés dans plusieurs domaines d'applications telles que le traitement de signal, la cryptographie, la bio-informatique, le biomédical, le véhicule électrique, les applications spatiales,...etc.

Néanmoins, le domaine des énergies renouvelables, plus précisément les systèmes photovoltaïques, est, sans conteste, l'un des domaines d'application les plus convoités ces dernières années. En effet, l'implémentation des contrôleurs MPPT sur circuit FPGA est devenue une évidence même pour l'optimisation de puissance dans les systèmes photovoltaïques. Ceci est dû, vraisemblablement, aux aspects liés aux grandes performances de ces dernières et surtout à l'aspect relatif aux coûts induits.

De plus, sur un plan technique, le système de contrôle ainsi conçu, est aisément modifiable et mis à jour, même s'il est en cours d'exécution. Les contrôleurs MPPT à base de méthodes conventionnelles (MC) ont été les premiers à être implémentés sur circuits FPGA. A titre d'exemple, on pourra citer les contrôleurs Perturber & Observer (P&O) en raison de leur facilité d'implémentation [159-160]. Par la suite, cette manière de faire, en l'occurrence, l'implémentation des contrôleurs MPPT sur circuit FPGA a été généralisée et appliquée aux autres méthodes [161].

Dans [162-166], les auteurs ont établi l'apport avéré des circuits FPGA pour la commande MPPT et démontré les performances des contrôleurs MPPT basées sur les méthodes d'intelligence artificielles (MIA). Le tableau.6.2. résume une synthèse non exhaustive des différents travaux de recherche répertoriés dans le cadre de l'implémentation de contrôleurs MPPT sur circuits FPGA.

Tableau.6.2. Synthèse des travaux relatifs à l'implémentation des MPPT sur circuit FPGA.

Auteurs	Référence	
N. Khaehintung et al.	[159] 2006	Les auteurs proposent l'implémentation d'un contrôleur P&O amélioré sur circuit FPGA. Le circuit est décrit en VHDL et implémenté sur un circuit FPGA XC2C384 de Xilinx. Les résultats expérimentaux montrent que l'algorithme proposé surpasse le contrôleur classique en termes de temps de réponse, d'atténuation des oscillations. L'efficacité globale du système est supérieure à 96%.
Y. Elbasri et al.	[160] 2010	Les auteurs proposent la mise en œuvre d'un contrôleur MPPT P&O sur circuit FPGA. Cette stratégie de contrôle a été validée en premier par simulations. Par la suite, une validation au travers d'une simulation en condition réelle à l'aide d'un panneau solaire BP585 et un convertisseur DC-DC Boost est réalisée.
Chen et al.	[162] 2010	Les auteurs proposent l'implémentation d'un contrôleur MPPT à base de logique floue. La méthode est simple, fiable et peut s'ajuster automatiquement aux changements climatiques. Le contrôleur MPPT flou est décrit en Verilog et implémenté sur circuit FPGA P2C8Q208C8 d'Altera.
Dzung et al.	[163] 2010	Les auteurs proposent un contrôleur basé sur une méthode hybride : les réseaux de neurones ANN et l'augmentation de la conductance Inc Cnd. L'efficacité de l'algorithme proposé est validée par simulation à l'aide de Matlab Simulink. Le contrôleur MPPT est implémenté sur circuit FPGA VirtexII pro de Xilinx.
Messai et al.	[164] 2010	Les auteurs proposent un contrôleur MPPT basé sur une méthode hybride : la logique floue combinée aux algorithmes génétiques. Le contrôleur MPPT est décrit en VHDL et implémenté sur circuit FPGA Virtex II (XC2v1000-4fg456 de Xilinx).
Messai et al.	[165] 2011	Les auteurs proposent l'implémentation hardware d'un contrôleur MPPT à base de logique floue sur circuit FPGA de Xilinx. Le contrôleur est décrit en VHDL. Les résultats de simulation obtenus avec ModelSim montrent des performances satisfaisantes.
Chekired, C.Larbes et al.	[166] 2011	Les auteurs proposent une méthode de recherche du PPM basée sur un contrôleur flou (FLC). Le contrôleur est décrit en VHDL. Les résultats de simulation obtenus avec ModelSim montrent des performances satisfaisantes. L'implémentation du contrôleur MPPT est faite sur un circuit FPGA Virtex II de la carte Memec V2MB1000 de Xilinx.
Cheng et al.	[167] 2011	Les auteurs proposent l'implémentation d'un contrôleur flou adaptatif sur circuit Cyclone II EP2C8Q208C8 d'Altera. Les étapes d'implémentation sont décrites en détails. Les tests expérimentaux montrent que la méthode présentée donne de bonnes performances en termes de temps de réponse, de précision et stabilité.
Chekired, C.Larbes et al.	[168] 2011	Les auteurs proposent l'implémentation hardware de deux contrôleurs MPPT : neuro flou et logique floue. Les contrôleurs sont décrits en VHDL, évalués et comparés par une simulation en temps réel. Les résultats de simulation obtenus avec ModelSim montrent des performances satisfaisantes. L'implémentation du contrôleur MPPT est faite sur un circuit FPGA Virtex II de la carte Memec V2MB1000 de Xilinx.

Chekired, C.Larbes et al.	[169] 2012	Les auteurs proposent une étude comparative pour l'implémentation hardware sur circuit FPGA de deux contrôleurs MPPT intelligents.
N. Chettib et al.	[170] 2012	Les auteurs proposent une implémentation du contrôleur Inc Cnd sur circuit FPGA. Le contrôleur est décrit en VHDL et mis en œuvre sur circuit Virtex-II-Pro xc2v1000-4fg456 de Xilinx. Les résultats de simulation confirment l'efficacité du contrôleur (96%) et le temps de réponse rapide (2 ms).
Basil et al	[171] 2012	Les auteurs proposent l'implémentation sur circuit FPGA d'un contrôleur MPPT à base de logique floue. L'implémentation est faite sur circuit Spartan 3AN de Xilinx. Une étude comparative avec le contrôleur conventionnel P&O est faite, les résultats montrent que le contrôleur FLC possède de meilleures performances.
K.Punitha et al.	[172] 2013	Les auteurs proposent un contrôleur basé sur une méthode hybride : la logique floue combinée incremental conductance pour la poursuite du PPM sous ombrage partiel et variation atmosphérique. Les résultats de la simulation montrent que le contrôleur présente de bonnes performances. L'implémentation hardware sur un circuit FPGA Spartan 3 ^E de Xilinx est également présentée. Les résultats expérimentaux montrent une performance satisfaisante des approches proposées.
K.Punitha et al.	[173] 2014	Les auteurs présentent l'implémentation d'un contrôleur hybride : réseaux de neurone combiné à l'incremental conductance. Le système est simulé et évalué sous conditions d'irradiation uniformes et non uniformes. Le contrôleur MPPT est décrit en VHDL et implémenté sur circuit FPGA d'Altera.
Chekired et al.	[174] 2014	Les auteurs proposent l'implémentation de quatre contrôleurs intelligents pour l'optimisation de puissance d'un SPV. L'efficacité de ces méthodes a été évaluée pour différents scénarios de simulation sous Matlab / Simulink. Les contrôleurs sont décrits en VHDL et mis en œuvre sur la carte FPGA Virtex5 de Xilinx.
S.Titri et al.	[175,176] 2014-2015	Les auteurs présentent une nouvelle méthodologie de conception basée sur le Model Based Design pour le prototypage rapide de contrôleurs. L'outil HDL Coder de Matlab est utilisé. La méthodologie de conception a été appliquée au contrôleur P&O, implémenté sur circuit FPGA Spartan 3 ^E de Xilinx.
H.Abbes	[177] 2016	Les auteurs présentent l'implémentation sur circuit FPGA d'un contrôleur MPPT à base de logique floue pour contrôler un système photovoltaïque. Le contrôleur est décrit en VHDL et mis en œuvre sur la carte FPGA Cyclone II 2C35.
K.Kaced, C.Larbes	[178] 2017	Les auteurs proposent une implémentation sur circuit FPGA d'un contrôleur pour le suivi du point de puissance maximale sous ombrage partiel. Le contrôleur proposé est à base d'algorithme de chauve-souris. Une étude comparative avec P&O et PSO est effectuée.
A.Youssef, M.El Telbany	[179] 2018	Les auteurs proposent une implémentation sur circuit FPGA reconfigurable pour le suivi du point de puissance maximale dans les systèmes photovoltaïques. Le contrôleur proposé est à base de logique floue et opère dans des conditions d'éclairement et de température variables. Ce dernier décrit en VHDL est implémenté sur un circuit programmable Spartan6.

Discussion :

Au travers de cette synthèse, nous pouvons dire, que tous les contrôleurs, qu'ils soient issus des méthodes conventionnelles (MC), intelligentes (MIA) ou bio inspiré (MBI), peuvent être implémentés sur des circuits FPGA.

Cependant, la simplicité d'implémentation, la surface et la zone mémoire des circuits FPGA requis, dépendent principalement de la structure finale du contrôleur MPPT à développer. Généralement, les circuits Spartan-3^E et Virtex II sont suffisants pour mettre en œuvre les contrôleurs issus des MC et MIA.

Pour exemple, l'implémentation hardware du contrôleur P&O de par sa simplicité, n'exploite que quelques ressources du circuit Spartan 3^E de Xilinx. Cependant, son principal inconvénient réside dans son manque de précision résultant des multiples oscillations provoquant des pertes de puissance.

Concernant l'implémentation hardware des MIA sur circuit FPGA, des efforts supplémentaires et des connaissances approfondies sur la configuration et l'implémentation des circuits FPGA sont indispensables pour l'optimisation en termes d'occupation (surface) des ressources des circuits FPGA. En effet, pour un contrôleur MPPT à base de réseau de neurones, si le nombre de couches et de neurones est grand, ce dernier influe sur la surface d'occupation du circuit FPGA. De plus, pour un contrôleur à base de logique floue FL-MPPT, ce dernier nécessite plus d'espace mémoire. Cependant, la difficulté du calcul flou réside dans la bonne conception des règles floues et des fonctions d'appartenance qui dépendent fortement de la connaissance préalable du système.

Le contrôleur hybride GA-FL-MPPT, a pour sa part, un inconvénient relatif à la complexité de son implémentation. En effet, l'implémentation du contrôleur nécessite une programmation lourde en VHDL.

Pour ce qui est des contrôleurs issus des MBI, en prenant comme exemple les contrôleurs à base d'algorithmes *PSO*, *DE*, *ABC*, etc..., ces derniers présentent un parallélisme inhérent, de ce fait, les circuits FPGA sont bien adaptés pour une implémentation hardware.

Ainsi, les principales caractéristiques des circuits FPGA, cités dans la section **6.2.2**, sont mises à profit à savoir :

- Leur mode d'exécution parallèle qui en fait leur force majeure ;
- Leur mode reconfigurable qui leur permet de s'adapter aux modifications en vue d'une amélioration fonctionnelle ;
- Leur mode de prototypage rapide qui permet à partir d'une simple idée ou concept, de concevoir, d'implémenter, de tester en temps réel, ...etc. réduisant ainsi le temps de mise sur le marché.

Au terme de cette synthèse, nous pouvons attester que l'apport des circuits FPGA pour la commande MPPT est justifié.

6.4 Conclusion

Au cours de ce chapitre, nous avons présenté les circuits FPGA. Ces derniers sont sélectionnés comme cible principale pour l'implémentation hardware des contrôleurs MPPT bio inspirés développés aux chapitres précédents.

Par la suite, nous avons donné un état de l'art sur l'implémentation hardware des contrôleurs MPPT sur circuit FPGA, à l'issue duquel, l'apport des circuits FPGA pour le control MPPT a été démontré et leur choix justifié.

Néanmoins, il est à retenir que l'implémentation des algorithmes de contrôle pour la commande MPPT sur circuits FPGA est une démarche qui nécessite une parfaite maîtrise des processus de conception, de langages de description hardware (HDL) et un travail spécifique d'adéquation entre l'algorithme et l'architecture de commande à intégrer. De ce fait, un savoir-faire méthodologique est nécessaire aux concepteurs utilisant les circuits FPGA afin de satisfaire l'ensemble des contraintes inhérentes de l'implémentation, tout en apportant une flexibilité de développement suffisante. C'est ce qui limite depuis longtemps l'utilisation des FPGAs aux personnes issues de différents milieux. C'est dans ce contexte, qu'au prochain chapitre, nous proposerons une nouvelle méthodologie de conception, basée sur la synthèse de haut niveau facilitant le développement et le prototypage rapide de contrôleurs MPPT.

Chapitre 7

**PROTOTYPAGE RAPIDE DE
CONTRÔLEUR BIO INSPIRÉ SUR CIRCUIT
FPGA**

7.1 INTRODUCTION

L'objectif de ce chapitre est de présenter une nouvelle méthodologie de conception qui vise à faciliter le prototypage rapide de contrôleurs MPPT sur des circuits FPGA. A cet effet, l'évolution des méthodologies depuis les approches de conception au niveau RTL jusqu'à la conception de synthèse de haut niveau (HLS) sur cible FPGA sera présentée. Ensuite, nous présenterons, dans le détail, de la méthodologie adoptée pour le prototypage rapide des algorithmes bio inspirés développés précédemment pour le contrôle MPPT.

7.2 METHODOLOGIES DE CONCEPTION HARDWARE : DU DOMAINE ALGORITHMIQUE JUSQU'A LA GENERATION D'IP

Communément, un algorithme se présente comme une description qui peut être sous forme : de formules mathématiques, de formes textuelles et/ou graphiques, de diagrammes de séquences,...etc. Les concepteurs d'algorithmes dits « concepteurs software » ont généralement tendance à penser seulement dans un cadre mathématique dans lequel l'algorithme symbolise une suite de processus codifiés destinés à une implémentation logicielle. Il s'avère d'un autre côté, que l'implémentation de ces algorithmes sur des cibles numériques (implémentation physique) n'est quelque fois pas facile pour les designers dits « concepteurs hardware », en charge de cette implémentation, surtout lorsque la complexité de ces derniers augmente.

Cependant, dans une optique de concrétisation d'une implémentation effective, la collaboration étroite entre ces deux communautés de concepteurs durant toutes les phases de conception, est primordiale.

Par ailleurs, ces concepteurs ont très tôt admis la nécessité d'adopter des méthodologies de conception et d'utiliser des outils de développement dédiés, qui facilitent en quelque sorte le passage de l'algorithme à l'architecture, à moindre coût, tout en réduisant le temps de conception.

7.2.1 Evolution des méthodologies de conception

Généralement, les méthodologies de conception s'appuient sur des approches très structurées. Elles sont basées sur la validation de représentations par simulation, à différents niveaux. Elles utilisent pour cela des bibliothèques de composants dans un environnement de conception assisté (CAO).

Le digramme de Gajski et Kuhn [180], illustré en Figure 7.1 ci-dessous, permet d'introduire et d'expliquer les différentes méthodologies de conception. Ce dernier donne une vision des domaines de conception et des différents niveaux d'abstraction illustrant comment un *algorithme/circuit* peut se présenter pendant le processus de conception.

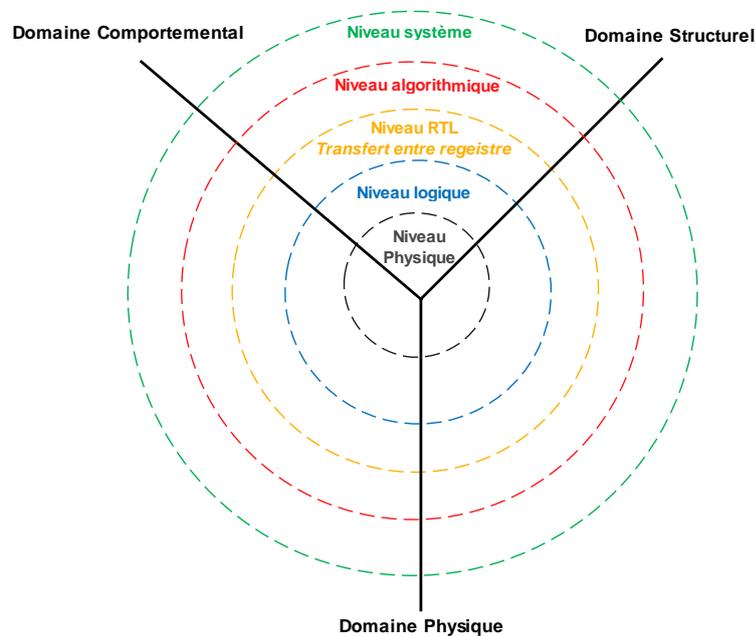


Figure.7.1 Diagramme de Gajski et Kuhn : domaine et niveaux d'abstraction [180].

La représentation des domaines de conception est décrite en utilisant trois axes, chacun représentant un domaine de description à savoir le *comportemental*, le *structurel* et le *physique*.

- La conception au niveau *comportemental* décrit la fonctionnalité du circuit, idéalement sans aucune référence à la façon dont le comportement du circuit est réalisé par une implémentation ;
- La conception au niveau *structurel* décrit l'implémentation structurée de la conception avec une hiérarchie des composants et de leurs interconnexions ;
- La conception au niveau *physique* décrit l'implémentation physique de la conception.

Comme l'illustre la Figure.7.2 (a), les cercles concentriques du diagramme en Y de Gajski et Kuhn représentent les niveaux d'abstraction. Ainsi, la description d'un algorithme/circuit peut être élaborée à différents niveaux d'abstraction selon les domaines de conception cités précédemment.

Il est clair que le niveau de détail augmente de l'extérieur vers l'intérieur.

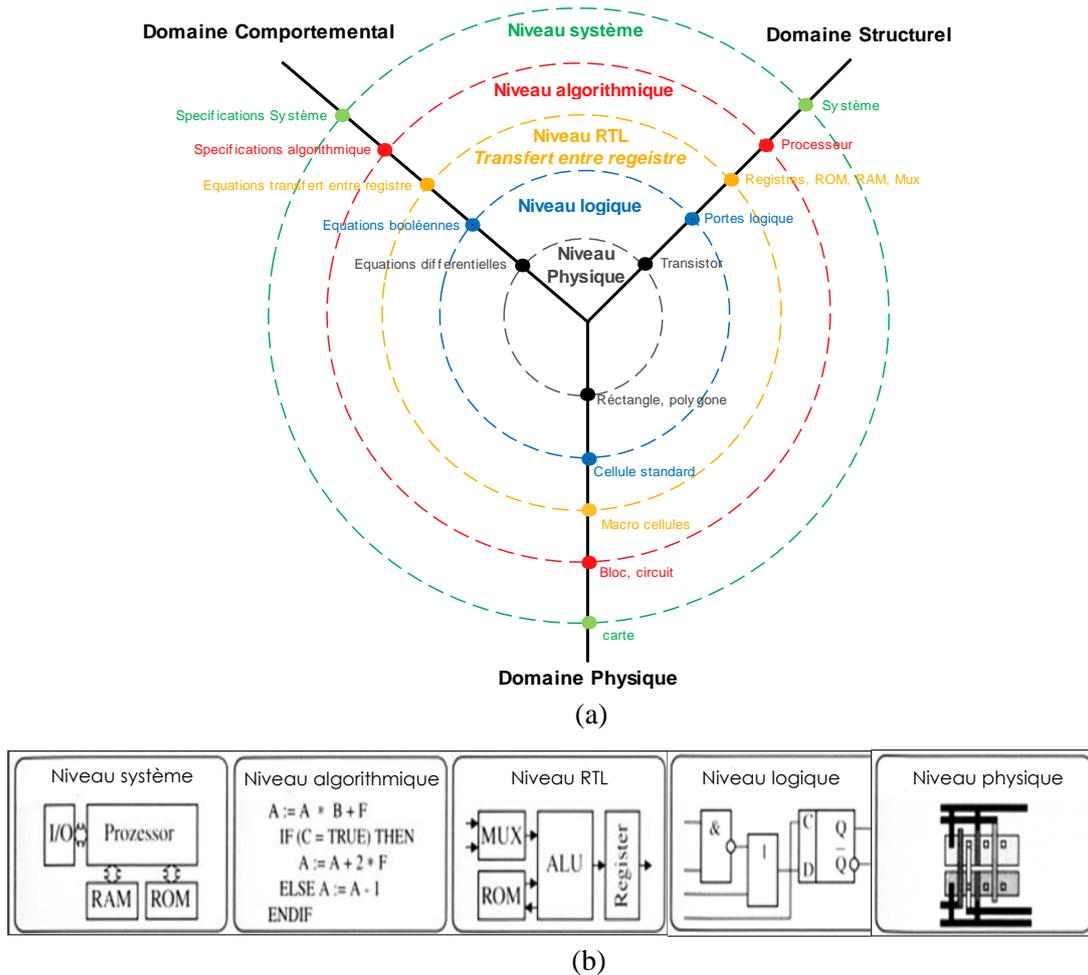


Figure.7.2. Diagramme de Gajski et Kuhn : (a) niveaux d'abstraction ;
(b) principaux composants aux différents niveaux d'abstraction.

Comme l'illustre la figure ci-dessus :

- Le niveau *système* représente le plus haut niveau d'abstraction pour lequel un système complexe est décrit en termes de processus communicants au moyen de langages tels C, C++, SystemC, SystemVerilog, ... etc ;
- Le niveau *algorithmique* est représenté par une modélisation comportementale avec des langages de programmation de haut niveau ;
- Le niveau *transfert de registre RTL* est représenté par une description de transfert de données entre des registres et des opérations internes arithmétiques et logiques (combinatoires) à réaliser sur ces données ;
- Le niveau *logique* représente le comportement des composants RTL avec un ensemble de portes logiques et de bascules ;

- Le niveau *physique/circuit* est représenté par une implémentation du comportement des portes logiques avec des transistors interconnectés.

Les principaux composants pour les différents niveaux d'abstraction sont représentés graphiquement dans la figure.7.2 (b).

Dans le diagramme en Y Gajski et Kuhn, la méthodologie de conception apparaît comme étant l'intersection entre les axes du domaine de conception et les cercles des différents niveaux d'abstraction.

- ***Méthodologie de conception ascendante « Bottom-Up »***

Comme son nom l'indique, cette méthodologie part du niveau d'abstraction bas vers celui du haut. La méthodologie consiste à consolider et à opérer une synthèse à partir des éléments de base initialement identifiés. En d'autres termes, elle privilégie la conception et la validation des modules élémentaires avant ceux du niveau hiérarchique plus haut. Ainsi, chaque niveau génère des bibliothèques pour le niveau d'abstraction supérieur. L'avantage de cette méthodologie réside dans le fait que les niveaux d'abstraction sont clairement séparés et que chacun possède sa propre bibliothèque.

- ***Méthodologie de conception descendante « Top-Down »***

Contrairement à la méthodologie ascendante, cette méthodologie commence par le plus haut niveau d'abstraction pour passer de la description fonctionnelle du système à la structure de celui-ci. Ainsi, le système est construit comme une hiérarchie d'objets, où les détails de réalisation se précisent au fur et à mesure que l'on descend dans cette hiérarchie.

Traditionnellement, la conception et l'implémentation d'une architecture « *algorithme* » sur circuit FPGA passe par des méthodologies de conception (approches) top-down dites *classiques*, très proches de la conception de circuits numériques dédiés ASIC (*Applied Specific Integrated Circuit*).

Cependant, la méthodologie de conception classique, qui passe depuis les spécifications algorithmiques jusqu'à arriver à la génération et l'implémentation du fichier bitstream (.bit), a vu émerger de nouvelles approches de conception comme l'illustre la Figure.7.3.

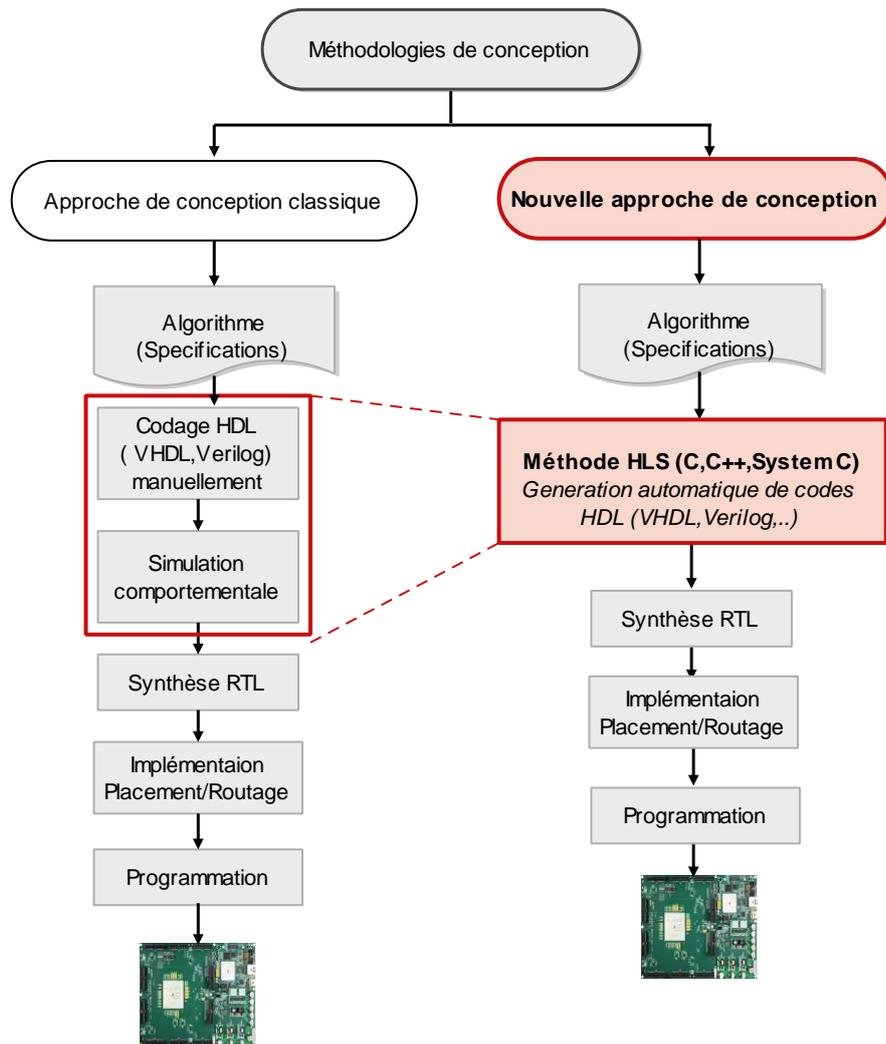


Figure.7.3. Les approches de conception.

Ces nouvelles approches de conception ont vu le jour afin de raccourcir la durée de conception et de palier à la difficulté croissante de programmation des architectures complexes avec des langages HDL (Hardware Description Language).

Dans ce qui suit, nous allons donner un aperçu de chacune de ces différentes approches existantes en présentant les limites des approches classiques et les avantages des nouvelles approches.

7.2.1.1 L'approche de conception classique RTL et ses limites

L'approche de conception classique consiste à décrire le système à concevoir au niveau RTL (Register Transfer Level) dans un langage de description matérielle (Hardware Description

Langage HDL), comme VHDL [182] ou Verilog [183]. Le flot de conception, dont les étapes seront détaillées par la suite, est illustré dans la Figure 7.4. Ce dernier, doit être rigoureusement respecté afin de pouvoir passer de l'écriture du code source (VHDL, Verilog) à son exécution sur une plateforme cible FPGA.

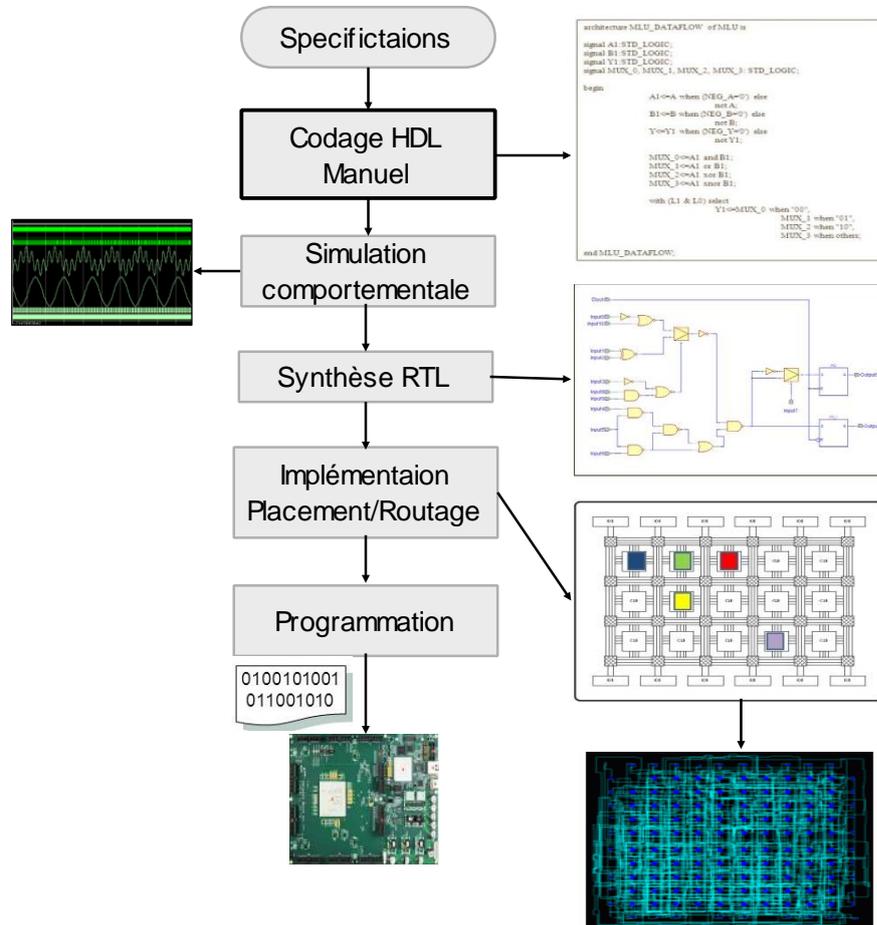


Figure.7.4. Flot de conception classique.

Les différentes étapes du flot de conception classique sont comme suit :

– **Description HDL**

La première tâche du concepteur consiste à représenter fidèlement les spécifications algorithmiques permettant de définir l'architecture par une démarche d'adéquation Algorithme-Architecture. Cette architecture est alors décrite en langage matériel (HDL).

– **Synthèse Register Transfer Level (RTL)**

Comme première étape, l'outil génère la synthèse qui traduit la transcription de la description **HDL** introduite par le concepteur en une représentation au niveau **RTL** (Register Transfer

Level). Ceci se fait par la génération, dans un premier lieu, des éléments de base tels que les bascules, les LUTs et les machines d'états. Puis, à partir du code source, il extrait les macros correspondantes à des comportements ou des opérations spécifiques comme celles relatives à la mémoire RAM, aux registres à décalage, aux multiplexeurs,...etc., et il les remplace par des composants propres du constructeur.

Suite à cela, l'outil opère un regroupement des ressources combinatoires et séquentielles pour minimiser le nombre de blocs logiques utilisés. Dès que le processus de synthèse est finalisé, une simulation fonctionnelle de l'algorithme ainsi décrit est déjà possible pour vérifier le comportement du système par rapport aux fonctionnalités attendues. Dans le cas contraire, si les spécifications ne sont pas atteintes, il est possible d'opérer les modifications nécessaires en modifiant le code afin d'y remédier.

– ***Placement et Routage***

A partir des fichiers de synthèse, l'outil transforme en premier la représentation RTL en une représentation au niveau « portes logiques », puis procède, par la suite, au placement et routage sur le circuit FPGA ciblé.

– ***Génération du bitstream et programmation du FPGA.***

Il s'agit de la dernière étape qui permet de générer, à partir de la représentation obtenue, un fichier de configuration « bitstream » qui servira à programmer les connexions internes et celles des blocs logiques du circuit FPGA.

Ainsi, le flot permet le passage d'une description HDL au niveau RTL jusqu'à arriver à la génération du fichier « bitstream » pour la programmation du circuit.

Il est utile de préciser que lors de la phase de conception de l'algorithme, le choix du type de description ainsi que l'ensemble des paramètres de l'outil de synthèse agissent fortement sur les performances de l'application au final.

Selon la littérature, une conception basée sur les langages de description HDL tels que le VHDL et Verilog, donne de meilleurs résultats en termes de ressources et temps de réponse. Néanmoins, cela suppose de bonnes connaissances en conception hardware, car il est question de transcrire l'algorithme original en une description HDL synthétisable. De plus, à cause du niveau de détail très élevé de ces langages, la transcription de l'algorithme en langage HDL

nécessiterait un temps alloué relativement important. Cette tâche de transcription va s'avérer d'autant plus laborieuse dans le cas d'algorithmes complexes, tant en termes de moyens humains que de temps.

Pour répondre aux contraintes de temps et de complexité de développement induits par l'usage des langages HDLs, de nombreux travaux ont été menés afin de relever le niveau d'abstraction pour la spécification de l'algorithme. L'objectif étant de proposer de nouveaux outils de génération automatique de description RTL à partir de spécifications algorithmiques dans des langages utilisant des niveaux d'abstraction plus hauts tels C, C++, SystemC. On parle alors de *synthèse de haut niveau* pour « *High Level Synthesis, (HLS)* » [184].

Ainsi, l'implémentation sur des circuits FPGA devient alors plus accessible, ne nécessitant pas une expertise accrue en conception hardware, ni même des connaissances approfondies des langages HDL (VHDL, Verilog).

7.2.1.2 L'approche de conception HLS et ses avantages

Pour l'implémentation sur circuits FPGA, les évolutions technologiques dans les langages de haut niveau de synthèse HLS qu'a connus le monde ces dernières années, ont permis de remplacer la conception au niveau RTL, devenue caduque. Pour cela, l'utilisation des outils HLS pour la programmation sur circuit FPGA, a permis de décrire le circuit avec un langage de spécification de haut niveau au lieu de passer par la méthode classique RTL.

Cette évolution permet d'élever le niveau d'abstraction du design et d'en contrôler la complexité. Ceci va réduire les coûts, toujours croissants, de la vérification fonctionnelle visant à améliorer la productivité (Time To Market).

Dans [185], les auteurs attestent que le codage avec la méthode HLS permettrait de réduire de 7 à 10 fois le nombre de lignes de code nécessaires comparativement avec la méthode RTL. De plus, cela réduirait de 10 à 30% l'utilisation des ressources FPGA [186].

La Figure.7.5, illustre le flot de conception au niveau HLS. Ce dernier, permet d'automatiser la génération de la description d'une architecture au niveau RTL.

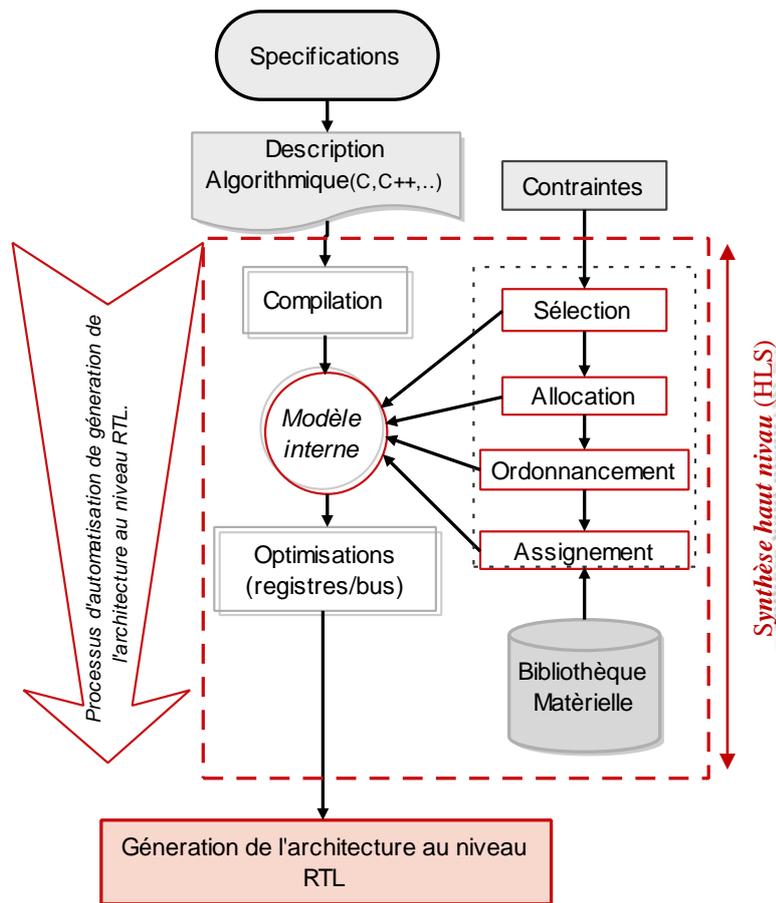


Figure.7.5. Flot de conception HLS.

La description de la fonctionnalité du circuit à concevoir est donnée sous forme d'un algorithme. Il s'agit d'une description de haut niveau purement fonctionnelle, qui utilise un jeu de contraintes spécifiées par le concepteur afin de générer automatiquement une description RTL. C'est les fonctionnalités de l'outil HLS qui se chargeront de l'exécution des tâches complexes et répétitives liées à l'écriture de la description de l'architecture au niveau RTL. En d'autres termes, l'outil HLS accepte comme entrées des descriptions comportementales et produit en sortie une description RTL.

Les différentes tâches sont comme suit :

- **Description comportementale** en un langage de haut niveau : C, C++, SystemC ...
- **Compilation et optimisation de la description algorithmique**, c'est la première tâche dans le flot HLS. Cette dernière consiste à compiler la description C, C++, SystemC, ...etc., et à traduire la description comportementale en une représentation (modèle) interne à l'outil de synthèse. Au cours de cette étape, les transformations des compilateurs de langages de programmation sont effectuées. Ces transformations incluent la simplification des expressions, la propagation des constantes, ...etc ;

- **Contraintes** correspondent aux directives que l’outil devra prendre en considération lors des différentes étapes que l’outil devra suivre ;
- **Sélection des opérateurs à mettre en œuvre** : lors de cette phase, le type de ressources matérielles nécessaires aux opérations à effectuer est choisi à partir des bibliothèques d’opérateurs disponibles dans le flot. Les contraintes spécifiées (latence, surface et technologie) sont prises en compte dans ce choix ;
- **Allocation** détermine le nombre de ressources nécessaires pour l’exécution de la description (*nombre d’opérateurs de même type qui vont être utilisés à la même date*). Cette étape fixe le nombre et le type d’unités de stockage (mémoire, registre), d’unité d’interconnexion (bus, multiplexeurs, ...etc.), d’unités fonctionnelles (multiplieur, additionneur, etc...) ;
- **Ordonnement** consiste en l’affectation des opérations produites par la compilation, ainsi, il détermine l’instant d’exécution de toutes les instructions, tout en garantissant le respect des dépendances de données ;
- **Assignment** associe à chaque opération sa ressource matérielle correspondante ;
- **Génération de l’architecture** est la dernière étape dans le flot de synthèse de haut niveau. Elle consiste à générer une description au niveau RTL du circuit qui sera exploitée par les outils de synthèse de plus bas niveau.

Cette approche présente les avantages majeurs liés essentiellement à la réduction des délais :

- de conception : avec une rapidité de conception et un prototypage rapide où les algorithmes sont plus faciles et plus rapides à écrire dans des langages de haut-niveau ;
- de développement de nouveau *IP (Intellectual Property)*. Ce dernier, est deux à trois fois plus rapide qu’avec une description manuelle en RTL (HDL hand writing) ; De plus, les IP sont conçues pour être réutilisées (Design Reuse) ;
- de simulation et vérification : en diminuant le nombre d’itérations nécessaires pour obtenir un composant fonctionnel. En déchargeant le concepteur de la gestion des horloges, du partage des ressources et de l’interfaçage mémoire, en les allouant à l’outil, les risques d’erreurs sont réduits. En effet, le délai de simulation et de vérification du système HLS est au moins dix fois plus rapide que celui en RTL [187]. De plus, la synthèse HLS permet d’obtenir rapidement des estimations des caractéristiques du circuit (surface, puissance consommée, énergie consommée).

De ce fait, nous pouvons aisément attester que cette approche, basée sur la synthèse de haut niveau (*HLS*), s’inscrit dans le cadre du prototypage rapide. Nous pourrions même avancer que cette approche permet au concepteur de focaliser ses efforts sur l’objectif qu’il se fixe à atteindre sans se soucier des moyens qu’il devra employer pour y parvenir. En d’autres termes, elle permet de modéliser le design, sans avoir à se soucier des contraintes de la cible technologique sur lequel ce design sera implémenté ou du langage avec lequel il sera décrit. Elle améliore ainsi la portabilité entre plateformes et langages [188].

Comme l’illustre la Figure.7.6, il existe deux approches de conception au niveau HLS [189,190] :

- L’approche basée sur les langages *C-like* qui consiste à décrire l’algorithme avec des langages de programmation de niveau d’abstraction plus haut que les HDLs, tels que les langages *C*, *C++*, *SystemC*, *SpecC*, *Handel-C*, etc.;
- L’approche basée sur le modèle « **Model Based Design**, (MBD) », consiste en un ensemble d’outils avec des interfaces graphiques qui permettent de décrire le comportement et la structure d’un système numérique avec des modèles graphiques instanciés à partir des bibliothèques prédéfinies.

Dans cette catégorie, on trouve MATLAB Simulink HDL Coder [191], Xilinx System Generator [192], LabView [193], Altera DSP Builder [194], etc...

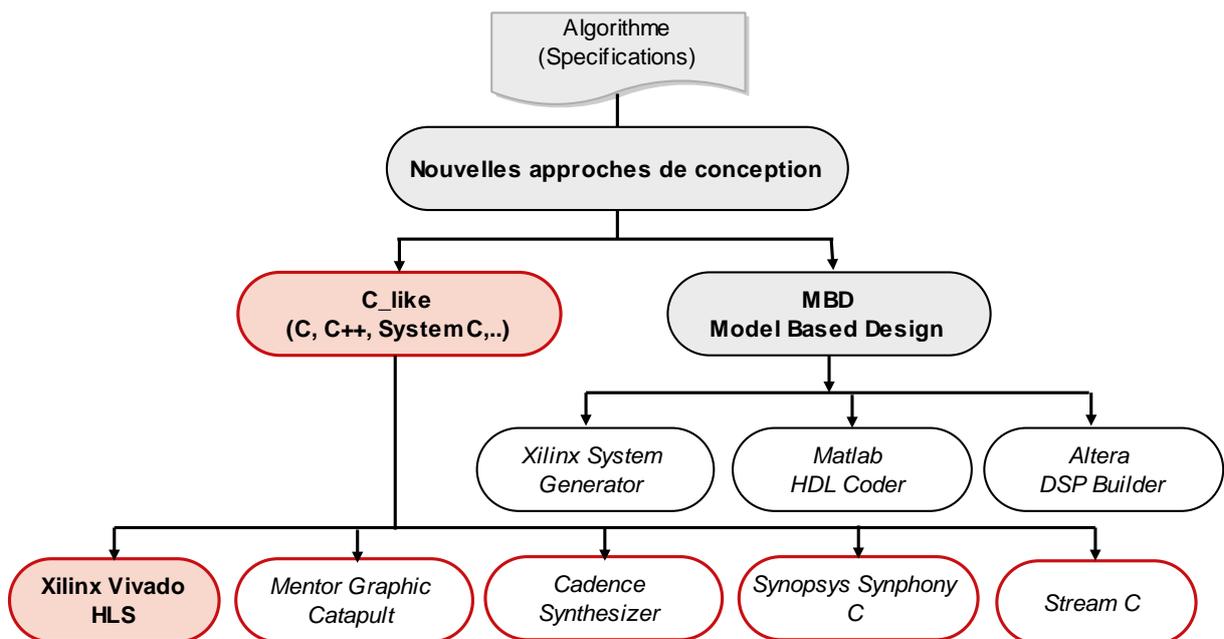


Figure.7.6 Nouvelles approches de conception.

Dans le cadre de nos travaux, nous nous intéressons à l'approche basée sur les langages *C-like*. Cette dernière fait appel à des outils de conception dédiés. C'est ce qui a permis durant la dernière décennie, à de nombreux outils académiques et commerciaux de voir le jour, dans le domaine de la synthèse de haut niveau. Dans ce qui suit, nous proposons un survol non exhaustif des principaux outils de synthèse de haut niveau existants :

- **Catapult** [195] est certainement l'un des outils industriels les plus répandus. Initié en 2004 par la société *Mentor Graphics*, cédé en 2011 à *Calypto Design Systems*, l'outil *Catapult* supporte *C++* et *SystemC* comme langages de description et génère du code *RTL* pour circuits *FPGA* ou *ASIC* ;
- **Synthesizer** [196] de *Cadence* supporte les descriptions *SystemC* et *C++*. Un large sous-ensemble de *System C/C++* est supporté pour la synthèse. Bien que *Cynthesizer* supporte des descriptions de comportement purement séquentielles, l'outil permet également l'application de directives afin de guider les étapes d'ordonnancement et l'allocation d'éléments de mémoire ;
- **Symphony C Compiler** [197] de *Synopsys* supporte les descriptions *SystemC*, *C* et *C++*. En 2010, *Synopsys* faisait l'acquisition du développeur d'outil de synthèse haut niveau *Synfora* pour renforcer sa position dans le domaine des outils de conception et de vérification de systèmes numérique ;
- **Stream-C** [198] a été proposé par le laboratoire de Los Alamos. Le langage est basé sur le modèle de programmation *Communicating Sequential Process (CSP)* de Hoare [197] et cible des applications opérant sur des flux de données ;
- **Vivado Xilinx HLS** [199] fait partie de la suite d'outil du fabricant *Xilinx*. L'outil utilise des directives de pré compilation afin de générer le code *RTL*. L'outil intègre également la compilation et la simulation *RTL* du composant. Lors du processus de transcription, l'outil génère un code en *SystemC*, *VHDL* et *Verilog*. Il est possible d'exporter directement un composant sous forme de bloc *IP* depuis le *HLS*, facilement intégrables sur d'autres plateformes.

7.3 PRÉSENTATION DE LA NOUVELLE MÉTHODOLOGIE DE CONCEPTION

La nouvelle méthodologie de conception est basée sur la combinaison des deux approches citées précédemment, à savoir l'approche de conception basée sur la synthèse de haut niveau combinée à l'approche classique. L'introduction de l'outil *HLS* dans le flot de conception classique permet d'automatiser une partie de la génération de la description *RTL*. Ainsi, le concepteur pourra aisément passer du domaine algorithmique jusqu'à la génération d'un *IP* et l'implémenter ensuite sur cible *FPGA*.

La figure 7.7. donne une vue globale de cette nouvelle méthodologie.

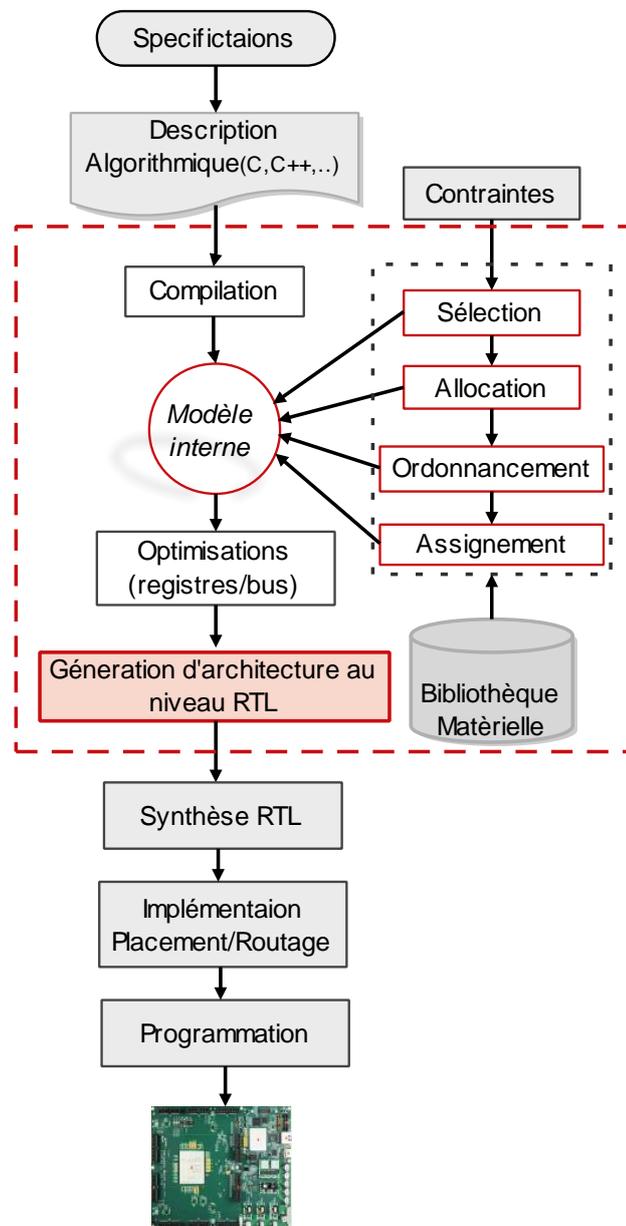


Figure.7.7. Nouvelle méthodologie de conception.

Cette nouvelle méthodologie, ainsi présentée, servira pour le développement hardware des deux contrôleurs bio inspirés *ACO_NPU_MPPT* et *BSO_MPPT*.

Ainsi, comme illustré dans la Figure.7.8, pour la méthodologie de conception adoptée, nous avons sélectionné conjointement les deux flots de conception *VIVADO HLS* [199] et *VIVADO ISE Design Suite* de *XILINX* [200]. Ces derniers permettront d’aboutir à une solution aux deux contrôleurs bio inspirés *ACO_NPU* et *BSO MPPT* décrits en langage C jusqu’à leur implémentation sur circuit FPGA.

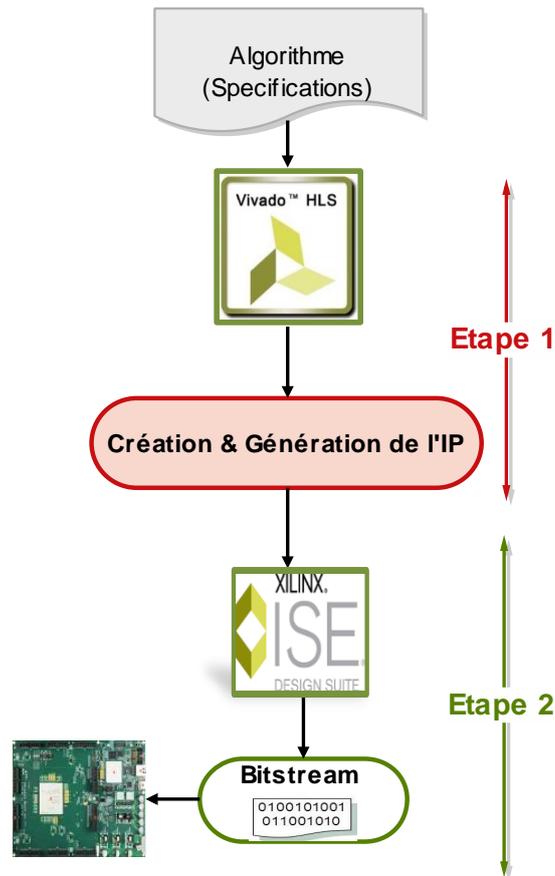


Figure.7.8. Flot de conception FPGA basé VIVADO HLS / VIVADO Design Suite.

La méthodologie de conception adoptée est subdivisée en deux étapes distinctes :

- La première consiste en la création d’un IP généré automatiquement à partir du flot de conception de l’outil de synthèse *Vivado HLS* de Xilinx, partant d’une description de haut niveau (algorithmique) ;
- Au cours de la seconde étape, l’IP ainsi créé sera intégré dans le flot de conception de l’outil *Vivado ISE Design Suite* de Xilinx. Ce dernier suivra toutes les étapes de conception classique jusqu’à l’implémentation sur cible FPGA.

Dans ce qui suit, nous allons donner un aperçu de chacun de ces deux flots, à savoir :

- le flot à base de l'outil *VIVADO HLS* ;
- le flot à base de l'outil *VIVADO Design Suite*.

7.3.1 L'outil *VIVADO HLS*

Le flot de conception *VIVADO High Level synthesis (HLS)* accepte en entrée des descriptions en langages C, C++, SystemC. Il offre un environnement graphique intégré pour la génération rapide d'IPs. Les IPs ainsi générés peuvent être intégrés dans la bibliothèque *IP Catalogue* du flot de conception *Vivado Design Suite* ou ils peuvent être instanciés aisément les rendant ainsi portables sur d'autres plateformes, ce qui facilite le prototypage rapide.

La figure ci-dessous illustre les différentes étapes constituant le flot de conception *VIVADO HLS*.

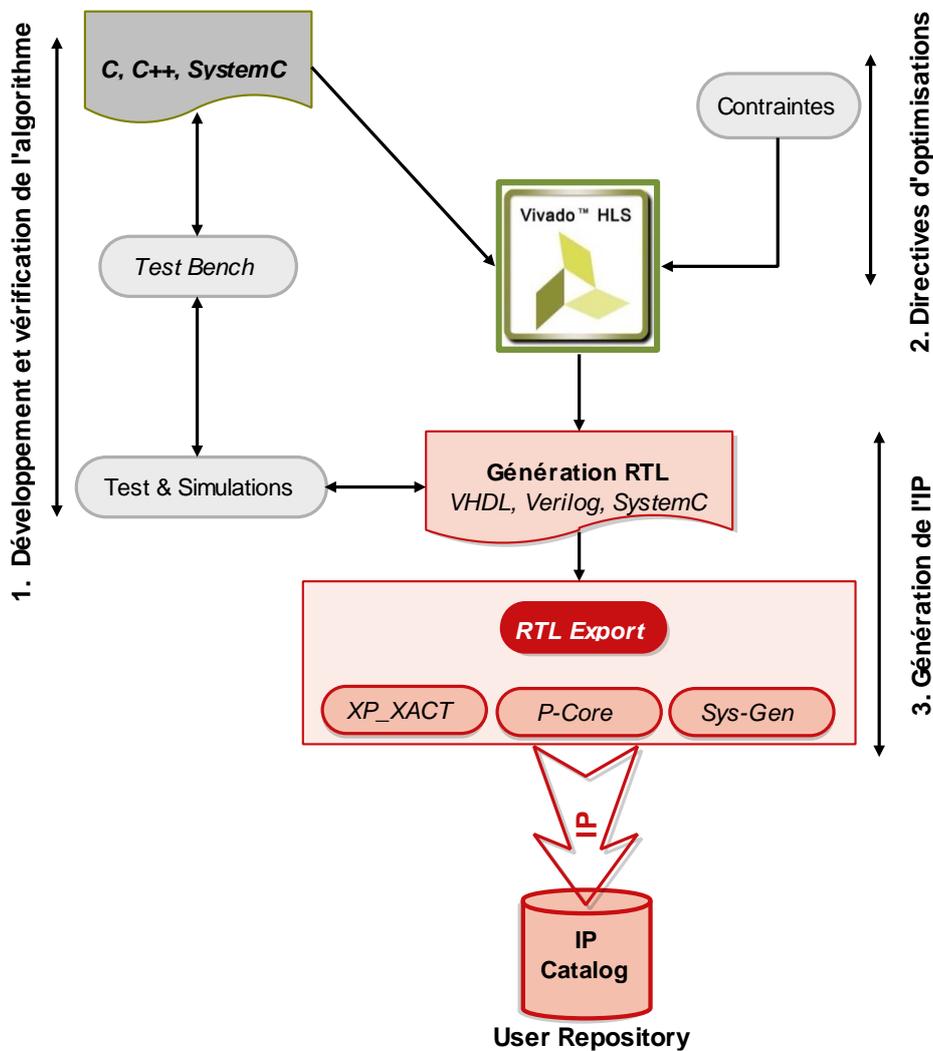


Figure.7.9. Flot de conception de l'outil *VIVADO HLS*.

Les différentes étapes du flot de conception sont comme suit :

Développement et vérification de l'algorithme

Au cours de cette étape, le concepteur doit :

- Concevoir l'algorithme ciblé pour l'implémentation ;
- Développer le code de l'algorithme dans un langage de haut niveau (langage C/C++/....) ;
- Tester et vérifier le fonctionnement de l'algorithme via un *Test Bench*.

La conception élaborée doit prendre en considération l'architecture ciblée. Vu que l'algorithme sera implémenté sur un circuit FPGA, il faudra prendre en considération les capacités (pour exploiter le parallélisme, ...) et les limites (pour réduire la consommation, réduire les délais, ...) de l'architecture.

Les contraintes et directives d'optimisation

Au cours de cette étape, le concepteur doit vérifier :

- Les contraintes sur les caractéristiques de la description finale générée par la synthèse, comme les temps de propagation, les temps de réponse, les types de ressources utilisés, ...etc. ;
- Les optimisations sur le code développé afin de réduire la consommation et les délais.

Ces optimisations seront discutées plus en détail dans la *section 7.3.1.1*.

Génération de la description RTL

Cette étape consiste à prendre le code développé de l'algorithme, les contraintes définies et les optimisations appliquées et générer la description au niveau RTL de cet algorithme dans un langage de description matérielle (VHDL/Verilog/SystemC). Dans cette phase, il est possible aussi que la synthèse ne réussisse pas, en raison des erreurs dans le code, des contraintes qui ne peuvent pas être respectés ou des optimisations non applicables. Pour cela, il faut revenir à l'étape précédente et fixer le problème. A la fin de cette étape, l'outil de synthèse génère un IP réutilisable. Ce dernier sera intégré dans la bibliothèque *IP Catalogue* et exploité en vue d'une implémentation Hardware.

7.3.1.1 Directives d'optimisation

Après la phase de développement et de vérification de l'algorithme, des transformations peuvent être appliquées selon des *directives d'optimisation* introduites par le concepteur. L'outil HLS lance alors un processus d'exploration en effectuant des tâches de transformation

spécifiées tout en évaluant la qualité des circuits résultants (surface, fréquence, parfois latence). Ceci va permettre de réduire la consommation de ressources, le temps de réponse, et la consommation en puissance. Les optimisations qui peuvent être appliquées sont de deux types :

- **L'optimisation de la logique du code C de l'algorithme :**

Cela consiste à modifier le code de l'algorithme afin d'améliorer la logique de ce dernier tout en gardant les mêmes objectifs.

- **L'optimisation du code C de l'algorithme :**

Cela consiste à appliquer certaines techniques sur le *type de données* utilisées, sur la *structure du code* (tableaux, boucles, etc..) ainsi que sur les *fonctions* utilisées, permettant ainsi d'agir sur l'architecture générée et de l'ajuster afin d'atteindre les mêmes spécifications de départ mais de manière optimale. En effet, le choix des types des données utilisées est très important pour la synthèse et peut parfois influencer considérablement les performances et la consommation de ressources du circuit final. De plus la structure du code permet d'appliquer des optimisations spécifiques, tels que :

Mapping des tableaux, permet de combiner plusieurs tableaux en un seul ;

Reshaping des tableaux qui permet de réarranger l'implémentation hardware du tableau (ex: combiner plusieurs éléments pour réduire le nombre d'accès) ;

Flattening des boucles qui convertit un ensemble de boucles imbriquées en une seule ;

Unrolling des boucles, convertit une boucle « for » en un ensemble d'opérations indépendantes ;

Pipelining des boucles, parallélise l'exécution des opérations de la boucle (**Figure.7.10.**) ;

Pipelining des fonctions, parallélise l'exécution des opérations d'une fonction (**Figure.7.11**) ;

Inlining des fonctions, permet de remplacer les appels de la fonction par son code ;

Dataflow Pipelining des fonctions, permet de paralléliser les appels de plusieurs fonctions, réduisant ainsi le code (**Figure.7.12**).

La Figure.7.10. illustre l'influence du *pipelining de boucle* appliquée dans un code. Le nombre de cycle de l'instruction est réduit à un cycle.

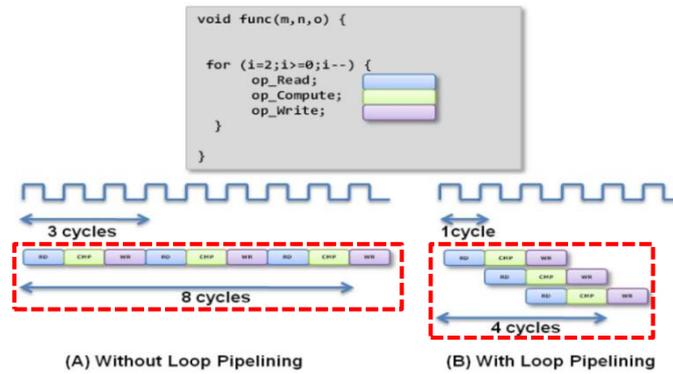


Figure.7.10. Optimisation « *Pipelining des boucles* » (Xilinx.2012).

La Figure.7.11. illustre l’influence du *pipelining de fonction* appliquée dans un code. Grace au parallélisme de fonction, le nombre de cycle est réduit à deux.

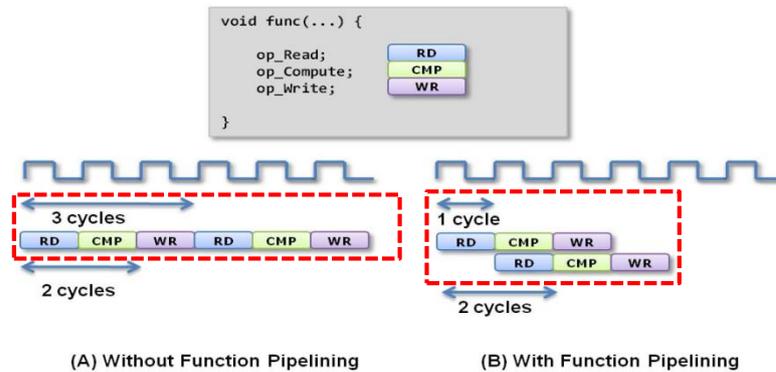


Figure.7.11. Optimisation « *Pipelining des fonctions* » (Xilinx.2012).

La Figure.7.12. illustre l’influence du *Dataflow Pipelining* appliqué pour les fonction. Grace au parallélisme de fonction, le nombre de cycle est réduit à deux.

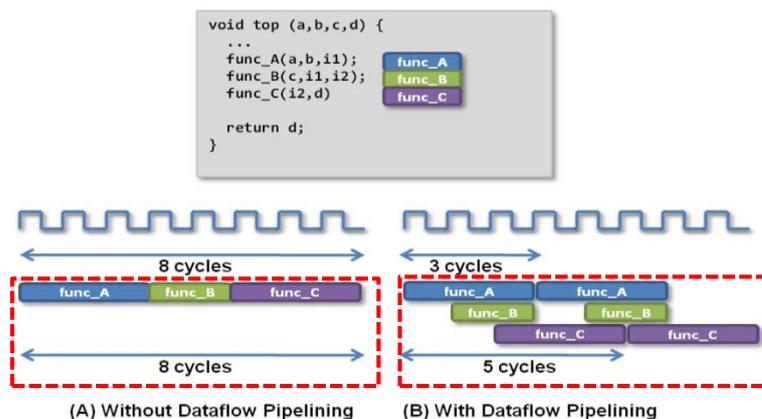


Figure.7.12. Optimisation « *Dataflow Pipelining des fonctions* » (Xilinx.2012).

En tenant compte des différentes optimisations que l'outil *HLS* offre [201], le concepteur élabore sa liste de directives pour la génération d'un *IP* optimisé.

7.3.2 L'outil *VIVADO ISE Design Suite*

Le logiciel Vivado ISE de Xilinx est un environnement de développement pour les systèmes numériques permettant de parcourir toutes les étapes de développement d'un projet de conception sur cible FPGA. Les designs peuvent être décrits sous trois formes principales : sous forme de schémas, de HDLs, ou de diagrammes d'états. De plus, les nouvelles versions facilitent l'intégration et la réutilisation des design sous forme de blocs IP (Intellectual Property) générés à partir des flots tel que *Vivado HLS*, *HDL Coder*, ...etc., ainsi que l'amélioration de la productivité de conception avec des implémentations sur les dernières familles de circuits programmables de Xilinx (Série7). Ces derniers incluent plusieurs fonctionnalités permettant, non seulement d'optimiser les produits délivrés, mais aussi, de réduire considérablement le temps de développement. La figure ci-dessous illustre les différentes étapes constituant le flot de conception *VIVADO ISE Design Suite* : depuis la phase d'intégration d'un *IP* jusqu'à la génération du fichier bitstream pour l'implémentation sur circuit FPGA.

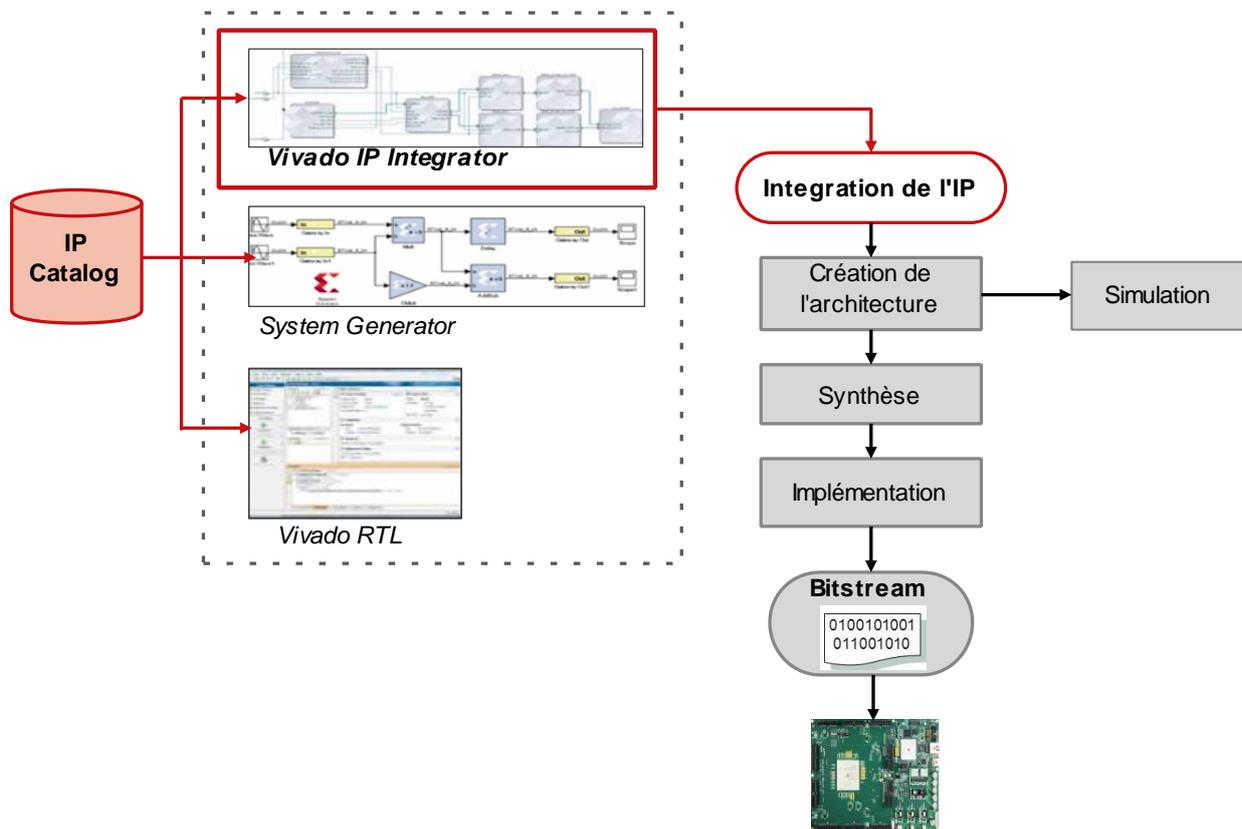


Figure.7.13. Flot de conception de l'outil *VIVADO ISE/ Integrateur d'IP*.

7.4 APPLICATION DE L'APPROCHE *HLS* POUR LE PROTOTYPAGE RAPIDE DES CONTRÔLEURS BIO INSPIRÉS *ACO_NPU_MPPT* ET *BSO_MPPT*

Après validation des performances des contrôleurs bio inspirés développés dans la partie software (*Chapitre.4*), au cours de cette section, la méthodologie de conception décrite au *chapitre.7.3* précédent, est adoptée pour un prototypage rapide et une implémentation hardware efficace de ces contrôleurs depuis leur description en langage *C* jusqu'à la création et l'implémentation hardware de leurs *IPs* respectifs : *IP_ACO_NPU_MPPT* et *IP_BSO_MPPT*. L'outil *Vivado HLS* sera exploité pour la génération des deux *IPs*. Ces derniers seront implémentés sur cible FPGA de la famille Xilinx avec l'outil *Vivado ISE Design Suite*.

7.4.1 Génération des *IPs ACO_NPU_MPPT* et *BSO_MPPT*

Dans le cadre de nos travaux de recherche, nous avons ciblé la carte de prototypage *Virtex7* avec le circuit FPGA *xc7vx485tffg1761-2* pour l'implémentation hardware des *IPs* à générer.

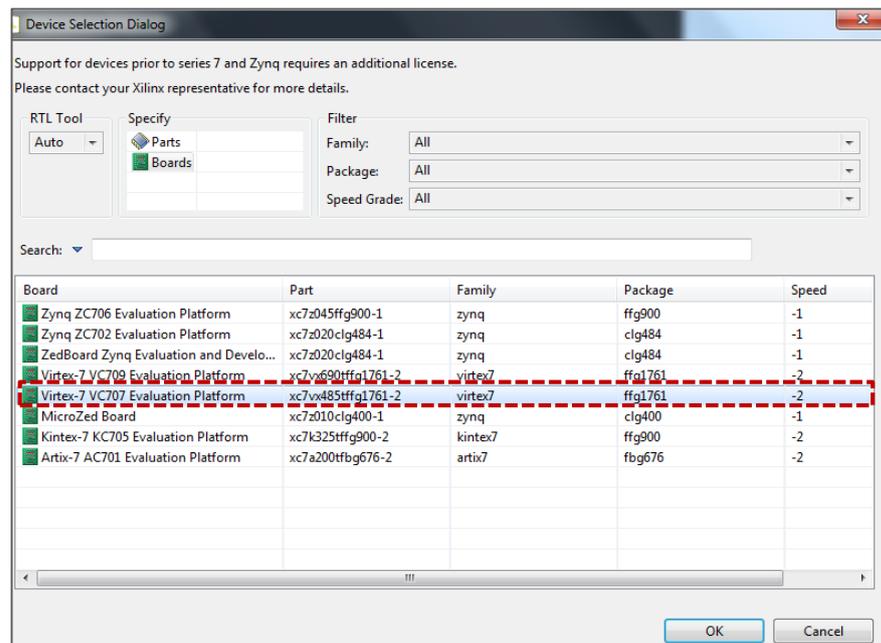


Figure.7.14. Sélection de la carte de prototypage/ circuit FPGA.

7.4.1.1 Génération de l'*IP ACO_NPU_MPPT*

La génération de l'*IP ACO_NPU_MPPT* se fait en trois étapes comme illustré en Figure.7.9. Les différentes tâches présentées en *section 7.2.1.2*, relatives à chacune de ces étapes seront appliquées par l'outil *HLS* pour la génération de l'*IP ACO_NPU_MPPT*.

7.4.1.1.1 Développement et vérification du code C pour le contrôleur ACO_NPU_MPPT

Au cours de cette étape, nous avons procédé à la description du contrôleur ACO_NPU_MPPT en langage C. Le code de ce dernier est subdivisé en deux : les *fonctions* et les *données* avec :

Le fichier *source* « *aco.c* » contenant le programme de l’algorithme ACO_NPU, constitué de la fonction principale et d’un ensemble de sous fonctions ;

Le fichier *header* « *aco.h* » contenant la déclaration des paramètres de l’algorithme ACO_NPU.

- La structure de *données* que nous avons utilisées afin de stocker les données relatives à l’algorithme est illustrée en Figure.7.15.

```

data_t  data[DATA_LENGTH]  =
data_t  fitness[DATA_LENGTH] =
data_t  pheromon[NB_ANT]   =

unsigned char  ACO_init
unsigned int   ACO_init_index
unsigned int   ACO_ant_index
data_t        ACO_pheromon_sum
data_t        ACO_alpha

unsigned int   ACO_iter_index

#ifdef MAX_ITER
    unsigned char ACO_finish
#endif // MAX_ITER

unsigned long  _holdrand
    
```

Figure.7.15. Les structures de données utilisées.

data : est un tableau, dans lequel l’ensemble des solutions de l’archive, ainsi que toutes les nouvelles solutions générées pour chaque itération de l’algorithme sont stockées ;

fitness : est un tableau contenant toutes les valeurs de la fonction objective (P_{PV}) ;

phéromone : est un tableau contenant la valeur du taux de phéromones pour chaque solution générée.

- Les *fonctions* ci-dessous que nous avons développées sont illustrées en Figure.7.16.

aco_search : la fonction principale de l’algorithme ;

aco_init : utilisé pour l’initialisation de l’archive ;

aco_evolve : contient le corps de l’algorithme ;

aco_monitor : pour la surveillance des changements climatiques ;

aco_generate : pour la génération de nouvelles solutions ;

aco_sort : utilisé pour trier l’archive de solution à la fin de chaque itération ;

```

/* generate a random value between 0 and 1 */
data_t random_t (void);

/* main function for ACO search algorithm */
data_t aco_search (data_t t, data_t i);

/* initialize the ACO solution archive */
data_t aco_init (data_t t, data_t i);

/* evolve the algorithm state one more iteration */
data_t aco_evolve (data_t t, data_t i);

/* monitor possible climatic changes */
data_t aco_monitor (data_t t, data_t i);

/* generate new solutions */
data_t aco_generate (void);

/* sort the archive according to fitness value */
void aco_sort (void);

```

Figure.7.16. Les fonctions utilisées.

Suite à cela, pour la validation du code C développé, nous avons élaboré un fichier *Test_Bench* « *aco_test.c* », afin de tester et valider les fonctionnalités du contrôleur. Pour cela, nous avons pris en considération deux occurrences à savoir : recherche du PPM dans des conditions standard et recherche du PPM sous ombrage partiel. Il est à noter que ce même fichier « *aco_test.c* » nous servira comme base pour la simulation comportementale de l’*IP_ACO_NPU_MPPT* durant la phase de synthèse logique et d’implémentation avec l’outil *Vivado ISE*.

La Figure.7.17. illustre la validation du code C du contrôleur *ACO_NPU_MPPT*

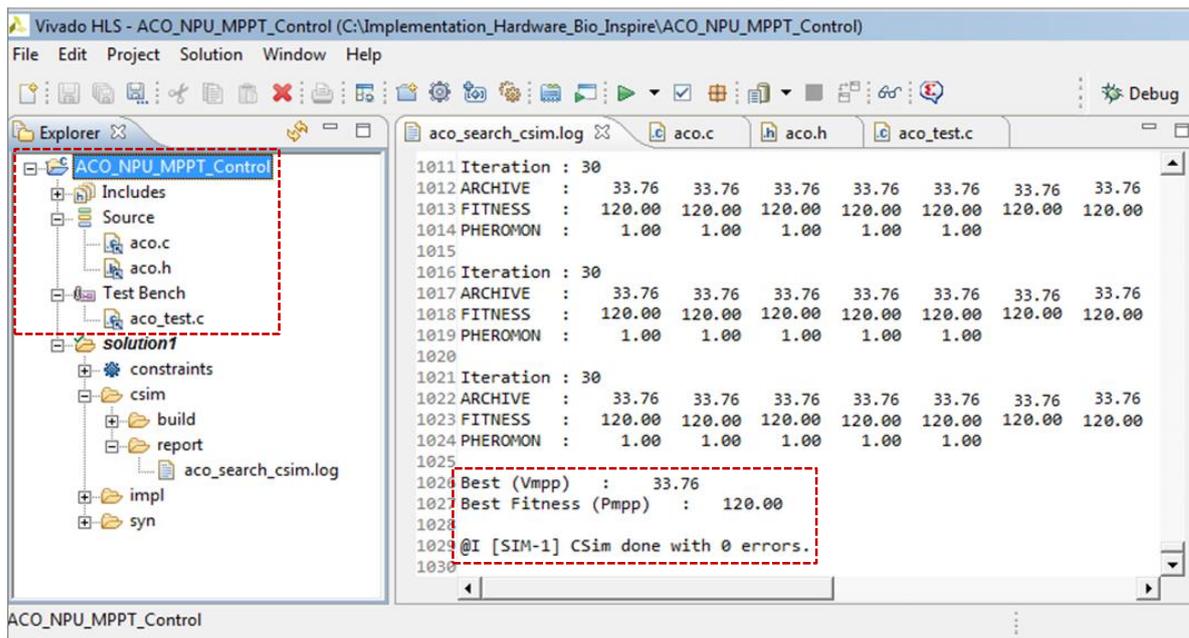


Figure.7.17. Validation du code C pour le contrôleur « *ACO_NPU_MPPT* ».

7.4.1.1.2 Application des directives d'optimisation pour le contrôleur ACO_NPU_MPPT

Afin de mettre en évidence l'apport de l'application des directives d'optimisation lors de la phase de synthèse du code C du contrôleur ACO_NPU_MPPT, nous avons en premier lieu, synthétisé le design sans tenir compte des directives d'optimisation.

Le tableau 7.1 illustre une estimation de l'utilisation des ressources du design après l'étape de synthèse du contrôleur ACO_NPU_MPPT.

Tableau.7.1. Estimation des ressources du design.

Summary				
Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	32
FIFO	-	-	-	-
Instance	0	40	20424	39166
Memory	4	-	0	0
Multiplexer	-	-	-	880
Register	-	-	375	-
Total	4	40	20799	40078
Available	2060	2800	607200	303600
Utilization (%)	~0	1	3	13

Au vu des résultats obtenus, il ressort que l'estimation des ressources consommées (Tableau.7.1) relatives au nombre de LUT, et FF utilisés est relativement importante. En effet, le nombre de FF utilisé est de 20799. Ce dernier, regroupe 375 Registres et 20424 Instances. De même pour les LUT utilisées, qui sont au nombre de 40078, regroupant 39166 Instances, 880 multiplexeurs et 32 Expressions. Afin de réduire ces ressources, il est nécessaire de modifier la structure du code en insérant des directives d'optimisation offertes par l'outil de synthèse. Pour cela, nous avons appliqué les directives d'optimisation ci-dessous :

- La réduction du nombre de tableaux utilisés, en combinant l'archive de solutions et l'ensemble des solutions générés dans un même tableau, pour garder les meilleures solutions à la fin de chaque itération, il suffit donc de trier seulement un seul tableau ;
- L'utilisation du type « float » pour les valeurs de courant, de tension et de puissance ;
- L'application du « flattening » dans la fonction « aco_sort », sur la première boucle imbriquée (Figure.7.18). Ceci permettra de convertir l'ensemble des boucles imbriquées en une seule boucle, ce qui va permettre de :
 - Réduire le nombre de tests de fin de boucle à un seul test ;
 - Eliminer le temps de passage entre les boucles ;

- Réduire la consommation vu que les mécanismes de control (variables de control, l'initialisation, condition de fin et l'incrémentation) des boucles intérieures vont être éliminés.
- L'application d'une optimisation de type « **inlining** » pour la fonction « *random_t* », vu que la fonction n'est appelée qu'une seule fois dans le code, cela permet d'éliminer le temps nécessaire pour le passage d'une fonction à une autre.

La Figure.7.18, illustre les directives d'optimisations appliquées au code source du contrôleur ACO_NPU_MPPT.

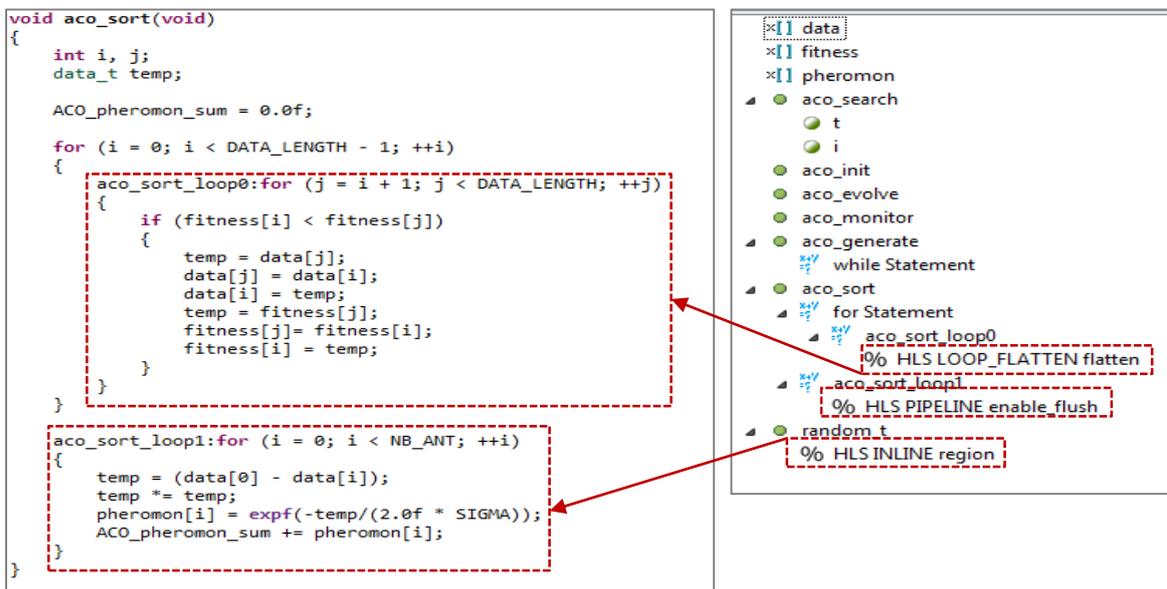


Figure.7.18 Directives d'optimisations appliquées au code C du contrôleur ACO_NPU_MPPT.

7.4.1.1.3 Synthèse du code C du contrôleur ACO_NPU_MPPT

Le code C du contrôleur ACO_NPU_MPPT ainsi validé et optimisé selon les directives appliquées peut être synthétisé. Au cours de cette étape, l'outil HLS prend le code C, les contraintes et les directives d'optimisation appliquées à ce dernier et génère la description matérielle du circuit. En effet, à partir des résultats illustrés Tableaux.7.2 à 7.5 ci-dessous, les résultats de la synthèse obtenus après application des directives d'optimisation montrent bien une réduction en terme d'estimation sur la consommation de ressources du design. L'outil Vivado HLS a pu réduire le nombre de LUTs, de registres, de multiplexeurs.

- Le nombre de *FF* a été réduit de 20799 à 5836 (soit une réduction de ~72%), où le nombre de *registres* est passé de 375 à 160 (soit une réduction de ~57%) et le nombre d'*instances* de 20424 à 5676 (soit une réduction de ~72%) ;
- Le nombre de LUT est réduit d'un total de 40078 à seulement 9785 (soit une réduction de ~76%), regroupant ainsi 9453 instances au lieu de 39166 (soit une réduction de ~76%), 300 multiplexeurs au lieu de 880 (soit une réduction de ~66%) et 32 expressions.

Tableau.7.2. Estimation des ressources du design après application des directives d'optimisation.

Summary				
Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	32
FIFO	-	-	-	-
Instance	0	40	5676	9453
Memory	4	-	0	0
Multiplexer	-	-	-	300
Register	-	-	160	-
Total	4	40	5836	9785
Available	2060	2800	607200	303600
Utilization (%)	~0	1	~0	3

Tableau.7.3. Détails des instances du design.

Instance					
Instance	Module	BRAM_18K	DSP48E	FF	LUT
grp_aco_search_aco_evolve_fu_87	aco_search_aco_evolve	0	21	3049	5009
grp_aco_search_aco_init_fu_123	aco_search_aco_init	0	7	1136	1910
grp_aco_search_aco_sort_fu_111	aco_search_aco_sort	0	12	1425	2462
aco_search_fcmp_32ns_32ns_1_1_U31	aco_search_fcmp_32ns_32ns_1_1	0	0	66	72
Total		4	40	5676	9453

Tableau.7.4. Mémoire BRAM instancié dans le design.

Memory								
Memory	Module	BRAM_18K	FF	LUT	Words	Bits	Banks	W*Bits*Banks
data_U	aco_search_data	2	0	0	15	32	1	480
fitness_U	aco_search_data	2	0	0	15	32	1	480
Total	2	4	0	0	30	64	2	960

Tableau.7.5. Registres instancié dans le design.

Register			
Name	FF	LUT	Bits
ACO_finish	8	0	8
ACO_init	8	0	8
ap_CS_fsm	9	0	9
grp_aco_search_aco_evolve_fu_87_ap_start_ap_start_reg	1	0	1
grp_aco_search_aco_init_fu_123_ap_start_ap_start_reg	1	0	1
grp_aco_search_aco_sort_fu_111_ap_start_ap_start_reg	1	0	1
p_0_reg_75	32	0	32
p_holdrand	32	0	32
r_1_reg_289	32	0	32
r_reg_274	32	0	32
tmp_18_reg_281	1	0	1
tmp_22_reg_296	1	0	1
tmp_6_reg_285	1	0	1
tmp_reg_270	1	0	1
Total	160	0	160

L'objectif visé est ainsi atteint avec moins d'efforts fournis de la part du concepteur. Ainsi, avec cette manière de faire, diverses configurations peuvent être choisies et implémentées en utilisant des directives d'optimisation que l'outil HLS prendra en charge. Le concepteur choisira la solution parmi plusieurs, présentant le meilleur rapport en termes de ressources consommées du circuit ciblé.

De plus, une fois le rapport de synthèse généré, l'outil *HLS* nous permettra de faire une analyse des performances du code du contrôleur de manière interactive lors de l'exécution des différentes opérations de ce dernier.

En effet, comme l'illustre la figure.7.18, les différentes *fonctions* relatives au code *C* du contrôleur *ACO_NPU_MPPT* peuvent être analysées en termes de cycles d'horloge nécessaires pour l'exécution de chaque fonction, de ressources allouées à chacune d'elles (opérateurs instanciés), ainsi que le nom de l'instance allouée à chaque fonction au niveau *RTL* fournit par l'outil *HLS*.

Pour exemple, en Figure.7.19, au cycle C_0 de la fonction principale du contrôleur *ACO_NPU_MPPT* « *aco_serach* », le premier état inclut des opérations de lecture, les instructions *i_read*, *voltage_read*, *aco_init_load* et *tmp* s'effectuent en parallèle en un seul cycle d'horloge.

Suite à cela, l'appel des sous-fonctions « *aco_init* », « *aco_sort* » et « *aco_evolve* » nécessite quant à lui, respectivement, deux cycles d'horloges chacun : C_1 et C_2 , C_5 et C_6 et C_7 et C_8 .

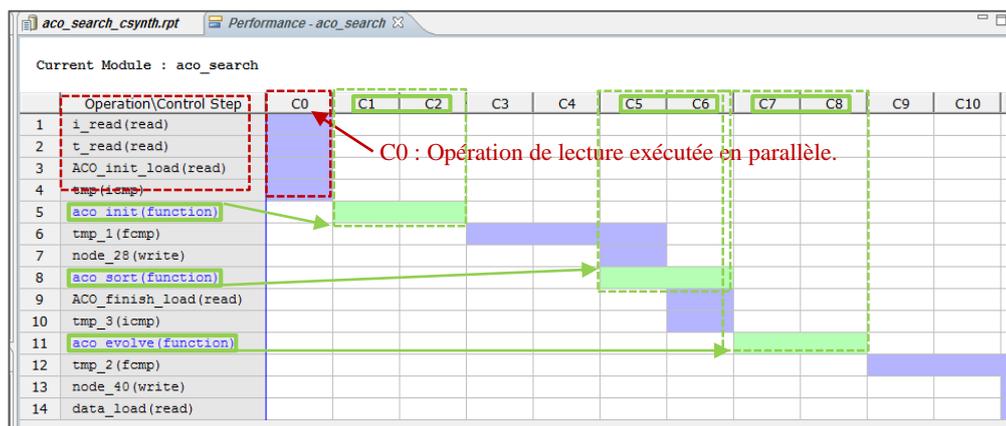


Figure.7.19. Analyse d'exécution des performances pour le contrôleur *ACO_NPU_MPPT*.

A chacune de ces fonctions correspond un nom d'instanciation. Pour exemple, la Figure.7.20, illustre les ressources allouées à la fonction « *aco_sort* », ainsi que le nom de l'instance qui lui correspondant au niveau *RTL*.

Property	Value
Call	aco_search_aco_sort
Control Input	
Control Output	
Data Input	
Data Output	
DSP	12
FF	1425
ID	36
Interval	207 ~ 753
Latency	207 ~ 753
Line number	38
LUT	2462
Name	node_36
Node Label	3
Node Latency	1
Opcode	call
Pipeline type	none
RTL Name	grp_aco_search_aco_sort_fu_111

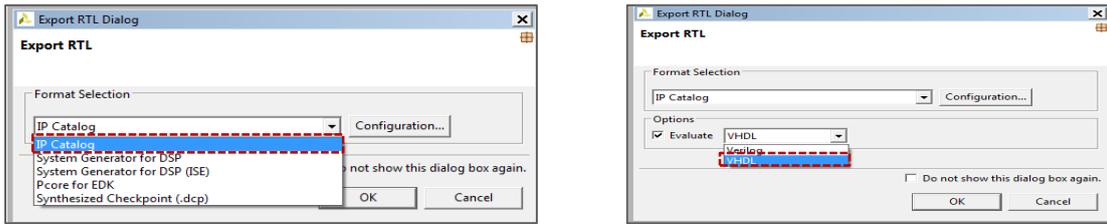
Figure.7.20. Ressources allouées à la fonction « aco_sort ».

7.4.1.1.4 Création de l'IP_ACO_NPU_MPPT : Exportation au niveau RTL

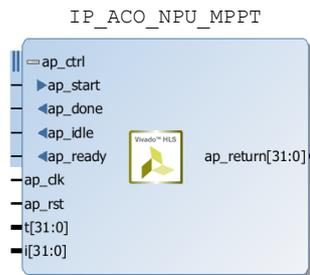
Au cours de cette étape, la création de l'IP pour le contrôleur *ACO_NPU_MPPT* sera générée en une description au niveau RTL. L'IP ainsi créé peut être intégré dans les flots *System Generator* et *Vivado ISE* de Xilinx pour une implémentation sur circuit FPGA. Le choix du langage HDL (*VHDL*, *Verilog* ou *SystemC*) et du format de génération de l'IP est laissé à l'appréciation du concepteur. Dans le cadre de nos travaux, l'IP est créé sous un format *IP_XAT* afin d'être intégré dans le flot Vivado ISE.

L'IP *ACO_NPU_MPPT* ainsi généré est stocké dans la bibliothèque *IP_Catalogue* de l'outil. Ce dernier, sera inséré dans le flot de conception Vivado Design Suite pour une implémentation hardware.

La Figure.7.21.(a)-(b)-(c) illustre respectivement les étapes relatives à la sélection du langage HDL, du mode de génération du bloc IP, la création de l'IP *ACO_NPU_MPPT* ainsi qu'un extrait du code *VHDL* généré pour la fonction « aco_sort ».



(a)



(b)

```

=====
-- RTL generated by Vivado(TM) HLS - High-Level Synthesis
from C, C++ and SystemC
-- Version: 2015.4
-- Copyright (C) 2015 Xilinx Inc. All rights reserved.
=====

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity aco_search_aco_sort is
port (
  ap_clk : IN STD_LOGIC;
  ap_rst : IN STD_LOGIC;
  ap_start : IN STD_LOGIC;
  ap_done : OUT STD_LOGIC;
  ap_idle : OUT STD_LOGIC;
  ap_ready : OUT STD_LOGIC;
  ACO_pheromon_sum : OUT STD_LOGIC_VECTOR (31 downto 0);
  ACO_pheromon_sum_ap_vld : OUT STD_LOGIC;
  fitness_address0 : OUT STD_LOGIC_VECTOR (3 downto 0);
  fitness_ce0 : OUT STD_LOGIC;
  fitness_we0 : OUT STD_LOGIC;
  fitness_d0 : OUT STD_LOGIC_VECTOR (31 downto 0);
  .....
  fitness_q1 : IN STD_LOGIC_VECTOR (31 downto 0);
  pheromon_address0 : OUT STD_LOGIC_VECTOR (2 downto 0);
  pheromon_we0 : OUT STD_LOGIC;
  pheromon_d0 : OUT STD_LOGIC_VECTOR (31 downto 0) );
end;

architecture behav of aco_search_aco_sort is

  component aco_search_faddsub_32ns_32ns_32_4_full_dsp IS

    port (
      clk : IN STD_LOGIC;
      reset : IN STD_LOGIC;
      din0 : IN STD_LOGIC_VECTOR (31 downto 0);
      din1 : IN STD_LOGIC_VECTOR (31 downto 0);
      opcode : IN STD_LOGIC_VECTOR (1 downto 0);
      ce : IN STD_LOGIC;
      dout : OUT STD_LOGIC_VECTOR (31 downto 0) );
    end component;

  component aco_search_fmuls_32ns_32ns_32_2_max_dsp IS

    port (
      clk : IN STD_LOGIC;
      reset : IN STD_LOGIC;
      din0 : IN STD_LOGIC_VECTOR (31 downto 0);
      din1 : IN STD_LOGIC_VECTOR (31 downto 0);
      ce : IN STD_LOGIC;
      dout : OUT STD_LOGIC_VECTOR (31 downto 0) );
    end component;

  component aco_search_fdiv_32ns_32ns_32_8 IS

    port (
      clk : IN STD_LOGIC;
      reset : IN STD_LOGIC;
      din0 : IN STD_LOGIC_VECTOR (31 downto 0);
      din1 : IN STD_LOGIC_VECTOR (31 downto 0);
      ce : IN STD_LOGIC;
      dout : OUT STD_LOGIC_VECTOR (31 downto 0) );
    end component;

  component aco_search_fcmp_32ns_32ns_1_1 IS

    port (
      din0 : IN STD_LOGIC_VECTOR (31 downto 0);
      din1 : IN STD_LOGIC_VECTOR (31 downto 0);
      opcode : IN STD_LOGIC_VECTOR (4 downto 0);
      dout : OUT STD_LOGIC_VECTOR (0 downto 0) );
    end component;

  component aco_search_fexp_32ns_32ns_32_6_full_dsp IS

    port (
      clk : IN STD_LOGIC;
      reset : IN STD_LOGIC;
      din0 : IN STD_LOGIC_VECTOR (31 downto 0);
      din1 : IN STD_LOGIC_VECTOR (31 downto 0);
      ce : IN STD_LOGIC;
      dout : OUT STD_LOGIC_VECTOR (31 downto 0) );
    end component;

```

(c)

Figure.7.21. (a) Sélection du mode de génération et du langage HDL ;
 (b) Création du bloc IP pour le contrôleur ACO_NPU_MPPT ;
 (c) Extrait du code VHDL généré pour la fonction « aco_search ».

7.4.1.2 Génération de l'IP BSO_MPPT

Pour la génération de l'IP BSO_MPPT, nous avons adopté la même démarche en suivant les mêmes étapes. C'est pour cela que certaines de ces étapes ne seront pas présentées dans le détail.

7.4.1.2.1 Développement et vérification du code C pour le contrôleur BSO_MPPT

Au cours de cette étape, nous avons procédé à la description du contrôleur en langage C, « *BSO.c* » et « *BSO.h* ». Tout comme pour le code du contrôleur ACO_NPU_MPPT, le code du contrôleur BSO_MPPT est subdivisé en deux : les *données* et les *fonctions*

- La structure de *données* utilisée afin de stocker les données relatives à l'algorithme est illustrée en Figure.7.22, avec :

```

data_t1 archive[];
data_t1 Fitness_initial[];
data_t1 rate[];
data_t1 Loudness[];
data_t1 Best;
data_t1 fit_init_max;

int i,cmp_affichage=0;
data_t1 vl=0.0f,c1=0.0f;
uint1 ps=0;
int main(void)
    
```

Figure.7.22. La structure de données utilisées pour le contrôleur BSO_MPPT.

- Les *fonctions* développées sont illustrées en Figure.7.23, avec :

- *main_BSO* : la fonction principale de l'algorithme ;
- *init_archive* : utilisée pour l'initialisation de l'archive ;
- *position_Bat* : utilisée pour la génération de nouvelles solutions ;
- *rank_fitness* : utilisée pour trier l'archive selon un ordre croissant de la fitness ;
- *best_solution* : utilisée pour sélectionner la meilleure solution à la fin de chaque itération ;
- *compare_solution* : utilisée pour identifier la nouvelle meilleure solution.

```

/* generate a random value between 0 and 1*/
data_t1 Val_random(void);

/*main function for BSO search algorithm*/
data_t1 Main_BSO(data_t1 v,data_t1 c);

/*initialize the BSO solution archive*/
data_t1 init_archive(data_t1 v,data_t1 c);

/*compute the freq, velo and position of the bats*/
data_t1 position_BAT(data_t1 v,data_t1 c);

/*best values of the fitness and voltage*/
data_t1 Best_Solution(void);

/*sort the archive (fitmax,fitmin) with the correspodng voltage
data_t1 rank(data_t1 sort_voltage[taille],
data_t1 rank_fitness[taille],int taille);

/*identifie the news best solution
data_t1 Compare_Solution(void);
    
```

Figure.7.23. Les fonctions utilisées pour le contrôleur BSO_MPPT.

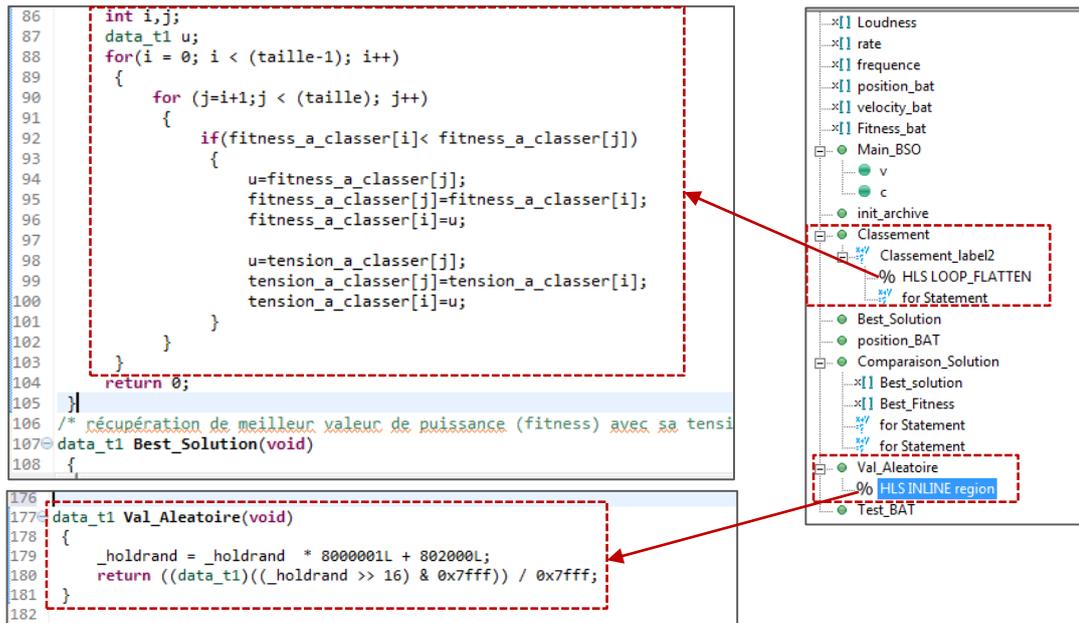


Figure.7.25. Directives d’optimisations appliquées au code C du contrôleur BSO_MPPT.

7.4.1.2.3 Synthèse du code C du contrôleur BSO_MPPT

Le code C du contrôleur *BSO_MPPT* ainsi optimisé peut être synthétisé. Le rapport d’estimation de ressources consommées après la synthèse est représenté par le **Tableau.7.6**. Les tableaux 7.6 à 7.9 illustrent l’utilisation des ressources du design après l’étape de synthèse du contrôleur *BSO_MPPT*.

Tableau.7.6. Estimation des ressources du design après application des directives d’optimisation.

Summary				
Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	2	0	228
FIFO	-	-	-	-
Instance	4	58	5710	8804
Memory	4	-	128	4
Multiplexer	-	-	-	437
Register	-	-	437	-
Total	8	60	6275	9473
Available	2060	2800	607200	303600
Utilization (%)	~0	2	1	3

Comme l’illustre sur le Tableau, le contrôleur *BSO_MPPT* n’a utilisé que 3% des ressources du circuit tout comme pour le contrôleur *ACO_NPU_MPPT*.

Tableau.7.7. Détails des instances du design.

Instance	Module	BRAM_18K	DSP48E	FF	LUT
grp_Main_BSO_Classement_1_fu_210	Main_BSO_Classement_1	0	0	218	191
grp_Main_BSO_Comparaison_Solution_fu_194	Main_BSO_Comparaison_Solution	4	0	288	379
Main_BSO_fadd_32ns_32ns_32_4_full_dsp_U39	Main_BSO_fadd_32ns_32ns_32_4_full_dsp	0	2	227	214
Main_BSO_fdiv_32ns_32ns_32_8_U41	Main_BSO_fdiv_32ns_32ns_32_8	0	0	359	802
Main_BSO_fmMul_32ns_32ns_32_2_max_dsp_U40	Main_BSO_fmMul_32ns_32ns_32_2_max_dsp	0	3	128	135
grp_Main_BSO_position_BAT_fu_162	Main_BSO_position_BAT	0	53	4362	6742
Main_BSO_sitofp_32ns_32_3_U42	Main_BSO_sitofp_32ns_32_3	0	0	128	341
Total		7	4	5710	8804

Tableau.7.8. Mémoire BRAM instanciées dans le design.

Memory	Module	BRAM_18K	FF	LUT	Words	Bits	Banks	W*Bits*Banks
archive_U	Main_BSO_archive	2	0	0	5	32	1	160
Fitness_initial_U	Main_BSO_archive	2	0	0	5	32	1	160
position_bat_U	Main_BSO_position_BAT_velocity_bat	0	64	2	3	32	1	96
Fitness_bat_U	Main_BSO_position_BAT_velocity_bat	0	64	2	3	32	1	96
Total		4	4	128	4	128	4	512

Tableau.7.9. Registre instanciées dans le design.

Name	FF	LUT
Best	32	0
ap_CS_fsm	32	0
compteur_archive	32	0
compteur_archive_load_reg_432	32	0
compteur_bat	32	0
compteur_iter	32	0
fit_init_max	32	0
grp_Main_BSO_Classement_1_fu_210_ap_start_ap_start_reg	1	0
grp_Main_BSO_Comparaison_Solution_fu_194_ap_start_ap_start_reg	1	0
grp_Main_BSO_position_BAT_fu_162_ap_start_ap_start_reg	1	0
p_0_reg_146	32	0
p_holdrand	32	0
reg_246	32	0
tmp_42_i_reg_478	1	0
tmp_45_i_cast_reg_482	15	0
tmp_47_i_reg_492	32	0
tmp_48_i_reg_497	32	0
tmp_50_i_reg_502	32	0
tmp_i_reg_454	1	0
tmp_reg_441	1	0
Total	437	0

Les Figures.7.26 et 7.27, illustrent respectivement l'analyse en termes du nombre de cycles d'horloge nécessaires pour l'exécution de chaque fonction, les ressources allouées à la sous-fonction « *comparaison_solution* » du contrôleur *BSO_MPPT* ainsi que le nom de l'instance correspondant à la fonction au niveau *RTL*.

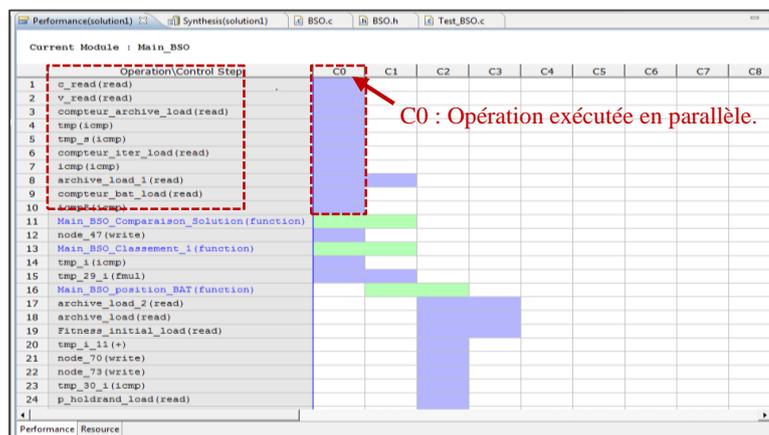


Figure.7.26. Analyse temporelle d'exécution du code pour le contrôleur *BSO_MPPT*.

Property	Value
Call	Main_BSO_Comparaison_Solution
Control Input	
Control Output	
Data Input	
Data Output	
DSP	0
FF	288
ID	43
Interval	82 ~ 208
Latency	82~208
Line number	53
LUT	379
Name	node_43
Node Label	0
Node Latency	1
Opcode	call
Pipeline type	none
RTL Name	grp_Main_BSO_Comparaison_Solution_fu_206
Source File	BSO_Source/Source_BSO/BSO.c

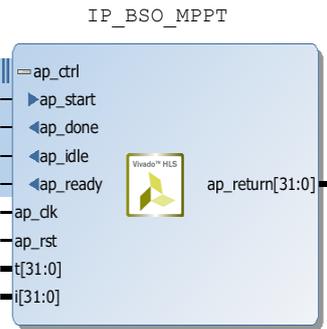
Figure.7.27. Ressources allouées pour la fonction « *comparison_solution* ».

7.4.1.2.4 Création de l'IP *BSO_MPPT* : Exportation au niveau RTL

Au cours de cette étape, la création de l'IP pour le contrôleur *BSO_MPPT* sera générée en une description au niveau RTL. Tout comme pour le contrôleur *ACO_NPU_MPPT*, le langage *VHDL* est sélectionné.

La Figure.7.28. (a)-(b) illustrent respectivement la création de l'IP *BSO_MPPT* ainsi qu'une partie du code *VHDL* généré pour la fonction « *comparaison_solution* ».

L'IP *BSO_MPPT* ainsi généré est stockée dans la bibliothèque *IP_Catalogue* de l'outil. Ce dernier, sera inséré dans le flot de conception *Vivado Design Suite* pour une implémentation hardware.



(a)

```

=====
-- RTL generated by Vivado(TM) HLS - High-Level Synthesis
from C, C++ and SystemC
-- Version: 2015.4
-- Copyright (C) 2015 Xilinx Inc. All rights reserved.
=====

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity Main_BSO_Comparaison_Solution is
port (
    ap_clk : IN STD_LOGIC;
    ap_rst : IN STD_LOGIC;
    ap_start : IN STD_LOGIC;
    ap_done : OUT STD_LOGIC;
    ap_idle : OUT STD_LOGIC;
    ap_ready : OUT STD_LOGIC;
    archive_address0 : OUT STD_LOGIC_VECTOR (2 downto 0);
    archive_ce0 : OUT STD_LOGIC;
    archive_we0 : OUT STD_LOGIC;
    archive_d0 : OUT STD_LOGIC_VECTOR (31 downto 0);

    position_bat_ce0 : OUT STD_LOGIC;
    position_bat_q0 : IN STD_LOGIC_VECTOR (31 downto 0);
    Fitness_bat_address0 : OUT STD_LOGIC_VECTOR(1downto 0);
    Fitness_bat_ce0 : OUT STD_LOGIC;
    Fitness_bat_q0 : IN STD_LOGIC_VECTOR (31 downto 0);
    Best : OUT STD_LOGIC_VECTOR (31 downto 0);
    Best_ap_vld : OUT STD_LOGIC;
    fit_init_max : OUT STD_LOGIC_VECTOR (31 downto 0);
    fit_init_max_ap_vld : OUT STD_LOGIC );
end;

architecture behav of Main_BSO_Comparaison_Solution is

component Main_BSO_Clasement IS
port (
    ap_clk : IN STD_LOGIC;
    ap_rst : IN STD_LOGIC;
    ap_start : IN STD_LOGIC;
    ap_done : OUT STD_LOGIC;
    ap_idle : OUT STD_LOGIC;
    ap_ready : OUT STD_LOGIC;
    tension_a_classer_add0:OUTSTD_LOGIC_VECTOR(2downto0);
    tension_a_classer_ce0 : OUT STD_LOGIC;
    tension_a_classer_we0 : OUT STD_LOGIC;
    fitness_a_classer_q0 : IN STD_LOGIC_VECTOR (31 downto 0);
    fitness_a_classer_address1:OUT STD_LOGIC_VECTOR(2downto
0);
    fitness_a_classer_ce1 : OUT STD_LOGIC;
    fitness_a_classer_we1 : OUT STD_LOGIC;
    fitness_a_classer_d1 : OUT STD_LOGIC_VECTOR (31 downto
0);
    fitness_a_classer_q1 : IN STD_LOGIC_VECTOR (31 downto 0)
);
end component;

component Main_BSO_Comparaison_Solution_Best_solution IS
port (
    clk : IN STD_LOGIC;
    reset : IN STD_LOGIC;
    address0 : IN STD_LOGIC_VECTOR (2 downto 0);
    ce0 : IN STD_LOGIC;
    we0 : IN STD_LOGIC;
    d0 : IN STD_LOGIC_VECTOR (31 downto 0);
    q0 : OUT STD_LOGIC_VECTOR (31 downto 0);
    address1 : IN STD_LOGIC_VECTOR (2 downto 0);
    ce1 : IN STD_LOGIC;
    we1 : IN STD_LOGIC;
    d1 : IN STD_LOGIC_VECTOR (31 downto 0);
    q1 : OUT STD_LOGIC_VECTOR (31 downto 0) );
end component;

begin

    Best_solution_U :
componentMain_BSO_Comparaison_Solution_Best_solution

port map (
    clk => ap_clk,
    reset => ap_rst,
    address0 => Best_solution_address0,
    ce0 => Best_solution_ce0,
    we0 => Best_solution_we0,
    d0 => Best_solution_d0,
    q0 => Best_solution_q0,
    address1 => Best_solution_address1,
    ce1 => Best_solution_ce1,
    we1 => Best_solution_we1,
    d1 => Best_solution_d1,
    q1 => Best_solution_q1);

    Best_Fitness_U :
componentMain_BSO_Comparaison_Solution_Best_solution

port map (
    clk => ap_clk,
    reset => ap_rst,
    address0 => Best_Fitness_address0,
    ce0 => Best_Fitness_ce0,
    we0 => Best_Fitness_we0,
    d0 => Best_Fitness_d0,
    q0 => Best_Fitness_q0,
    address1 => Best_Fitness_address1,
    ce1 => Best_Fitness_ce1,
    we1 => Best_Fitness_we1,
    d1 => Best_Fitness_d1,
    q1 => Best_Fitness_q1);

end behav;

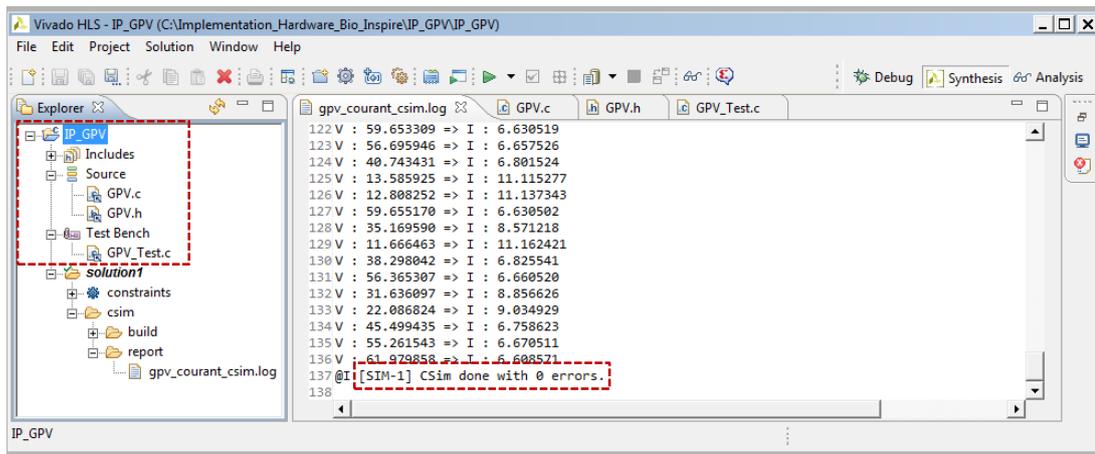
```

(b)

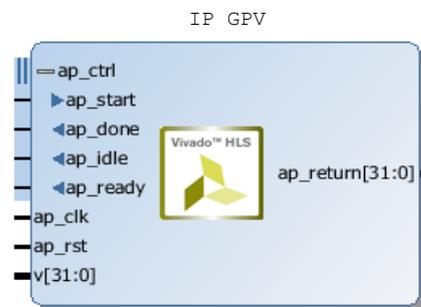
Figure.7.28. (a) Création du bloc IP pour le contrôleur *BSO_MPPT* généré ;
 (b) Code VHDL généré pour la fonction « *main_BSO_compaison_solution* ».

Afin de valider les deux IPs ainsi créés, il est nécessaire que ces derniers soient intégrés dans un environnement de test. Pour cela l'élaboration d'une architecture permettant d'intégrer ces derniers et de simuler leurs comportements dans un système PV est nécessaire. Par conséquent, la création d'un IP générateur PV s'avère indispensable. Ainsi, toutes les étapes citées précédemment pour la création des deux IPs_contrôleurs seront appliquées, cette fois, pour la création et la génération de l'IP_GPV.

La figure ci-dessous illustre respectivement la validation du code C pour le GPV ainsi que la création et la génération du bloc IP_GPV.



(a)



(b)

Figure.7.29. (a) Validation du code C pour le «GPV»
(b) Création et génération du bloc IP_GPV

La bibliothèque *IP_Catalogue* contient les IPs_MPPT générés. Il s'agit d'une bibliothèque contenant des blocs susceptibles d'être réutilisés. Par le biais d'un mécanisme de réplique, ces derniers pourront être utilisés sur d'autres plateformes telles que *IP_integrator* de Vivado. En d'autres termes, il doit pouvoir être intégré dans un système par d'autres concepteurs, n'ayant aucune information sur le détail du design. Ceci réduira considérablement le coût que pourrait impliquer sa conception à nouveau (Design Reuse et Time To Market).

7.4.2 Implémentation des IPs ACO_NPU_MPPT et BSO_MPPT sur FPGA

La méthodologie présentée en section 7.3.2 et illustrée par la **Figure.7.13** est adoptée pour l'implémentation hardware des deux IPs *ACO_NPU_MPPT* et *BSO_MPPT* ainsi créée. Ces derniers seront intégrés via *IP_integrator* du flot de conception *Vivado ISE* pour subir toutes les étapes nécessaires jusqu'à la génération du fichier bitstream respectifs. Il est à noter que lors de la phase de création du projet, il est important de cibler le même circuit FPGA à savoir le « *xc7vx485tffg1766-2* » Série 7, sélectionné lors de la phase de génération des IPs par l'outil Vivado HLS.

7.4.2.1 Implémentation de l'IP ACO_NPU_MPPT

L'implémentation hardware de l'IP généré est régie par un certain nombre d'étapes à savoir:

- *Integration de l'IP ACO_NPU_MPPT* dans la bibliothèque « *User Repository / IP Catalog* » de l'outil Vivado(*Figure7.30*) ;

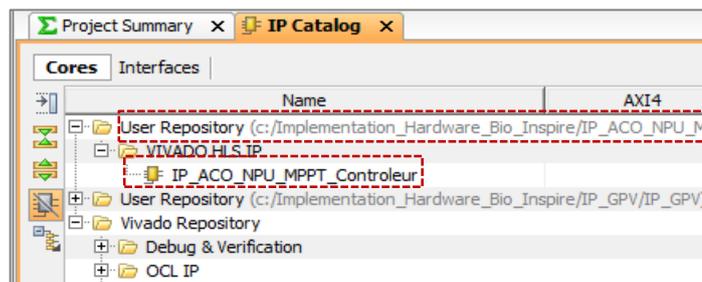


Figure.7.30. Intégration de l'IP «*ACO_MPPT*» dans *IP Catalog* la bibliothèque de l'outil Vivado.

- Par la suite *instancier l'IP intégré* et créer un **Wrapper HDL** afin d'élaborer l'architecture globale du système PV (*Figure7.31*) constituée de l'*IP_ACO_NPU_MPPT* et de l'*IP_GPV*, constituant ainsi le système PV ;

Cette étape permet de définir le module principal (Top Level Module) qui va englober l'ensemble de tous les composants.

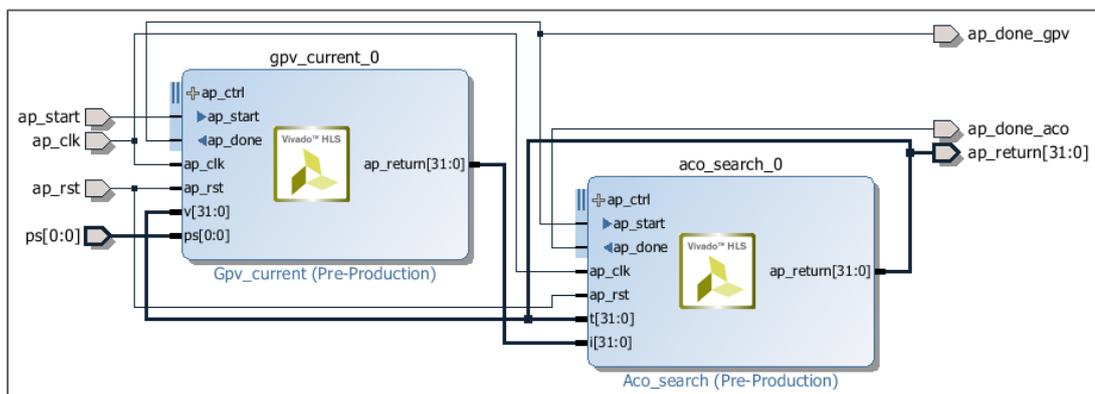


Figure.7.31. Architecture globale *IP_ACO_NPU_MPPT* et *IP_GPV* Wrappée.

A ce stade de développement, l'architecture obtenue suivra toutes les étapes relatives à un flot de conception classique (*simulation, synthèse, implémentation et génération du fichier bitstream*).

7.4.2.1.1 Simulation de l'IP_ACO_NPU_MPPT

Afin de valider l'architecture créée, une simulation comportementale s'avère nécessaire. A cet effet, l'outil *ModelSim* sera utilisé. Le système est simulé en premier lieu dans des conditions des conditions constantes de l'éclairage et de la température, par la suite, ce dernier est soumis à des conditions d'ombrage partiel. Pour cela, les signaux : les signaux : « *ap_clk* » pour la génération du signal d'horloge est mis à 1, « *ap_rst* » est mis à 0, « *ap_start* » pour l'activation du bloc IP validation de l'entrée est mis à 1, l'indicateur d'ombrage partiel « *ps* » est mis à 0 en premier, afin de simuler le comportement du système sous condition standard, puis par la suite, ce dernier est activé à 1 simulant ainsi le cas d'un ombrage partiel. En sortie, le signal « *ap_return* » illustre l'évolution de la phase de recherche du PPM.

Les Figures.7.32 et 7.33 illustrent les résultats de simulation respectifs.

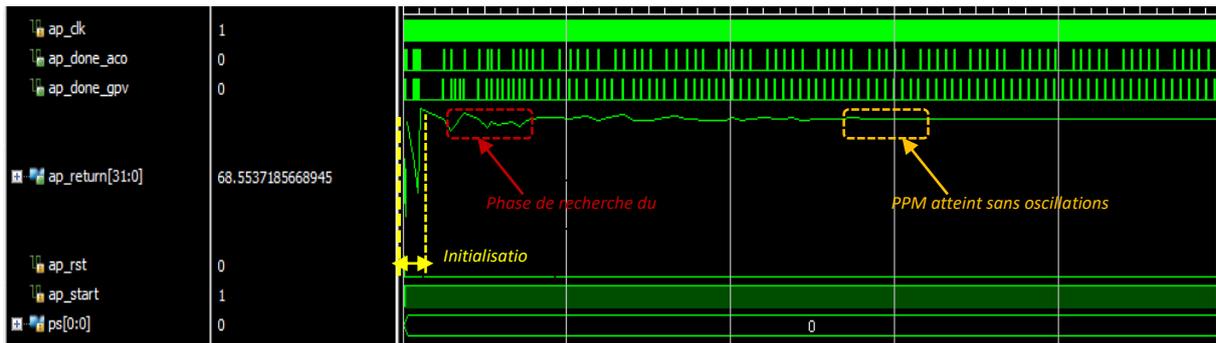


Figure.7.32. Simulation Comportementale l'IP_ACO_NPU_MPPT (sans ombrage).

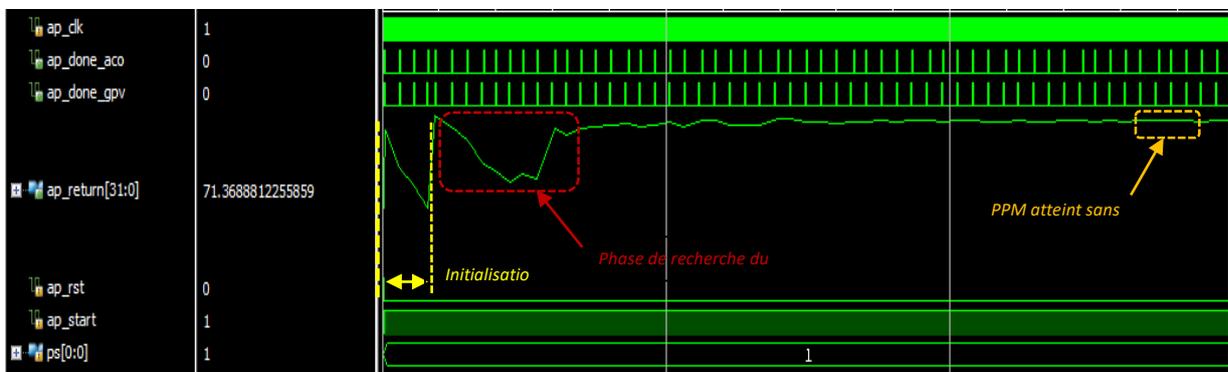


Figure.7.33. Simulation Comportementale de l'IP_ACO_NPU_MPPT (avec ombrage).

Au vu des résultats obtenus en Figures.7.32 et 7.33, pour les deux scénarios, à savoir, avec et sans ombrage partiel, le contrôleur *IP_ACO_NPU_MPPT* après une phase d'initialisation et de recherche du PPM, converge vers ce dernier, se stabilise sans oscillations.

7.4.2.1.2 Synthèse de l'*IP_ACO_NPU_MPPT*

Après validation de la simulation comportementale de l'architecture du système, l'étape de synthèse logique, qui permet d'avoir une transcription du code HDL de l'*IP_ACO_NPU_MPPT* en une représentation d'une architecture sous forme RTL est effectuée par l'outil de synthèse. Les Figures.7.34 à 7.36 illustrent respectivement une vue globale de l'architecture du système après l'étape de synthèse de, la vue interne du schéma RTL des différents blocs constituant l'architecture, ainsi que le détail de l'architecture de la fonction « *aco_sort* » au niveau RTL.

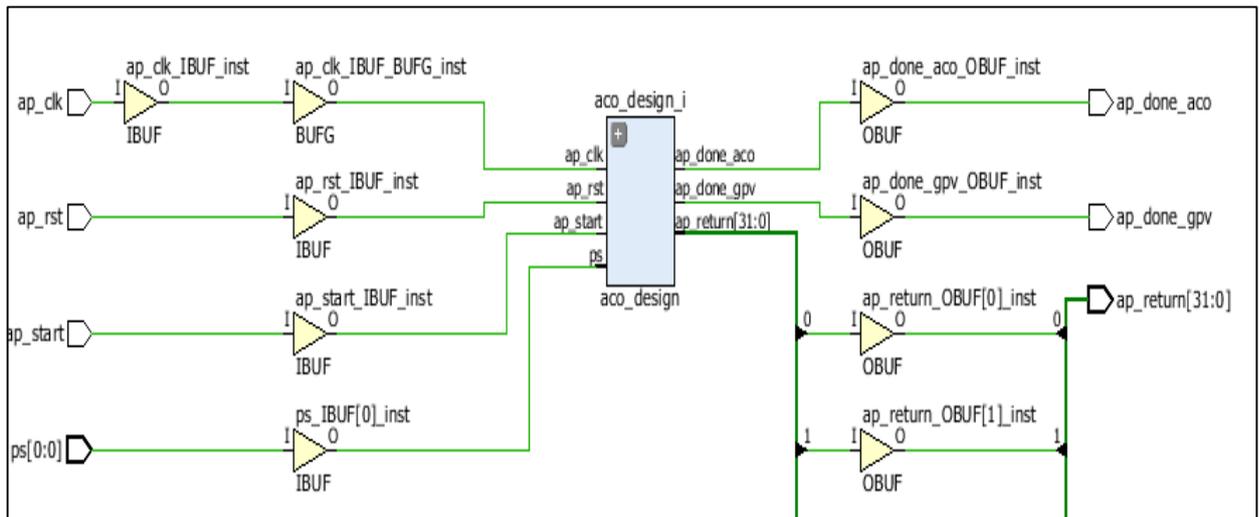


Figure.7.34. Architecture globale au niveau RTL après étape de synthèse.

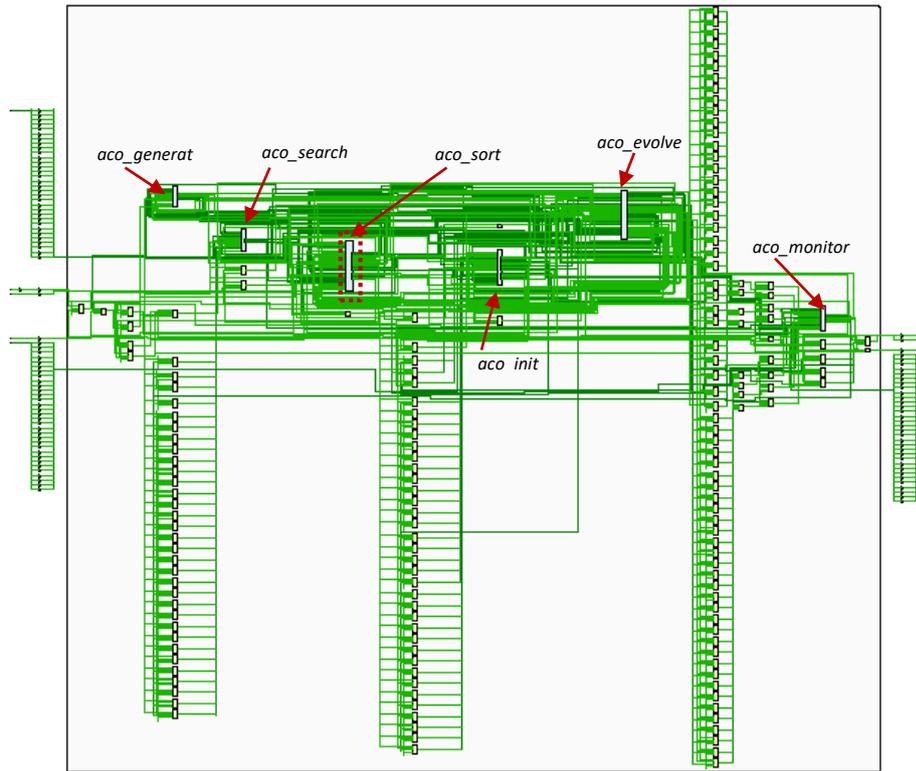


Figure.7.35. Schéma RTL des différents blocs constituant l'architecture.

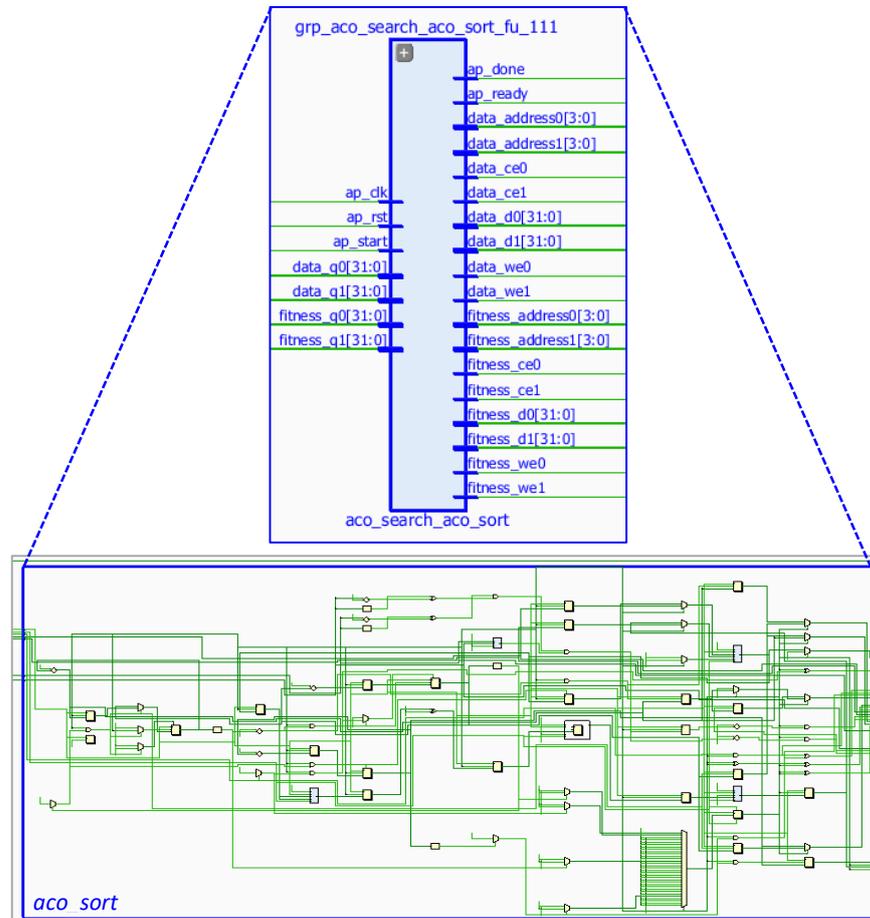


Figure.7.36. Détail de l'architecture de la fonction « *aco_sort* » au niveau RTL.

7.4.2.1.3 Implémentation de l'IP_ACO_NPU_MPPT

La dernière étape de développement est l'implémentation de l'architecture qui consiste à placer et router le design sur le circuit FPGA ciblé. L'outil va prendre en considération les contraintes et les caractéristiques physiques de la carte ciblée et faire l'implémentation du circuit sur cette dernière (placement, routage,...). Plusieurs optimisations sont aussi effectuées afin de réduire la consommation des ressources et assurer la synchronisation entre les différents composants. A la fin de cette étape, un fichier binaire dont l'extension est '.bit' est généré, il contient le code final à implémenter sur le circuit FPGA.

La figure.7.37. illustre l'espace occupé par le design après l'étape d'implémentation (placement/routage) sur circuit FPGA.

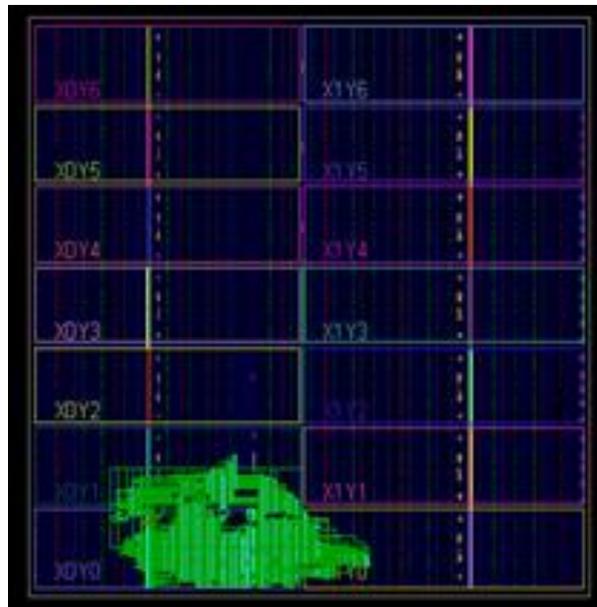


Figure.7.37. Implémentation de l'architecture sur circuit FPGA.

Le tableau.7.10 ci-dessous illustre une synthèse du taux d'utilisation des ressources du circuit après l'étape d'implémentation pour chaque fonction constituant l'architecture.

Tableau.7.10. Taux d'utilisation des ressources du circuit pour chaque fonction constituant l'architecture.

Name	Slice LUTs (303600)	Slice Registers (607200)	Block RAM Tile (1030)	DSPs (2800)	Bonded IOB (700)	BUFGCTRL (32)
aco_search_0	7853	4492	2	40	102	1
inst (aco_search)	7853	4492	2	40	0	0
aco_search_fcmp_32ns...	101	0	0	0	0	0
data_U (aco_search_data)	48	0	1	0	0	0
fitness_U (aco_search_...)	21	0	1	0	0	0
grp_aco_search_aco_ev...	4018	2344	0	21	0	0
grp_aco_search_aco_ini...	1526	889	0	7	0	0
grp_aco_search_aco_so...	2118	1113	0	12	0	0

7.4.2.2 Implémentation de l'IP *BSO_MPPT*

Pour l'implémentation hardware de l'IP *BSO_MPPT* généré, tout comme pour l'IP *ACO_NPU_MPPT*, nous adoptons la même approche avec les mêmes étapes qui seront appliquées. Il est à noter que certaines de ces étapes ne seront pas détaillées afin d'éviter de dupliquer l'information.

- Ajout de l'IP *BSO_MPPT* dans la bibliothèque « *User Repository / IP Catalog* » de l'outil Vivado (Figure 7.38) ;

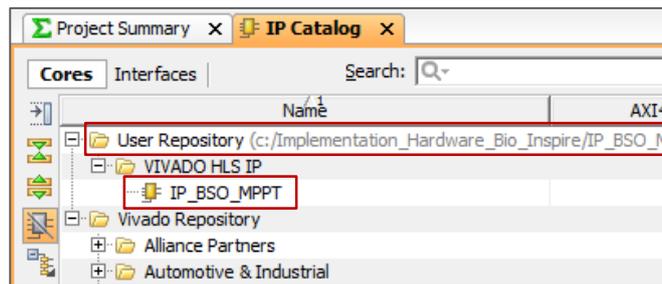


Figure 7.38. Intégration de l'IP «*BSO_MPPT*» dans *IP Catalog* la bibliothèque de l'outil Vivado.

- *Instanciation* de l'IP intégré et création d'un **Wrapper HDL** afin d'élaborer l'architecture globale du système constituée de l'IP *BSO_MPPT* et de l'IP *GPV* (Figure 7.39).

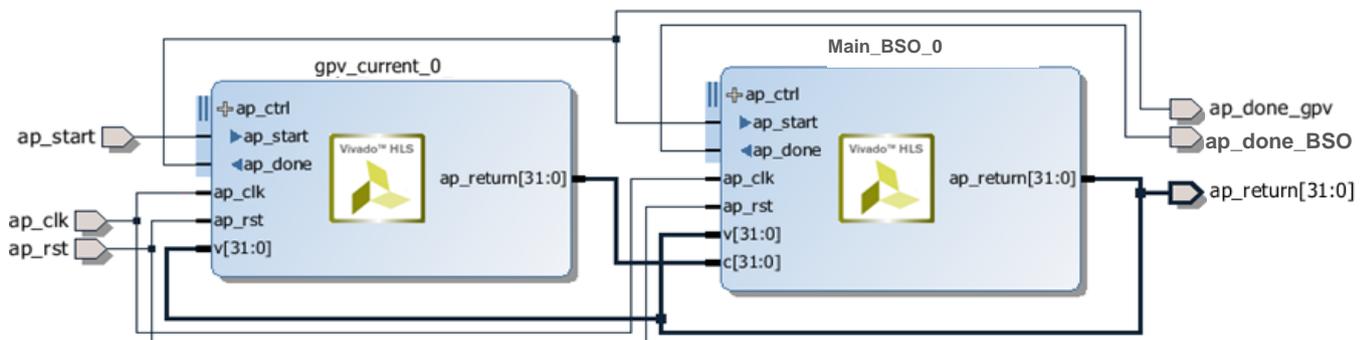


Figure 7.39. Architecture globale IP *BSO_MPPT* et IP *GPV* Wrappée.

7.4.2.2.1 Simulation de l'IP *BSO_MPPT*

Le système est simulé dans des conditions constantes d'éclairage et de la température. Tout comme pour l'IP *ACO_NPU_MPPT*, les signaux : « *ap_clk* » pour la génération du signal d'horloge est mis à 1, « *ap_rst* » est mis à 0, « *ap_start* » pour l'activation du bloc IP et la validation de l'entrée est mis à 1. En sortie, le signal « *ap_return* » illustre l'évolution de la phase de recherche du PPM. La Figure 7.40 illustre les résultats de simulation.

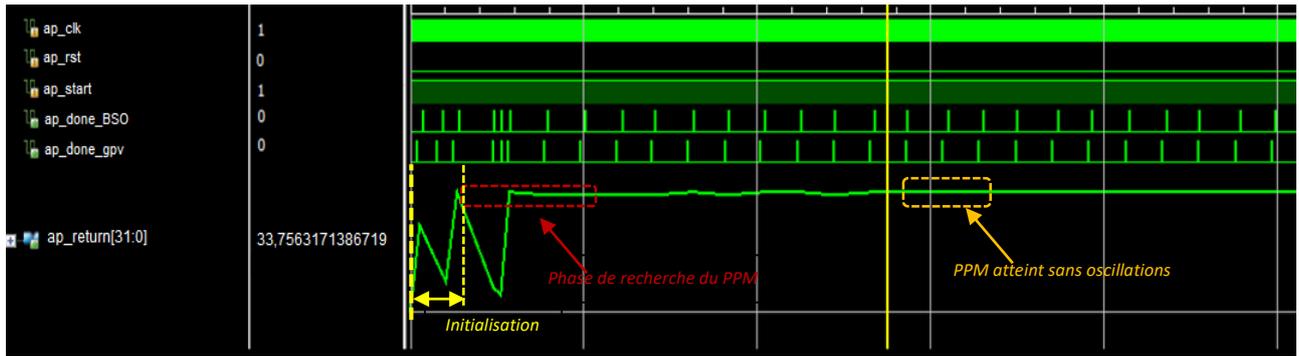


Figure.7.40. Simulation Comportementale l'IP_BSO_MPPT.

Au vu des résultats obtenus, le contrôleur *IP_BSO_MPPT* après une phase d'initialisation et de recherche du PPM, converge vers ce dernier où il se stabilise sans oscillations.

7.4.2.2.2 Synthèse de l'IP_BSO_MPPT

Les Figures.7.41 à 7.43 donnent respectivement une vue globale de l'architecture du système après l'étape de synthèse de l'architecture du système, la vue interne du schéma RTL des différents blocs constituant l'architecture, ainsi que le détail de l'architecture de la fonction « *best_solution* » au niveau RTL.

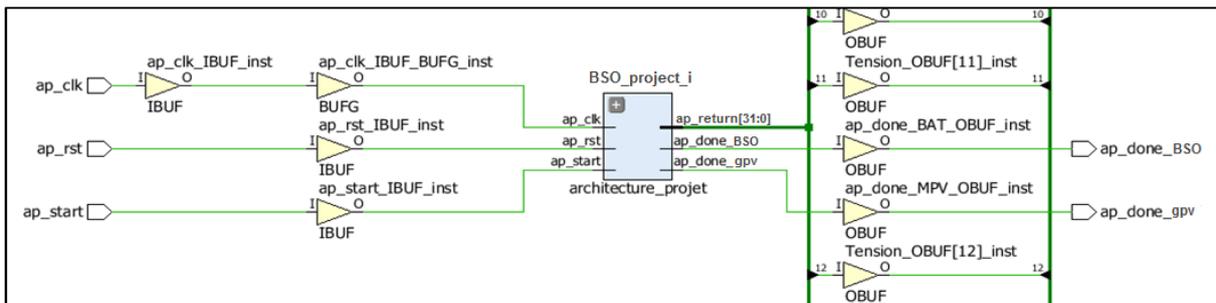


Figure.7.41. Architecture globale au niveau RTL après étape de synthèse.

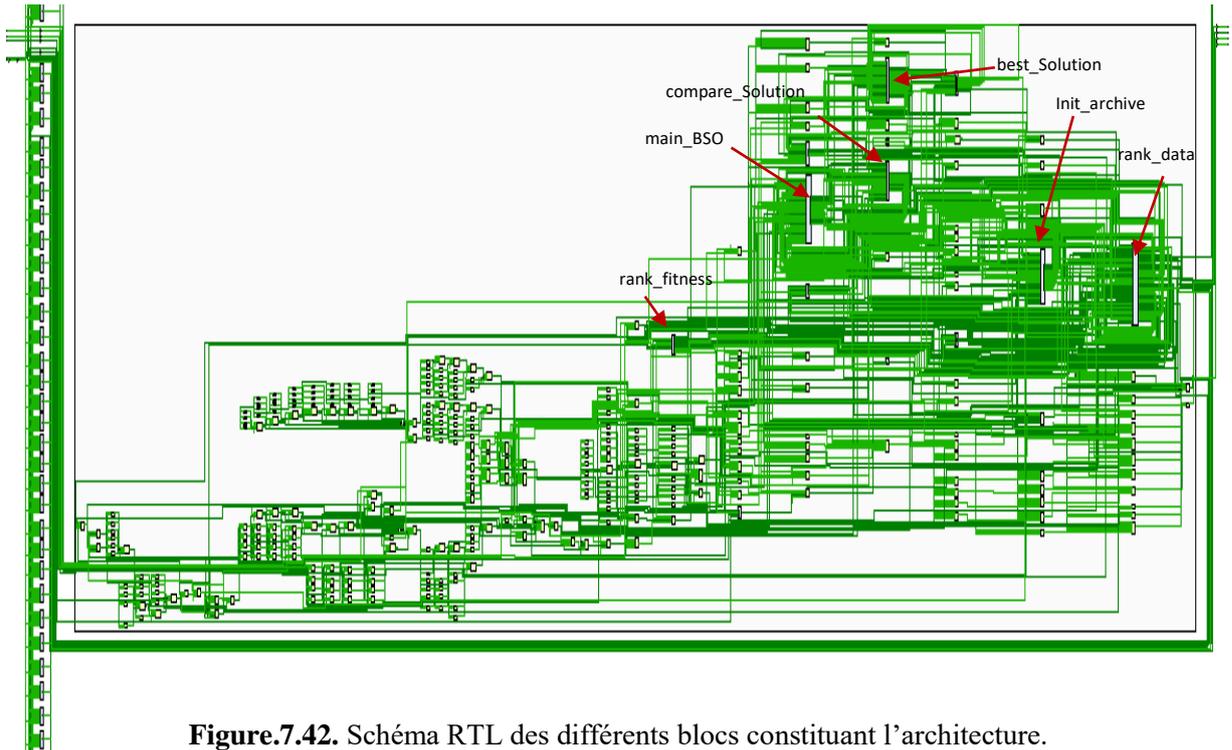


Figure.7.42. Schéma RTL des différents blocs constituant l'architecture.

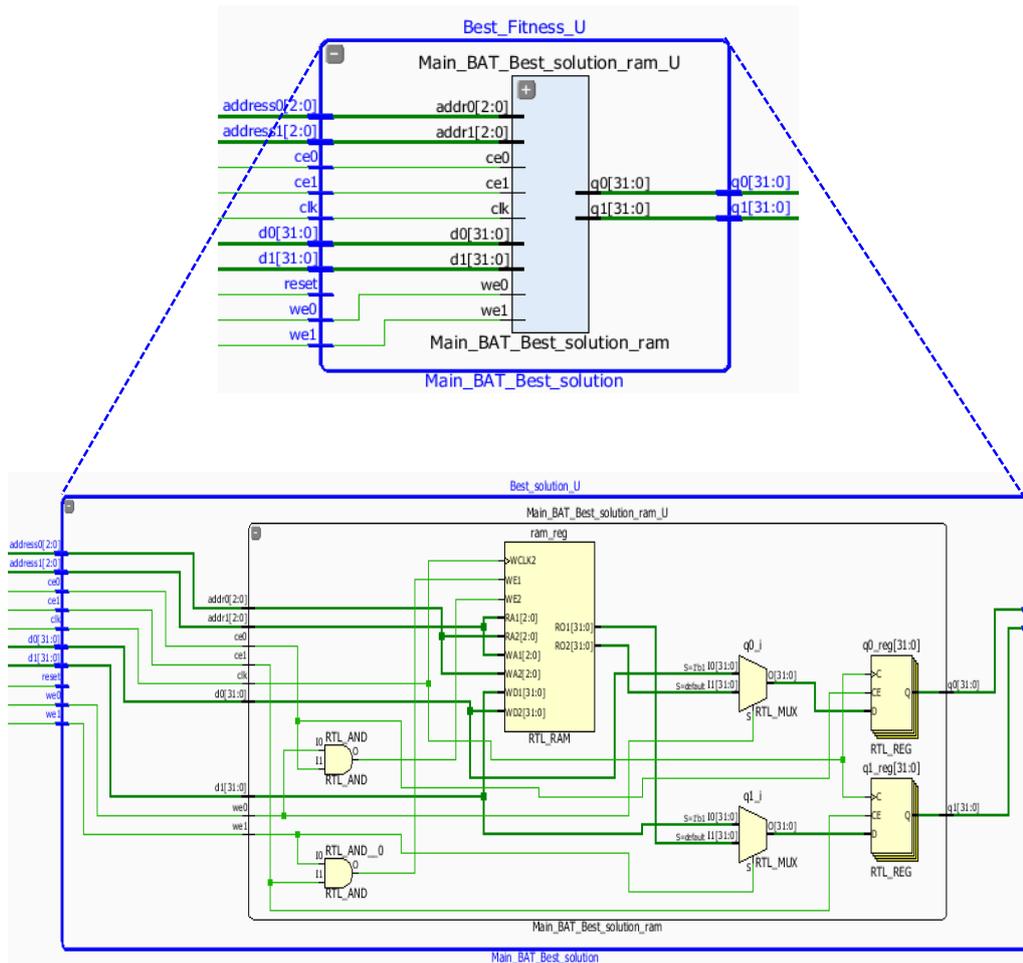


Figure.7.43. Détail de l'architecture de la fonction « Best_solution » au niveau RTL.

7.4.2.2.3 Implémentation de l'IP_BSO_MPPT

La dernière étape de développement qui consiste à implémenter l'architecture finale sur le circuit FPGA ciblé est illustrée par la figure.7.44.

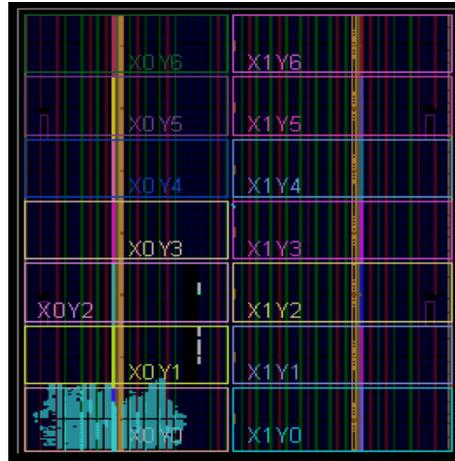


Figure.7.44. Implémentation de l'architecture du système PV sur circuit FPGA.

La Figure.7.45 illustre une synthèse relative à l'utilisation des ressources du circuit pour chaque fonction constituant l'architecture.

Name	Slice LUTs (303600)	Slice Registers (607200)	Block RAM Tile (1030)	DSPs (2800)	Bonded IOB (700)	BUFCTRL (32)
Main_BAT_0	2045	2475	4	4	102	1
inst (Main_BAT)	1949	2475	4	4	0	0
archive_U (Main_BAT_ar...)	168	0	1	0	0	0
Best_Fitness_U (Main_B...)	22	0	1	0	0	0
Best_solution_U (Main_B...)	1	0	1	0	0	0
Fitness_bat_U (Main_BA...)	42	32	0	0	0	0
Fitness_initial_U (Main_B...)	61	0	1	0	0	0
grp_Main_BAT_Classem...	114	85	0	0	0	0
grp_Main_BAT_Classem...	113	85	0	0	0	0
grp_Main_BAT_position...	1101	1593	0	2	0	0
grp_Main_BAT_Val_Alea...	32	94	0	2	0	0
Main_BAT_fadd_32ns_3...	0	32	0	0	0	0
Main_BAT_fmnl_32ns_3...	16	64	0	0	0	0
position_bat_U (Main_B...)	32	32	0	0	0	0

Figure.7.45. Synthèse des ressources utilisées pour chaque fonction du design.

7.4.3 Discussion

Les ressources logiques FPGA utilisées par les deux IPs sont illustrées dans le tableau.7.11.

Tableau.7.11. Ressources utilisées pour chaque IP.

Device Utilization Summary Virtex7 « xc7vx485tffg1766-2 »	Available	IP_ACO_NPU_MPPT	IP_BSO_MPPT
		Used	Used
Number of Slice LUTs	303600	7853 (~2.6%)	3651 (~1.20%)
Number of Slice Registers	607200	4492 (~0.74%)	2475 (~0.41%)
Number of DSP	2800	40 (~1.42%)	4 (~0.2%)
Number of BUFCS	32	1 (~3%)	1 (~3%)
Number of Block RAM	1030	2 (~0.19%)	4 (~0.4%)
Number of Bouded IOB	700	102 (~14.6%)	120 (~14.6%)

Il est à noter que la carte de prototypage ciblée *Virtex7* dispose d'un circuit FPGA de la Série 7 « xc7vx485tffg1766-2 », ce dernier dispose d'un nombre important de ressources, c'est ce qui explique que l'architecture des deux IPs implémentées ne consomment pas beaucoup de ressources du circuit.

Les figures 7.46 et 7.47 illustrent respectivement le taux d'utilisation des ressources matérielles ainsi que la répartition des différentes fonctions en termes de LUTs lors de l'implémentation de chacun des deux IPs développées.

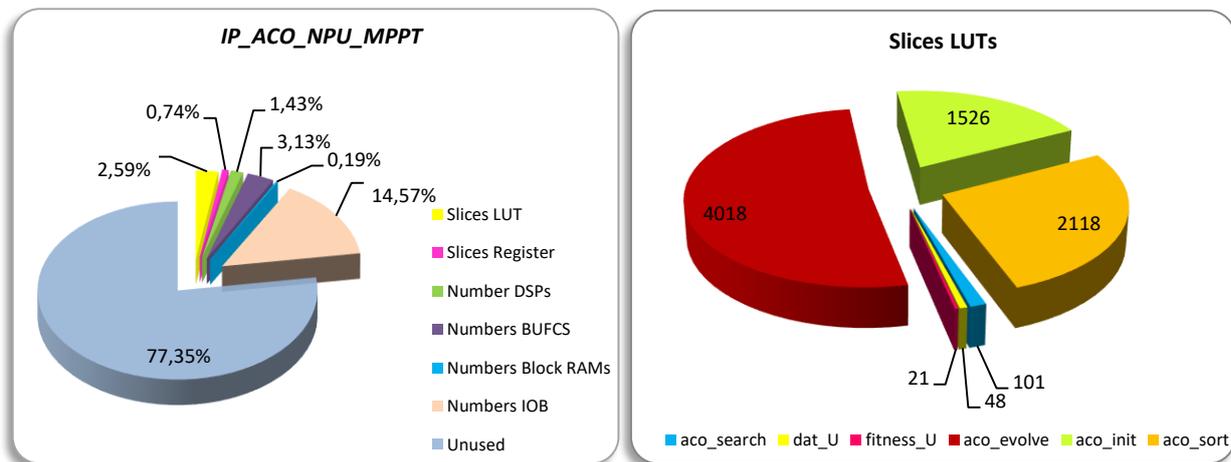


Figure.7.46. (a) Taux des ressources utilisées pour l'IP_ACO_NPU_MPPT ;
(b) Répartition des principales fonctions en termes de LUTs de l'IP ACO_NPU_MPPT.

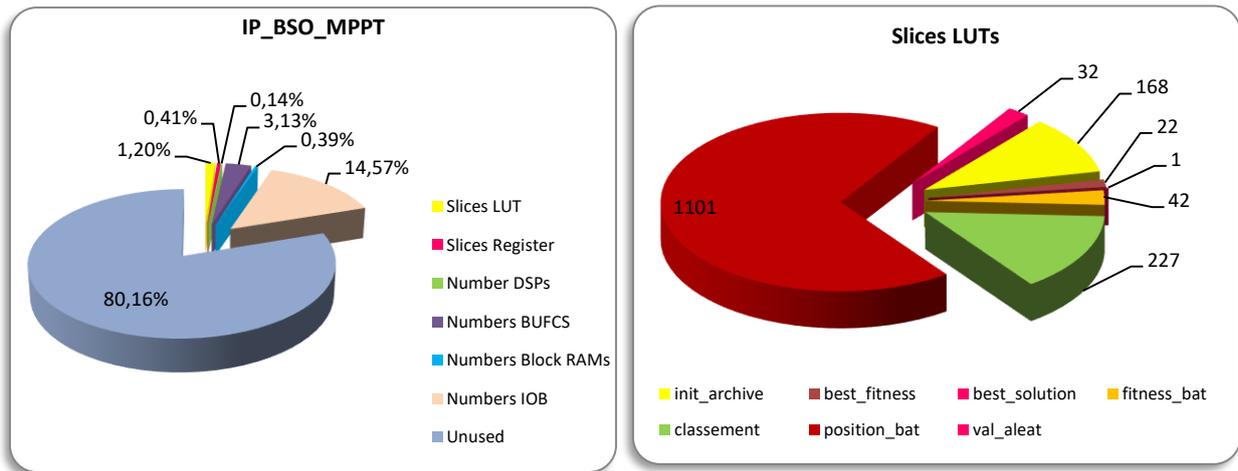


Figure.7.47. (a) Taux des ressources utilisées pour l'*IP_BSO_MPPT* ;
 (b) Répartition des principales fonctions en termes de LUTs de l'IP *BSO_MPPT*.

Les deux IP contrôleurs *IP_ACO_NPU_MPPT* et *BSO_MPPT* ainsi conçu ont subi toutes les étapes de conception et de développement depuis leur description au niveau algorithmique en C, jusqu'à leur implémentation sur circuit FPGA, en passant par la création et génération des IPs respectifs.

Les Figures.7.48 (a) et (b) illustrent la structure des directories relative à chaque IP. Ces dernières, à savoir « *csim* », « *impl* », « *sim* », « *syn* » sont générées automatiquement lors des différentes phases de développement.

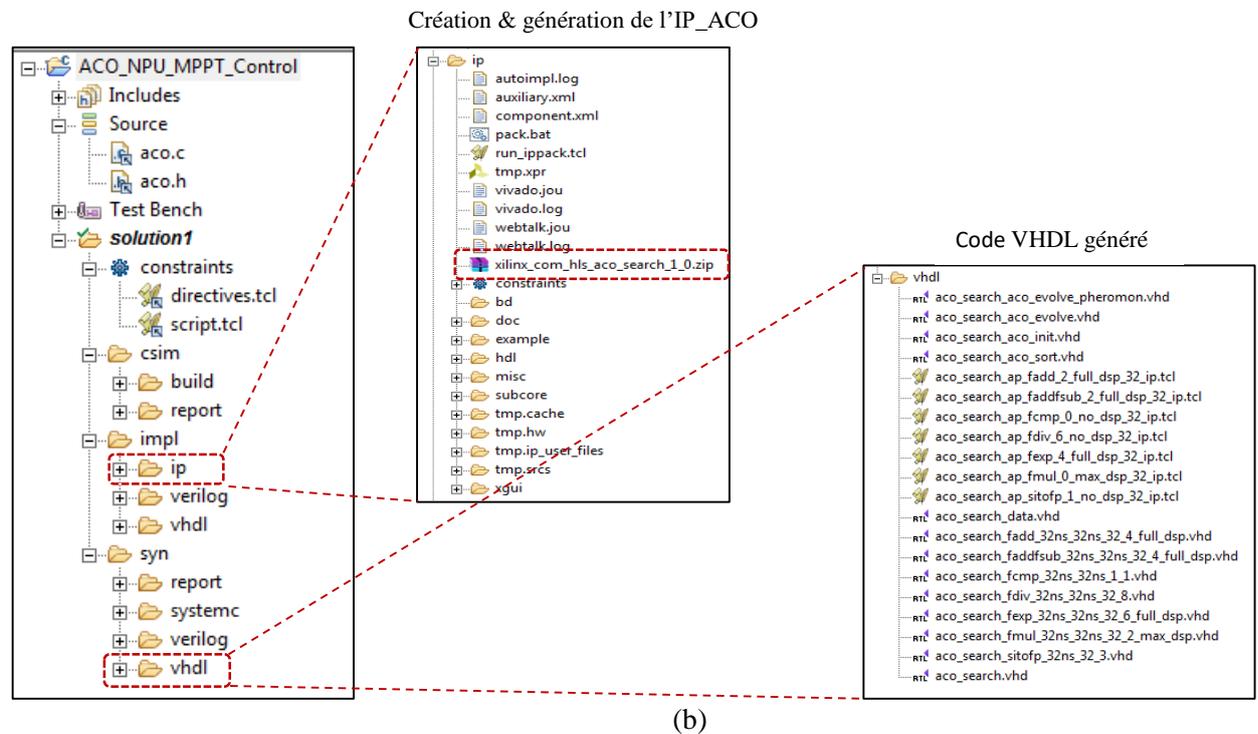
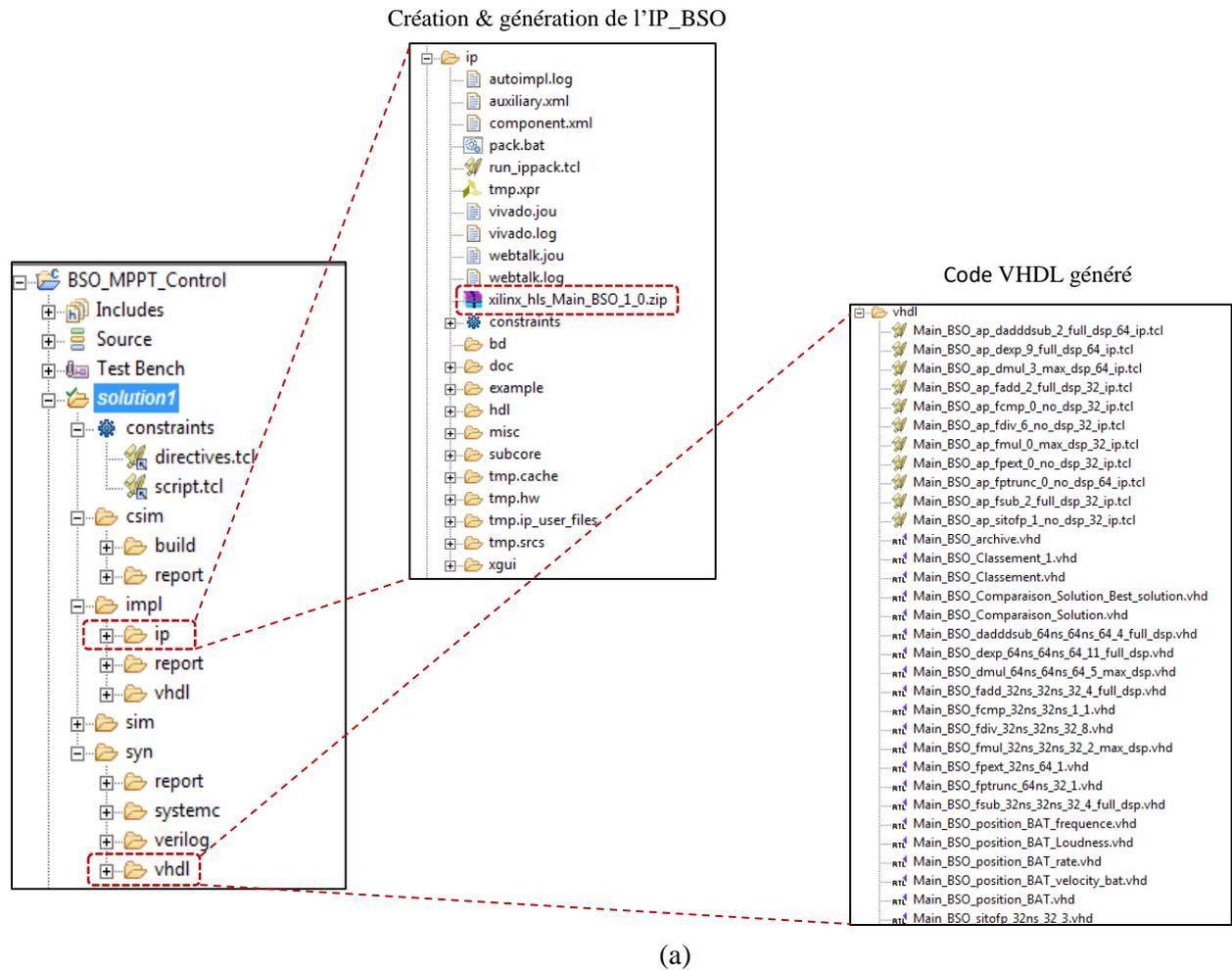


Figure.7.48. Structure des directories (a) IP_BSO_MPPT ;
(b) IP_ACO_NPU_MPPT

7.5 PROPOSITION D'UNE PLATEFORME HARDWARE POUR LE PROTOTYPAGE RAPIDE DES CONTRÔLEURS MPPT

7.5.1 Présentation de la plateforme

Afin de permettre au concepteur de palier au problème du temps énorme que prend la phase de développement des codes HDL (VHDL, Verilog), de simulation et de test lors de l'élaboration d'une application lorsque la complexité algorithmique augmente, il est important de mettre en place des plateformes de travail intégrant des outils qui permettent de passer d'un algorithme vers un prototype fonctionnel à tester sur circuit FPGA.

Néanmoins à ce jour, il n'existe pas de plateforme facilitant le prototypage rapide pour l'implémentation hardware et la génération de contrôleurs *IP_MPPT*.

C'est pour cela, que nous proposons une plateforme de développement hardware, permettant de passer d'un algorithme vers un prototype d'*IP_MPPT* fonctionnel.

Ainsi, la plateforme proposée facilite le test et l'implémentation hardware d'algorithmes pour le développement de contrôleurs MPPT, permettant ainsi le prototypage rapide et la génération d'*IP_MPPT* en validant rapidement des contrôleurs MPPT.

7.5.2 Architecture de la plateforme

La plateforme hardware proposée intègre les deux flots de conception *HDL Coder* de Matlab ainsi que *Vivado HLS* de Xilinx. Ces derniers permettent la génération automatique des codes HDL (VHDL, Verilog, SystemC) relatif aux contrôleurs MPPT. Par la suite, le flot *Vivado Design Suite* de Xilinx est exploité pour l'implémentation hardware des algorithmes générés en HDL jusqu'à la génération des contrôleurs *IP_MPPT*. Les *IP_MPPT* ainsi générés seront stockés dans la bibliothèque *IP_MPPT_Library*.

Les deux flots constituant la plateforme acceptent en entrée des descriptions à un haut niveau d'abstraction. Selon la description, le concepteur sélectionne l'outil à utiliser. Pour exemple, si le contrôleur à implémenter est modélisé sous *Matlab/Simulink* (.m, .mdl), l'outil *HDL Coder* sera exploité. En revanche, si le contrôleur est décrit en C, C++, alors le concepteur fera appel à l'outil *Vivado HLS*.

En parcourant toutes les étapes relatives à chacun des deux flots, le concepteur récupère en sortie un code HDL, minimisant ainsi le temps énorme alloué au codage manuel (VHDL, Verilog) surtout lorsque les algorithmes sont complexes.

La figure.7.49 illustre l'architecture de la plateforme hardware proposée.

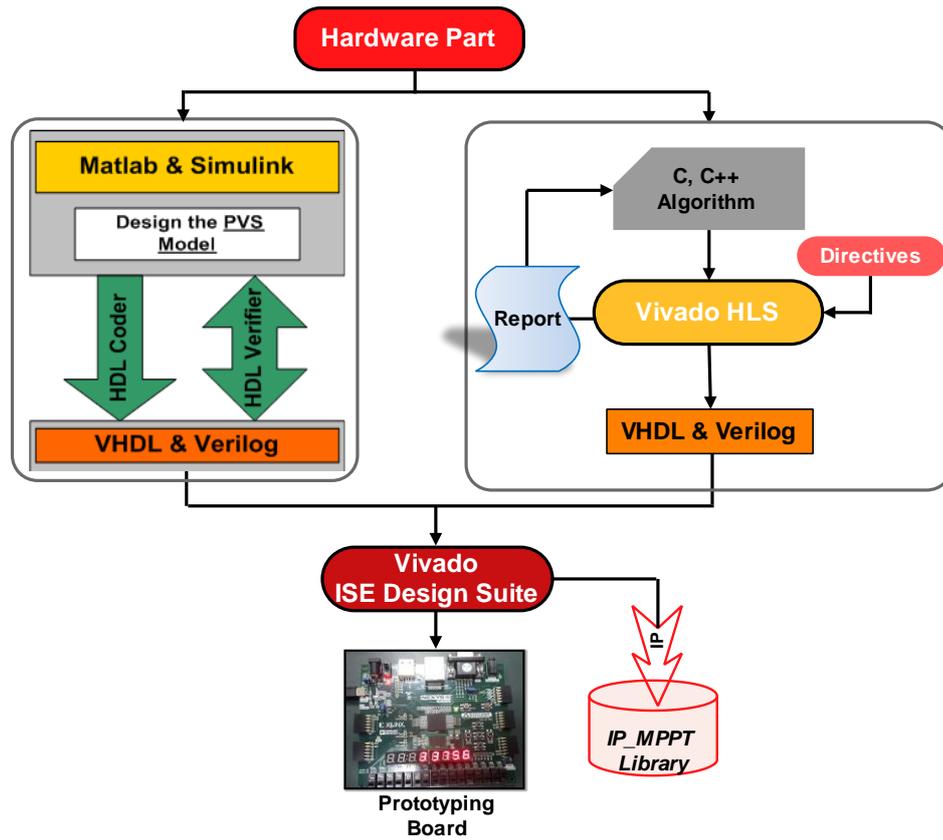


Figure.7.49. Architecture de la plateforme hardware.

7.6 Vers une plateforme globale Software/Hardware pour le prototypage rapide des contrôleurs *IP_MPPT*

Comme mentionné dans la section précédente, la nouvelle tendance consiste à utiliser des plateformes intégrant des outils modulaires, permettant l'automatisation de toutes les étapes du design en passant d'un algorithme vers un prototype fonctionnel à tester sur circuit FPGA.

Plusieurs travaux de recherche se sont penchés sur l'idée de proposer des plateformes pour le prototypage rapide, offrant ainsi aux concepteurs une méthodologie optimale de conception, de validation, de simulation et d'optimisation d'IPs à travers différents flots de conception.

La méthodologie intégrant ainsi les différents flots de conception offre un environnement de développement adéquat, réduisant considérablement les coûts et le temps de développement (*Time To Market*). Ainsi, le concept de réutilisation (*Design Reuse*) est favorisé, où un algorithme donné peut être utilisé pour l'exécution de plusieurs types d'applications.

Néanmoins, à ce jour, il n'existe pas de plateforme facilitant le prototypage rapide pour l'implémentation à la fois software et hardware pour la génération de contrôleurs *IP_MPPT*.

C'est dans ce but, que nous proposons comme perspective une plateforme évolutive conçue autour d'un ensemble d'outils modulaires qui permettent de passer d'un algorithme vers un prototype d'*IP_MPPT* fonctionnel tout en utilisant la même plateforme.

Ainsi, la plateforme proposée, permettra alors la mise en œuvre, le test et l'implémentation hardware d'algorithmes pour le développement de contrôleurs MPPT.

La prochaine section est consacrée à la présentation de ladite plateforme.

7.6.1 Présentation de la plateforme globale Software/Hardware

La figure.7.50 illustre l'architecture de la plateforme proposée.

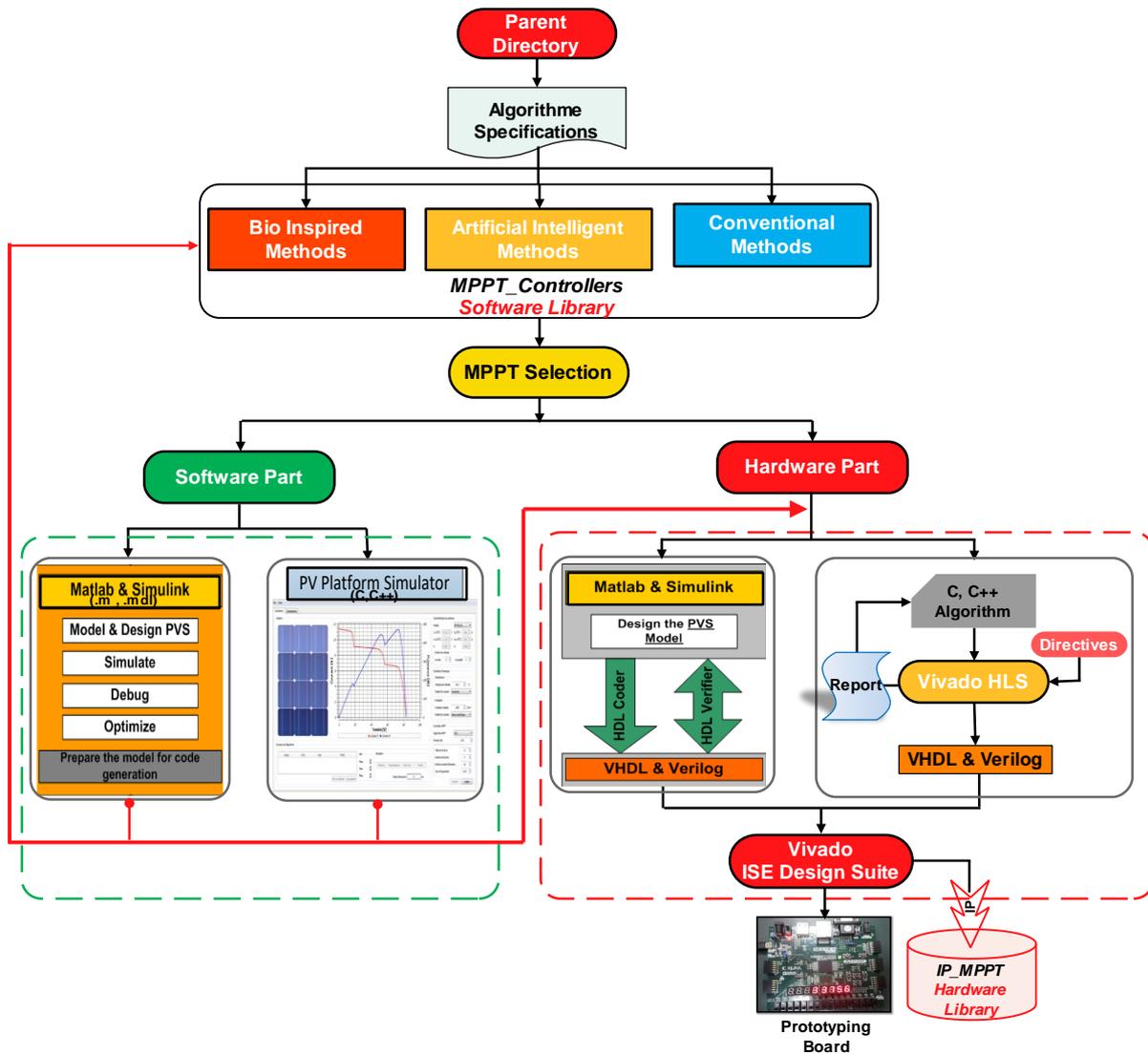


Figure.7.50. Architecture de la plateforme globale Software/Hardware.

Cette dernière est subdivisée en deux parties :

- Une partie software (*Software Part*), regroupant l'outil *Matlab* ainsi que la plateforme *PVSP* développée au chapitre.5. Ces derniers, permettent la mise en œuvre des algorithmes sélectionnés (*développement C, C++, .m, .mdl, paramétrage, test et simulation*) pour l'élaboration de contrôleurs MPPT adéquats.

- Une partie hardware (*Hardware Part*), regroupant les deux outils **Vivado HLS** (*High Level Synthesis*) de Xilinx et **HDL Coder** de Matlab utilisés pour la génération du code HDL des contrôleurs MPPT mis en œuvre et validés dans la partie software. Le troisième outil **Vivado Design Suite** de Xilinx, quant à lui, permet l'implémentation hardware des algorithmes générés précédemment en HDL jusqu'à la génération des contrôleurs *IP_MPPT*. Les *IP_MPPT* ainsi générés seront stockés dans la bibliothèque ***IP_MPPT_Library***.

7.6.2 Fonctionnement de la plateforme globale Software/Hardware

Partant des spécifications et exigences du contrôleur à implémenter, le concepteur sélectionne l'algorithme le plus approprié pour une application déterminée.

Le passage des spécifications algorithmiques à la génération des *IP_MPPT* contrôleurs se fait à travers une suite d'étapes relatives au flot de conception utilisé.

Comme illustré en figure.7.50, combien même l'algorithme soit issu des méthodes conventionnelles, intelligentes ou bio inspirées, il n'en demeure pas moins que ce dernier suivra les mêmes étapes de développement depuis sa mise en œuvre au niveau algorithmique (*Software Part*), jusqu'à son implémentation et sa génération au niveau IP (*Hardware Part*).

La mise en œuvre, l'implémentation et la génération d'un *IP_MPPT* se fait en deux grandes étapes distinctes :

- L'étape 1 « **Software Part** » consiste en la mise en œuvre de l'algorithme en utilisant soit les outils **Matlab/Simulink** ou la plateforme **PVSP** développée et présentée à cet effet au chapitre.5. Ces derniers intègrent toutes les phases nécessaires pour le développement d'algorithmes et simulations de contrôleurs MPPT soumis à diverses contraintes environnementales (éclairage, température, problème d'ombrage partiel...etc.).

Le concepteur commence par :

1. *Un codage en, C, C++, .m, .mdl de l'algorithme selon les spécifications requises;*
2. *Un paramétrage de l'algorithme (sélection des meilleurs paramètres) ;*

3. *Des tests et simulations relatifs aux critères de convergence, stabilité, précision, robustesse face aux variations des conditions climatiques.*

Cette manière de faire, demeure à l'heure actuelle la meilleure approche visant à évaluer les contrôleurs MPPT en analysant leur comportement, permettant ainsi de sélectionner le plus approprié à l'application déterminée/ciblée ;

- L'étape 2 « **Hardware Part** » quant à elle, consiste en la génération automatique du code *HDL* relatif au contrôleur en vue de la création d'un *IP_MPPT* contrôleur grâce aux flots de conception **HDL Coder** de Matlab ou **VIVADO HLS** de Xilinx. Suite à cela, toutes les étapes relatives à l'implémentation hardware (*synthèse, placement routage, etc..*) du contrôleur sur circuit FPGA sont réalisées en utilisant le flot de conception **Vivado Design Suite** de Xilinx.

Le concepteur commence par :

1. *Récupérer le codage en (C, C++, .m, .mdl) de l'algorithme élaboré et validé dans la partie software « **Software Part** » de la plateforme;*
2. *Appliquer les directives d'optimisations ;*
3. *Génération du code HDL (**Vivado HLS** ou **HDL Coder**) ;*
4. *Création de l'IP ;*
5. *Implémentation hardware (synthèse, implémentation, placement, routage, génération .bit...).*

La plateforme de prototypage ainsi décrite dans son fonctionnement, fournit de multiples avantages résumés ci-après :

Les flots de conception contenues dans la partie software « **Software Part** » possèdent une flexibilité au niveau de la programmation facilement modifiable. Il s'agit de codes sources standards. Les codes en entrée sont décrits à un haut niveau *C, C++, .m, .mdl*.

De plus, grâce aux flots de conception contenus dans la partie hardware « **Hardware Part** », l'approche *(C, C++, .m, .mdl)-vers-HDL(VHDL, Verilog, System C, etc...)* facilite grandement le prototypage rapide de contrôleur MPPT sur cible FPGA.

Ainsi, un concepteur n'ayant pas de connaissances approfondies des aspects liés à la conception hardware (langage de description *VHDL, Verilog*, architecture interne des *FPGA*),

peut désormais développer des contrôleurs MPPT en utilisant des descriptions élaborées à un haut niveau d'abstraction et validées dans la partie software de la plateforme.

La connaissance d'un langage de description HDL n'est désormais plus indispensable vu que la génération du code (*VHDL*, *Verilog*, *SystemC*) est automatique. Ceci constitue un facteur clé pour encourager l'utilisation des circuits FPGA.

En offrant une souplesse de programmation d'une part, et en permettant l'exploitation de la bibliothèque *IP_MPPT_Library* contenant les *IP_MPPT* contrôleurs développés en vue d'une réutilisation par d'autres concepteurs (*Design Reuse*) d'autre part, la plateforme proposée réduit considérablement le temps de conception et donc le *Time to Market*. Elle permet ainsi d'atteindre rapidement l'objectif fixé concernant le prototypage.

7.7 Conclusion

Tout au long de ce chapitre nous avons présenté les approches de conception classiques, l'évolution de ces dernières, suivie de la nouvelle approche de conception basée sur la synthèse de haut niveau (HLS).

Nous avons également détaillé la méthodologie adoptée pour l'implémentation des deux IPs : *IP_ACO_NPU* et l'*IP_BSO* sur circuit FPGA selon l'approche de synthèse de haut niveau, avec les outils Vivado et Vivado HLS. Ce dernier réduit considérablement le temps de développement de l'algorithme en automatisant la génération de la description matérielle du circuit.

Aussi, nous avons proposé une plateforme hardware facilitant le prototypage rapide de contrôleur MPPT. Il est intéressant de noter que tout comme pour la plateforme *PVPS* présentée au chapitre.5, la plateforme hardware proposée pour le prototypage rapide de contrôleurs MPPT garde, elle aussi, un aspect évolutif. En effet, la bibliothèque *IP_MPPT_Library* de la plateforme contient une liste non exhaustive de contrôleurs *IP_MPPT*. Cette dernière peut être enrichie continuellement, en intégrant de nouveaux contrôleurs facilitant ainsi le prototypage rapide d'applications liées aux énergies renouvelables en adoptant le concept du *Design Reuse*.

Enfin, en combinant les deux aspects de conception au niveau software et hardware sur la même plateforme, ceci va faciliter grandement le travail des concepteurs qui œuvrent dans la génération de contrôleurs *IP_MPPT*.

De plus, cela va beaucoup aider à vulgariser la conception car la plateforme constitue en soi un outil pédagogique innovant dans l'enseignement ainsi que pour la communauté scientifique dans leurs travaux de recherche qui portent sur l'optimisation de puissance, notamment dans les systèmes et les applications photovoltaïques.

CONCLUSION GÉNÉRALE

L'objectif des travaux réalisés tout au long de cette thèse s'inscrit dans le cadre de recherche de solutions pour l'optimisation de puissance dans les systèmes photovoltaïques.

Notre démarche a consisté à explorer les différentes méthodes d'optimisation existantes, de sélectionner une approche visant à élaborer un contrôleur permettant d'améliorer l'étage d'adaptation entre le générateur photovoltaïque et sa charge, de sorte à avoir, en permanence, une puissance générée optimale.

Cette exploration nous a conduit à examiner dans le détail les performances des algorithmes bio-inspirés pour traiter le problème d'extraction et de poursuite de la puissance maximale (MPPT) d'un générateur photovoltaïque. Parmi ces algorithmes, le choix s'est porté sur l'algorithme de colonies de fourmis (ACO_NPU) et celui de la Chauve-Souris (BSO).

Ce choix s'est avéré judicieux puisque depuis la conception des deux contrôleurs jusqu'à leur implémentation hardware sur circuit programmable FPGA, les résultats obtenus ont été concluants. Il est utile de préciser ici que l'implémentation hardware a été réalisée en mettant en œuvre une nouvelle approche de conception basée sur la synthèse de haut niveau (HLS).

La justesse des choix a été confortée par la mesure des performances des contrôleurs bio inspirés. En effet, ces derniers présentent des performances qui sont de loin plus importantes comparativement avec les contrôleurs issus des Méthodes Conventionnelles et les Méthodes Intelligence Artificielles.

Les observations que nous avons pu mener sous conditions standard ($1000\text{W}/\text{m}^2$, 25°C) ont démontré une amélioration palpable du temps de réponse pour atteindre le PPM tout en faisant en sorte que les oscillations autour du PPM soient quasi nulles.

L'efficacité des deux contrôleurs bio inspirés a été par la suite confirmée, une nouvelle fois, par les multiples simulations exécutées sous différentes conditions environnementales (Température, Eclairage, Ombrage).

Il est important de préciser que la conception HLS adoptée pour l'implémentation hardware de ces contrôleurs, consiste à générer automatiquement du code *HDL* (*VHDL*, *Verilog*, *System C*...etc.) à partir d'une description haut niveau, réduisant ainsi considérablement le temps de conception.

L'approche est subdivisée en deux parties : la première consiste en la création d'un IP généré automatiquement à partir du code C de l'algorithme introduit. Pour la seconde, l'IP ainsi créé est intégré dans le flot de conception *Vivado ISE* en vue d'une implémentation hardware.

Toujours dans le même souci de réduire le temps de conception, nous avons proposé deux plateformes de prototypage, la première, dédiée pour le développement et l'évaluation des commandes MPPT, la seconde, quant à elle, est destinée au prototypage rapide de ces contrôleurs en vue d'une implémentation hardware. L'efficacité des deux plateformes a été prouvée par une étude de cas.

Enfin, avant de conclure, il est utile de rappeler, que parmi les réalisations effectuées dans le cadre de ce travail de recherche, la plateforme « *PVSP* » réalisée, constitue dans l'absolu, un outil très utile pour la communauté scientifique dont les travaux de recherche portent sur le développement de nouveaux contrôleurs MPPT, particulièrement en facilitant les tests et les simulations avec un prototypage rapide.

Comme perspectives pour cette plateforme, grâce à son caractère portable ainsi élaboré, cette dernière permet d'ores et déjà, de concevoir et d'implémenter de nouveaux contrôleurs MPPT, qui pourraient être intégrés dans la bibliothèque (*IP_Catalog*) et être disponibles pour la communauté des chercheurs (*Designer Reuse*) pour d'autres types d'applications dans divers domaines de recherche.

En combinant les deux aspects de conception au niveau software et hardware sur la même plateforme, cela va faciliter grandement la tâche des concepteurs qui travaillent dans la génération de contrôleurs *IP_MPPT*.

De plus, cela va grandement contribuer à vulgariser la conception et le développement de contrôleurs IP_MPPT vu que la plateforme constitue, en soi, un outil pédagogique innovant que ce soit dans l'enseignement ou pour la recherche scientifique qui traitent des thématiques liées à l'optimisation de puissance, notamment dans les systèmes et les applications photovoltaïques.

Enfin, comme perspectives générales, nous suggérons d'explorer davantage les possibilités de développement de scripts visant à optimiser les fonctionnalités de la plateforme, ainsi que le développement d'autres contrôleurs IP_MPPT qui prendraient en compte la problématique de l'ombrage partiel et qui vont enrichir encore plus la bibliothèque. Ces deux pistes constituent, à notre sens, des pistes de recherche prometteuses.

BIBLIOGRAPHIE

- [1] A. F. Boehinger, "Self -adaptive DC converter for solar spacecraft power supply", IEEE Transactions on Aerospace and Electronic Systems, AES, pp. 102-111, 1968.
- [2] H. Mathieu, "Physique des semi-conducteurs et des composants électroniques", 6ème édition (Dunod), France, 2009.
- [3] H.J.Möller, Semiconductors for Solar Cells, Artech House, Inc, Norwood, MA, 1993.
- [4] H. Knopf, "Analysis, simulation, and evaluation of maximum power point tracking (MPPT) methods for a solar powered vehicle", Master of Science in Electrical and Computer Engineering, Portland State University, 1999.
- [5] J.P Charles, A. Haddi, et al, "The junction, of solar with microelectronics", Rev. Energ. Ren. Vol.3, pp 1-16, 2000.
- [6] M.S. Imamura, J.I. Portscheller, "Year Evaluation of the Methods of Determining Solar Cell Series Resistance, Proc. 8 HT Photovoltaic Specialists' Conf., Seattle, WA, pp 102-107, 1970.
- [7] R. Gottschalg, M. Rommel, D.G. Ineld, H. Ryssel, Comparison of different methods for the parameter determination of the solar cells double exponential equation. In 14th European Photovoltaic Science and Engineering Conference (PVSEC), Barcelona, Spain, 1997.
- [8] T.Mambrini, "Caractérisation de panneaux solaires photovoltaïques en conditions réelles d'implantation et en fonction des différentes technologies", Thèse de doctorat, Université Paris-Sud, 2014.
- [9] F. Belhachat, C. Larbes, "Modeling, analysis and comparison of solar photovoltaic array configurations under partial shading conditions", Solar Energy, Vol.120, pp.399-418, 2015.
- [10] M.Z.M Ahmed Shams El-Dein, "Novel Designs for Photovoltaic Arrays to Reduce Partial Shading Losses and to Ease Series Arc Fault Detection", PHD Thesis, Waterloo, Ontario, Canada, 2012.
- [11] Priyanka O Singh, "Modeling of Photovoltaic Arrays under Shading Patterns with Reconfigurable Switching and Bypass Diodes", Master of Science thesis, University of Toledo, 2011.
- [12] D. Picault , B. Raison , S. Bacha , J. de la Casa , J. Aguilera, "Forecasting photovoltaic array power production subject to mismatch losses", Solar Energy, Vol.84, pp.1301-1309, 2010.
- [13] A. Houari, "Contribution à l'étude de micro-réseaux autonomes alimentés par des sources photovoltaïques", Thèse de doctorat, Université de Lorraine, 2012.
- [14] A.C.Pastor, "Conception et réalisation de modules photovoltaïques électroniques", Thèse de Doctorat, INSA de Toulouse, 2006.
- [15] C.Cabal, "Optimisation énergétique de l'étage d'adaptation électronique dédié à la conversion photovoltaïque", Thèse de Doctorat, Université Toulouse, 2008
- [16] N.Karami, N.Moubayed, R.Outbib, "General Review and Classification of Different MPPT Techniques", Renewable and Sustainable Energy Reviews, 2017, Vol.68, 1-18, <http://dx.doi.org/10.1016/j.rser.2016.09.132>.
- [17] N.Onat, "Recent developments in maximum power point tracking technologies for photovoltaic systems", International journal of photo energy, Vol.2010, doi:10.1155/2010/245316.
- [18] M.D.Goudar, B.P.Patil, V.Kumar, "A review of improved maximum power tracking algorithms for photovoltaic systems", International journal of electrical engineering and technology IJEET, ISSN 0976-6553, Vol.1, N° 1, june 2010, pp85-107.
- [19] N.Karamadi, N.Moubayed, et al, "General review and classification of different MPPT techniques", Renewable and Sustainable Energy Reviews, 2017, Vol.68, 1-18, <http://dx.doi.org/10.1016/j.rser.2016.09.132>.
- [20] A.Amir, J.Selvaraj, et al, "Study of MPP tracking algorithms: Focusing the numerical method techniques", Renewable and Sustainable Energy Reviews, 2016, Vol.62, 350-371, <http://dx.doi.org/10.1016/j.rser.2016.04.039>
- [21] D.Verma, S.Nema, A.M.Shandilya, S.K.Dash, "Maximum Power Point Tracking (MPPT) Techniques : Recapitulation in Solar Photovoltaic Systems", Renewable and Sustainable Energy Reviews, 2016, Vol.54, 1018-1034, <http://dx.doi.org/10.1016/j.rser.2015.10.068>.
- [22] K.Sobaihi, "Etude et Réalisation d'un Hacheur de Tracking du Point de Puissance Maximale (MPPT) à Contre Réaction de Tension", Mémoire de Magistère, Ecole Nationale Polytechnique (ENP), Département Electronique, Laboratoire des Dispositifs de Communications et de Conversion Photovoltaïque (LDCCP), 2005.

- [23] P. Bhatnagar, R.K Nema, "Maximum Power Point Tracking Control Techniques : State of the Art in Photovoltaic Applications", *Renewable and Sustainable Energy Reviews*, 2013, Vol.23, 224-241, <http://dx.doi.org/10.1016/j.rser.2013.02.011>.
- [24] Priety et Y.Kumar, "Review paper on various types of techniques for PV system", *International Journal of Engineering & Science Research*, IJESR/May 2014/ Vol-4/Issue-5/320-330, e-ISSN 2277-2685, p-ISSN 2320-976.
- [25] A.R.Reisi, M.H.Moradi, et al, "Classification and comparison of maximum power point tracking techniques for photovoltaic system : A review", *Renewable and Sustainable Energy Reviews*, 2013, Vol.19, 433-443, <http://dx.doi.org/10.1016/j.rser.2012.11.052>.
- [26] A. Gupta, Y.K.Chauhan, et al," A comparative investigation of maximum power point tracking methods for solar PV system", *Solar Energy*, 2016, Vol.136, 236-253, <http://dx.doi.org/10.1016/j.solener.2016.07.001>.
- [27] I. Kashif, S. Zainal, "A Review of Maximum Power Point Tracking Techniques of PV System for Uniform insolation and Partial Shading Conditions", *Renewable and Sustainable Energy Reviews*, 2013, Vol.19, 475-488, <http://dx.doi.org/10.1016/j.rser.2012.11.032>
- [28] G. Dileep, S.N. Singh, "Application of soft computing techniques for maximum power point tracking of SPV system", *Solar Energy*, 2017, Vol.141, 182-202, <http://dx.doi.org/10.1016/j.solener.2016.11.034>.
- [29] M.Balamurugan, S.K.Sahoo, et al, "Application of soft computing methods for grid connected PV system: A technological and status review", *Renewable and Sustainable Energy Reviews*, 2016, <http://dx.doi.org/10.1016/j.rser.2012.11.210>.
- [30] M.Seyedmahmoudian, B.Horan, et al, "State of the art artificial intelligence based MPPT techniques for mitigating partial shading effects on PV systems: A review", *Renewable and Sustainable Energy Reviews*, 2016, Vol.64, 435-455, <http://dx.doi.org/10.1016/j.rser.2016.06.053>.
- [31] Z.Salam, A.Jubaer, et al, "The application of soft computing methods for MPPT of PV system: a technological and status reviews", *Applied Energy*, 2013, Vol.107, 135-148, <http://dx.doi.org/10.1016/j.apenergy.2013.02.008>.
- [32] V. Salas, E. Olàs, et al, "Review of the maximum power point tracking algorithms for stand-alone photovoltaic systems", *Solar Energy Materials & Solar Cells*, 90, 2006, pp. 1555-1578.
- [33] L.Merwe, G.Merwe, "Maximum power point tracking implementation strategies", *Proceedings of 1998 IEEE International Symposium on Industrial Electronics (ISIE'98)*, 1998, pp. 214-217.
- [34] M.Veerachary, T.Senjyi, et al, "Voltage based maximum power point tracking control for PV systems", *IEEE Transaction on Aerospace and electronic Systems*, Jan 2002, 38, 262-270.
- [35] C.Lee.Sanghoey, "Design and implementation of photovoltaic power conditioning system using a current based MPPT", *ICSET, IEEE*, 2008, 295-300.
- [36] H.T.Duru, "MPPT algorithm based on $I_{mp} = f(P_{max})$ function for matching passive and active loads to a photovoltaic generator", *Solar energy*, Vol. 80, N°.7, July 2006, 812-822, doi: 10.1016/j.solener.2005.05.016.
- [37] N.Femia, D.Gronozio, "Predictive and adaptative MPPT perturb and observe method", *IEEE Transaction on Aerospace and Electronic Systems*, 2007, Vol.43, 934-950.
- [38] N.Femia, G.Ptrone, et al, "Optimization of Perturb & Observe maximum power point tracking method", *IEEE Power Electron*, 2005, Vol. 48, 963-973.
- [39] X.Liu, L.Lopez, et al, "An improved perturbation and observation MPPT algorithm for PV arrays", *35th IEEE Power Electronics Specilists Conference*, 2004, 2005-2010.
- [40] F.Liu,S.Duan, et al, "A variable step size Inc MPPT method for PV systems", *IEEE transactions on industrial electronics*, Vol.55, N°.7, july2008.
- [41] Z.Yan, L.Fei, et al, "Study on realizing MPPT by improved incremental conductance method with variable step size", *Huazhong university of science and technology, Hubei province, China*, 2008.
- [42] A.Safari, S.Mkhilef, et al, "Simulation and hardware implementation of incremental conductance MPPT with direct control method using cuck converter", *Industrial Electronics, IEEE transactions*, 2011, Vol.58, 1154-1161.
- [43] L.A.Zadeh, "Soft computing and fuzzy logic", *IEEE Software*, 11(6), pp 48-56, 1994.
- [44] L.A.Zadeh, "Fuzzy Logic, Neural networks, and soft computing", *Communication of the ACM*, Vol.37, N°.3. March 94.

- [45] A.Youssef, M.El-Telbany, et al, "The role of artificial intelligence in PV systems design and control : A review", *Renewable and Sustainable Energy Reviews*, Vol.78, pp72-79, 2017, <http://dx.doi.org/10.1016/j.rser.2017.04.046>
- [46] A.Mellit, S.K. Kalougirou, "MPPT based artificial intelligence techniques for photovoltaic systems and its implementation into field programmable gate array chips: Review of current status and future perspectives", *Energy*, Vol.70, 1-21, <https://doi.org/10.1016/j.energy.2014.03.102>.
- [47] Y.H.Liu, C.L.Liu, et al,"Neural Network Based MPPT Methods for Photovoltaic Systems Operating Under Fast Changing Environments, *Solar energy*, 2013, Vol.89, 42-53.
- [48] A.K. Rai,N.D. Kaushika, et al, "Simulation Model of ANN Based MPPT Controller for Solar PV System", *Solar Energy Materials and Solar Cells*, 2011, Vol.95, 773-778.
- [49] M.S.Ait Cheikh, M.Haddadi, et al, "Design of a Neural Network Control Scheme for the Maximum Power Point Tracking", *Revue des energies Renouvelables*, 2007, Vol.10, N°1, 109-118.
- [50] R.M.Essefi, M.Souissi,H. Abdallah, "Maximum Power Point Tracking Control Using Neural Networks for Stand-Alone Photovoltaic Systems", *International Journal of Modern Nonlinear Theory and Application*, 2014, 3,53-65, <http://dx.doi.org/10.4236/ijmnta.2014.33008>.
- [51] D.Vasarevicius, R.Martavicius, M.Pikutis, "Application of Artificial Neural Networks for Maximum Power Point Tracking of Photovoltaic Panels". *Electronika IR Electrotechnika*, ISSN 1392-1215, Vol. 18, N°10, 2012, <http://dx.doi.org/10.5755/j01.eee.18.10.3065>
- [52] M.S.Ait Cheikh, C.Larbes, et al," Maximum Power Point Tracking Using a Fuzzy Logic Control Scheme", *Revue des Energies Renouvelables*,2007,Vol.10, N°3, 387-395.
- [53] F.Bouchafa, D.Beriber,M.Boucherit, "Modeling and Simulation of a Grid Connected PV Generation system with MPPT Fuzzy Logic Control",In: *Proceeding of the Systems Signals and Devices (SSD)*, 2010 7th International Multi-Conference on.2010.pp1-7. <http://dx.doi.org/10.1.1.109/SSD.2010.5585530>.
- [54] F.Chekired, C.Larbes, D.Rekioua, F.Haddad, "Implementation of a MPPT Fuzzy Controller for PV Systems on FPGA Circuit", *Energy Procedia* 2011, 6(0): 541-549. <http://dx.doi.org/10.1016/j.egypro.2011.05.062>.
- [55] M.A.Othmana, M.M.ElArinia, A.Ghitasb, "Realworld Maximum Power Point Tracking Simulation of PV System based on FLC", *NRIAG Astron Geophys* 2012:1(2):186-94.
- [57] H.Bounechba, A.Bouزيد, K.Nabti, H.Benalla, "Comparison of P&O and Fuzzy Logic in MPPT for PV Systems". *Energy Procedia* 2014:50:677-84.
- [58] C.Larbes, S.M.Ait Cheikh et al, "Genetic Algorithms optimized Fuzzy Logic Control for the Maximum Power Point Tracking in Photovoltaic system", *Renewable Energy*,2009,Vol.34,2093-2100.
- [59] F. Chekired, A. Mellit, C. Larbes, S.A. Kalogirou, "Intelligent Maximum Power Point Tracking for PV Applications using FPGA chip: A Comprehensive Study", *Solar energy*, 101, 2014, 83-99.
- [60] A. Messai, A. Mellit, et al, "Maximum Power Point Tracking using Genetic Algorithm Optimized fuzzy Logic controller and its FPGA Implementation", *Solar Energy*, 2011,Vol.85, 265-277.
- [61] A.A.Kulaksiz, R.Akkaya, "A Genetic Algorithm Optimized ANN Based MPPT Algorithm for a Stand Alone PV System with Induction Motor Drive", *Solar Energy*, 2012, Vol.86, Issu.9, 2366-2375.
- [62] R.Ramaprabha, V.Gothandaraman, et al, "Maximum Power Point Tracking Using GA-Optimized ANN for Solar PV System", *International Conference on Electrical Energy Systems, ICEES-2011*, 264-268.
- [63] Syafaruddin, E.Karatep, "Artificial Neural Network Polar Coordinated Fuzzy controller Based Maximum Power Point Tracking Under Partially Shaded Conditions", *IET Renewable Power Generation*, Vol.3, Issue:2, 2009, 239-253.
- [64] A.M.Z Alabedin, EF.El-Saadany, et al, "Maximum Power Point Tracking for Photovoltaic System using Fuzzy Logic and Artificial Neural Networks", *IEEE Power and Energy Society General Meeting*,2011,1-9.
- [65] F.Chekired, C.Larbes,et al, "FPGA Based Real time Simulation of ANFIS MPPT Controller for Photovoltaic Systems", *International Review on Modelling and Simulations, IREMOS*, 2011, V.4, N°5, 2361-2366.

- [66] F.Chekired, C.Larbes, A.Mellit, "Comparative Study between two Intelligent MPPT Controller implemented on FPGA: Application for Photovoltaic Systems", *International Journal for Sustainable Energy*, doi:10.1080/14786451.20120742896.
- [67] F.Belhachet, "Commande Neuro-Floue d'un Hacheur MPPT", Mémoire de Magister, Ecole Nationale Polytechnique (ENP), Département Electronique, Laboratoire des Dispositifs de Communications et de Conversion Photovoltaïque (LDCCP), 2007.
- [68] A.H.Hanif, I.Ismail, "Bio Inspired Optimization method: A review", *International journal of artificial Intelligence*, 2014, Vol.1.
- [69] S.Binitha, S.Siva, et al, "A survey of Bio Inspired optimization Algorithm", *International journal of soft computing and Engineering*, 2012, Vol.2, 137-150.
- [70] H. Duan, X.Zhang, C. Xu, "Bio-Inspired Computing", Beijing, China, Science Press, 2011.
- [71] J.Holland, "Adaptation in natural and artificial systems", University of Michigan Press, 1975.
- [72] S.Hadji, J.P.Gaubert, et al, "Genetic Algorithms for MPPT in PV systems", *Power Electronics and Applications*, 2011, 1-9.
- [73] R.Ramaprabha, B.L. Mathur, "Genetic algorithm based maximum power point tracking for partially shaded solar photovoltaic array", *International Journal of Research and Reviews in Information Sciences (IJRRIS)*, Vol. 2, N°.1, March 2012, ISSN: 2046-6439
- [74] H.R.Mohajeri, M.P.Moghaddam, et al, "Development of a new algorithm for maximum power point tracking of partially shaded photovoltaic arrays", *Proceeding of the 20th Iranian conference on electrical engineering (ICEE), IEEE 2012*, pp 489-494.
- [75] R.Storn, K.Price, "Differential Evolution: A Simple and Efficient Heuristic for global optimization over Continuous Spaces", *Journal of Global Optimization*, Vol.11, pp. 341-359, 1997
- [76] K.S.Tey, S.Mekhilef, et al, "A differential evolution based MPPT method for photovoltaic modules under partial shading conditions". *International Journal of Photo Energy*, 2014.
- [77] S. Ramasamy, S. Jeevananthan, et al, "An Intelligent Differential Evolution Based Maximum Power Point Tracking (MPPT) Technique for Partially Shaded PhotoVoltaic (PV) Array", *International Journal of Advances in Soft Computing & its Applications*, Vol.6, Issue 2, 2014, 1-16.
- [78] S.Taheri, Z.Salam, et al, "A novel Maximum Power Point Tracking control of Photovoltaic System under Partial and Rapidly Fluctuating Shadow Conditions Using Differential Evolution", *IEEE Symposium on Industrial electronics & Application, (ISIEA), 2010*, 82-87.
- [79] M.F.Tajuddin, M.A. Shahrin, et al, "Evolutionary based maximum power point tracking technique using differential evolution algorithm", *Energy and Buildings*, Vol.67, 2013, pp 245-252.
- [80] I.Kashif, S.Zainal, "An improved Particle Swarm Optimization (PSO) based MPPT for PV with reduced steady-state oscillation". *IEEE Trans Power Electron* 2013, 27(8):3627-3638.
- [81] I.Kashif, S. Zainal, "A deterministic particle swarm optimization maximum power point tracker for photovoltaic system under partial shading condition", *IEEE Trans Ind Electron*, 2013, 60, 3195-3206.
- [82] S.R.Chowdhury, H.Saha, "Maximum power point tracking of partially shaded solar photovoltaic arrays", *Solar Energy Mater Solar Cells*, 2010, Vol.94, 14, 41-7.
- [83] V.Phimmasone, T. Endo, Y. Kondo, et al, "Improvement of the maximum power point tracker for photovoltaic generators with particle swarm optimization technique by adding repulsive force among agents", *International Conf on Electrical Machines and Systems, ICEMS 2009*, doi: 10.1109/ICEMS.2009.5382713.
- [84] M.Miyatake, M.Veerachary, et al, "Maximum power point tracking of multiple photovoltaic arrays: a PSO approach", *IEEE Transaction on Aerospace Electronic Systems*, Vol.47, Issue.1, 2011, pp.367-380.
- [85] M.Seyedmahmoudian, S.Mekhilef, et al, "A Maximum power point tracking of partial shaded photovoltaic array using an evolutionary algorithm: A particle swarm optimization technique", *Journal of Renewable and Sustainable Energy*, Vol.6, Issue.2, 023102 (2014); doi: <http://dx.doi.org/10.1063/1.4868025>
- [86] Abou soufyane Benyoucef, et al, "Artificial bee colony based algorithm for maximum power point tracking (MPPT) for PV systems operating under partial shaded conditions", *Applied Soft Computing*, Volume 32, July 2015, Pages 38-48, <http://dx.doi.org/10.1016/j.asoc.2015.03.047>.

- [87] B.Bilal, "Implementation of artificial bee colony algorithm on MPPT for PV modul", 8th International Symposium on Advanced Topics in Electrical Engineering, ATEE Bucharest, Romania, 23-25 May, 2013.
- [88] A. S.Oshaba, E. S. Ali and S. M. Abd Elazim," Artificial Bee Colony Algorithm Based Maximum Power Point Tracking in Photovoltaic System", WSEAS TRANSACTIONS on POWER SYSTEMS, Volume 10, 2015, 123-134. E-ISSN: 2224-350X
- [89] H.Salmi, A.Badri ,M.Zegrari, Maximum Power Point Tracking (MPPT) Using Artificial Bee Colony Based Algorithm for Photovoltaic System, International Journal of Intelligent Information Systems, Volume 5, Issue 1, February 2016, 1-4.
- [90] Kinattungal Sundareswaran, Peddapati Sankar, et al, "Enhanced Energy Output From a PV System Under Partial Shaded Conditions Through Artificial Bee Colony", IEEE Transaction on sustainable energy, 10.1109/TSTE.2014.2363521.
- [91] B. Babar, A. Craciunescu, "Comparison of Artificial Bee Colony Algorithm with other Algorithms used for Tracking of Maximum Power Point of Photovoltaic Arrays", International Conference on Renewable Energies and Power Quality (ICREPQ'14), Cordoba ,8th to 10th April, 2014, ISSN 2172-038 X, No.12, April 2014.
- [92] X.S.Yang,"Nature inspired metaheuristic algorithm", Luniver press, 2010.
- [93] X.S.Yang,"Firefly algorithms for multimodal optimization Stochastic algorithms foundation and applications", Springer, 2009, pp.169-178.
- [94] Sundareswaran Kinattungal,et al, "MPPT of PV Systems Under Partial Shaded Conditions Through a Colony of Flashing Fireflies", IEEE TRANSACTIONS ON ENERGY CONVERSION, VOL. 29, NO. 2, JUNE 2014, <http://dx.doi.org/10.1109/TE.2014.2298237>.
- [95] B.B.Prasad, M A. Chitra, "Implementation of MPPT Algorithm Using Firefly Technique for Solar Photovoltaic Systems", International Journal for Scientific Research & Development| Vol. 3, Issue 02, 2015.
- [96] D.F.Teshome, C. H. Lee,et al, "A Modified Firefly Algorithm for Photovoltaic Maximum Power Point Tracking Control Under Partial Shading", IEEE Journal of Emerging and Selected Topics in Power Electronics, Vol.5, Issue.2, June 2017.
- [97] S.Mirjalili, S.M.Mirjalili, et al, "Grey Wolf Optimizer", Advances in Engineering Software, Volume 69, March 2014, pp. 46-61, <https://doi.org/10.1016/j.advengsoft.2013.12.007>.
- [98] Satyajit Mohanty,et al, "A New MPPT Design Using Grey Wolf Optimization Technique for Photovoltaic System Under Partial Shading conditions", IEEE TRANSACTIONS ON SUSTAINABLE ENERGY, DOI: 10.1109/TSTE.2015.2482120
- [99] Satyajit Mohanty et al, "A grey wolf optimization based MPPT for PV system under changing insolation level", Technology Symposium (TechSym), 2016 IEEE Students', doi: 10.1109/TechSym.2016.7872677.
- [100] Santhan kumar Cherukuri, Srinivasa Rao Rayapudi, "Enhanced Grey Wolf Optimizer based MPPT Algorithm of PV system under Partial Shaded Condition", International Journal of Renewable Energy Development (IJRED), doi: <https://doi.org/10.14710/ijred>.
- [101] X.S.Yang, M.Karamanoglu, "Multi Objective Flower Algorithm for Optimization, (cs.NE).2 Apr 2014.
- [102] J. Prasanth Ram,et al, "A new global maximum power point tracking technique for solar photovoltaic (PV) system under partial shading conditions (PSC)", Energy, <http://dx.doi.org/10.1016/j.energy.2016.10.084>.
- [103] J. Prasanth Ram and N. Rajasekar, "A Novel Flower Pollination Based Global Maximum Power Point Method for Solar Maximum Power Point Tracking", IEEE Transactions on Power Electronics, Vol. 32, Issue. 11, Nov. 2017, doi: 10.1109/TPEL.2016.2645449.
- [104] A.Z.Diab, H.Rezk, "Global MPPT based on flower pollination and differential evolution algorithms to mitigate partial shading in building integrated PV system", Solar Energy, Vol.157, Nov2017, pp 171-186, <https://doi.org/10.1016/j.solener.2017.08.024>.
- [105] K.N.Krishnanand, D.Ghose,"GlowwormSwarm Optimization for simultaneous capture of multiple local optima of multimodal functions", Swarm Intell, 2009,3,87-124.
- [106] Y.Jin, W. Hou, "A Glowworm Swarm Optimization-Based Maximum Power Point Tracking for photovoltaic/Thermal Systems under Non-Uniform Solar Irradiation and Temperature Distribution", Energies, 2017, 10, 541 ; doi:10.3390/en10040541.

- [107] W.Hou, Y.Jin, "A Novel Maximum Power Point Tracking Algorithm Based on Glowworm Swarm Optimization for Photovoltaic Systems", *International Journal of Photoenergy*, Vol. 2016, Article id. 4910862, <http://dx.doi.org/10.1155/2016/4910862>
- [108] Rao, R.V.,Patel, V. Chen, J.P. "An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems". *Int. J. Ind. Eng. Comput.* 2012, 3, 535–560.
- [109] R.V.Rao, V. Savsani, et al, "Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems", *Journal. Comput. Aided Des*, 2011, 43, 303–315.
- [110] R.V. Rao, V. Patel, "An improved teaching-learning-based optimization Algorithm for solving unconstrained optimization problems", *Comput. Sci. Eng. Electr. Eng.* 2013, 20, 710–720.
- [111] Ahmed fathy, et al, "An Improved Teaching Learning Based Optimization Algorithm for Simulating the Maximum Power Point Tracking Controller in Photovoltaic system, The Egyptian International Journal of Engineering Sciences and Technology, Vol.21,Oct2016, pp9-18.
- [112] Kuei-Hsiang Chao, et al, « Global Maximum Power Point Tracking (MPPT) of a Photovoltaic Module Array Constructed through Improved Teaching-Learning-Based Optimization", *Energies* 2016, 9, 986; doi:10.3390/en9120986.
- [113] X.S.Yang, S.Deb, "Engineering Optimisation by Cuckoo Search", *Int. J. Mathematical Modelling and Numerical Optimisation*, Vol. 1, No.4, 330-343 (2010).
- [114] J.Ahmed, Zainal Salam , "A Maximum Power Point Tracking (MPPT) for PV system using Cuckoo Search with partial shading capability, *Applied Energy*, Volume 119, 15 April 2014, Pages 118-130
- [115] J.Ahmed, Zainal Salam" A Soft Computing MPPT for PV systems Based on Cuckoo Search Algorithm", 4th International Conference on Power Engineering, Energy and Electrical Drives, (POWERENG), 13-17 May 2013, 558-562. <http://dx.doi.org/10.1016/j.apenergy.2013.12.062>
- [116] M.Eusuff, K.E.Lansey,"Optimization of water distribution network design using the shuffled frog leaping algorithm" ,*Water Resources Planning and Management* 129(3), 210–225,2003.
- [117] R. Sridhar, et al, "New maximum power tracking in PV system during partially shaded conditions based on shuffled frog leap algorithm", *Journal of Experimental & Theoretical Artificial Intelligence*, doi:10.1080/0952813X.2016.1186750.
- [118] M.Dorigo, Thomas.Stutzle, "Ant Colony Optimization", MIT Press, Cambridge, MA, 2004.
- [119] J.Dréo, P.Siarry, et al, "Metaheuristiques pour l'optimisation difficile", Eyrolles, 2003.
- [120] B.bilchev, C.Parmee, "The Ant Colony Metaphore for searching Continious Design Spaces", *Proceeding of the AISB Workshop, Evolutionary Computing, Lecture Note in Computer Science*, Vol. 993,1993,25-39.
- [121] M. Mathur, S.B Karale, et al, "Ant Colony Approach to Continious Function Optimization", *Industrial & Engineering Chemistry Research*, 39 (10), 2000, 3814-3822.
- [122] O.Kovarik, "Ant Colony Optimization for Continious Problems, Master Thesis", Czech Technical University in Prague, The Faculty of Electrical Engineering (2006).
- [123] K.Socha, "Ant Colony Optimization for continuous and mixed variable domain", Phd thesis, IRIDIA, Université Libre de Bruxelles, CP 194/6, (2008).
- [124] K.Socha, M.Dorigo, "Ant Colony Optimization for Continuous Domains", *European Journal of Operational Research*, Vol.185, 2008, 1155-1173. doi:10.1016/j.ejor.2006.06.046.
- [125] M. Adly, A.H. Beshar, "An Optimized Fuzzy Maximum Power Point Tracker for Stand Alone Photovoltaic Systems : Ant Colony Approach", 7th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2012, 113-119.
- [126] M.Adly, A.H.Beshar, "A Meta Heuristics Search Algorithm as a Solution for Energy Transfer Maximization in Stand Alone Photovoltaic Systems", *International Journal of Electrical Power and Energy Systems*, Vol.51, 2013, 243-254.
- [127] L.Jiang, L.Douglas et al, "A Novel Ant Colony Optimization Based MPPT for Photovoltaic Systems Under Partially Shaded Conditions", *Energy and Buildings*, Vol.58, 2013, 227-236.
- [129] X.S.Yang, "Bat Algorithm: Literature review and applications", *Bio inspired Computations*, Vol.5,N°.3,pp141-149,2013.
- [130] A.S.Oshaba, E.S.Ali, et all, "PI controller design for MPPT of photovoltaic system supplying SRM via BAT search algorithm", *Neural Computing and Applications*, April 2017, Volume 28, Issue 4, pp 651–667.

- [131] A.S.Oshaba, E.S.Ali, et al, "MPPT Control Design of PV System Supplied SRM using BAT Search Algorithm", Sustainable Energy Grids and Networks, Vol 2, 2015, 51-60.
- [132] M.K.Karagoz, H.Demirel, "A Novel MPPT Method for PV Arrays Based on Modified Bat Algorithm with Partial Shading Capability", International Journal of Computer Science and Network Security, Vol.17, N°.2, 2017
- [133] C.Hamzaçebi,F.Kutay, "A heuristic approach for finding the global minimum: adaptative random search technique", Appl.Math.Comput.173(2),2006,1323-1333.
- [134] U.S Serap, E.Yunus et al, "Ant Colony Optimization for Continuous Functions by Using Novel Pheromone Updating", Applied Mathematics and Computations, Vol.219, Issue 9, 2013, 4163-4175, <http://dx.doi.org/10.1016/j.amc.2012.10097>
- [135] Matlab, www.mathwork.com.
- [136] <http://photovoltaic-software.com/>
- [137] <https://www.capterra.com/p/87533/PV-DesignPro/>
- [138] <http://www.pvsyst.com>
- [139] <http://www.trace-software.com/archelios/>
- [140] <https://www.homerenergy.com/>
- [141] energyPRO
- [142] Qt
- [142] Qwt
- [143] <http://www.ni.com/white-paper/8043>
- [144] www.bdti.com
- [145] www.xilinx.com
- [146] www.altera.com
- [147] www.xilinx.com/wp373
- [148] www.xilinx.com/wp380
- [149] www.xilinx.com/wp
- [150] www.xilinx.com/wp405
- [152] www.xilinx.com/Ug848
- [153] A.Zbeeb, "A new microcontroller based MPPT algorithm for photovoltaic applications", Master thesis, department of electrical and computer engineering, Concordia University, Montréal, Quebec, Canada, 2009.
- [154] N.Khaehintung, K.Pramotung, et al, "RISC microcontroller built-in fuzzy logic controller of maximum power point tracking for solar-powered light-flasher applications". In: ICON The 30th Annual Conference of the IEEE Industrial Electronics Society, 2004, pp. 2673–2678
- [155] R.Akkaya, A.Kulaksız, et al, "DSP implementation of a PV system with GA-MLP NN based MPPT controller supplying BLDC motor drive", Energy Conversion and Management 48 (1), 2007, pp. 210–218.
- [156] T.E.Persen, "FPGA Based Design of a MPPT system for space applications", Master of Science, B.S.University of Central Florida, Orlando, 2004.
- [157] J. Detrey, "Arithmétiques réelles sur FPGA-Virgule fixe, flottante et système logarithmique", Thèse de Doctorat de l'Ecole Normale Supérieure de Lyon, 2007.
- [158] R.C. Cofer, B.F. Harding, "Rapid system prototyping with FPGAs", USA: Elsevier Inc; 2006.
- [159] N.Khachintung, T.Wiangtong, et al, "FPGA implementation of MPPT using variable step size P&O algorithm for PV applications", IEEE International Symposium on Communication and Information, 2006, 212-215, <http://doi:10.1109/ISCIT.2006.340033>
- [160] Y. Elbasri, P.Stephane, et al, "New P&O MPPT algorithm for FPGA implementation", IECON, 36th annual conference on IEEE industrial electronics society, 2010. pp. 2868-2873.
- [161] A.Mellit, S.Kalogirou, "MPPT based artificial intelligence techniques for photovoltaic systems and its implementation into field programmable gate array chips : Review of current status and future perspectives", Energy, 2014.
- [162] Z.Cheng, H.Yang, et al, "FPGA-based PV systems fuzzy MPPT control algorithm", 7th International conference on Fuzzy Systems and Knowledge Discovery, 2010, pp. 1244-1248.
- [163] P.Dzung, L.Khoa, et al, "The new MPPT algorithm using ANN-based PV", International Forum on Strategic Technology, 2010. pp. 402-407.

- [164] A.Messai, A.Mellit, et al, "FPGA implementation of an intelligent MPPT for photovoltaic panel", International symposium on hydrocarbons & chemistry, University M'hamed Bouguara, Boumerdes, 2010.
- [165] A.Messai, et al, "FPGA based implementation of a fuzzy controller MPPT for PV module", Energy Conversion & Management, Vol.52, Issue.7, 2011, pp.2695-2704.
- [166] F.Chekired, C.Larbes, et al. "Implementation of a MPPT fuzzy controller for photovoltaic systems on FPGA", Energy procedia, 2011, 6 pp.541-549.
- [167] Z.Cheng, H.Yang, et al, "Self-adjusting fuzzy MPPT PV system control by FPGA", IEEE Asia-Pacific Power and Energy Engineering Conference, 2011, pp. 1-4.
- [168] F.Chekired, C.Larbes, et al, "FPGA based real time simulation of ANFIS MPPT controller for photovoltaic systems", International Review on Modeling and simulations, Vol.4, Issu.5, pp.2361-2366.
- [169] F.Chekired, C.Larbes, et al, "Comparative study between two intelligent MPPT-controllers implemented on FPGA : application for photovoltaic systems", International Journal of Sustainable Energy, DOI:10.1080/14786451, 2012, pp.742-896.
- [170] A.Chetibi, A.Mellit, "FPGA based implementation of InCnd algorithm for photovoltaic application", 24th International Conference Microelectronics, 2010, DOI.10.1109/ICM.2012.6471401.
- [171] M.Basil, S.ElMoghany, "Fuzzy controller design using FPGA for photovoltaic MPPT", International journal of advanced research in artificial intelligence, Vol.1, N°3, 2012, pp14-21.
- [172] K. Punitha, D. Devaraj, et al, "Development and analysis of adaptive fuzzy controllers for photovoltaic system under varying atmospheric and partial shading condition", Applied Soft computing, 2013, <http://dx.doi.org/10.1016/j.asoc.2013.06.021>
- [173] K. Punitha, D. Devaraj, et al, "Artificial neural network based modified incremental conductance algorithm for maximum power point tracking in photovoltaic system under partial shading conditions", Energy, Vol.62, 2013, pp.330-340. <http://dx.doi.org/10.1016/j.energy.2013.08.022>
- [174] F.Chekired, A.Mellit, C.Larbes et al, "Intelligent maximum power point trackers for photovoltaic applications using FPGA chip : a comparative study", Sol Energy, 2014, Vol.10, pp 83-99.
- [175] S.Titri, C.Larbes, K.Youcef-Toumi, "Rapid prototyping of PVS into FPGA : From Model Based Design to FPGA/ASICs implementation", 9th International Design and Test symposium, IDT'2014, pp.162-167.
- [176] S.Titri, C.Larbes, K.Youcef-Toumi, "A New design methodology for rapid prototyping of MPPT controllers into FPGA", 6th International Renewable energy Congress, IREC'2015.
- [177] H.Abbes, K.Loukil, "Implementation of a photovoltaic Maximum Power Point Tracking fuzzy controller on FPGA ", Journal of information assurance and security, ISSN 1554-1010, Vol.11, 2016, pp.97-106.
- [178] K.Kaced, C.Larbes, et al, "FPGA implementation of PSO based MPPT for photovoltaic system under partial shading condition", 6th International conference on systems and control (ICSC), 2017, DOI.10.1109/ICoSC.2017.7958682.
- [179] A.Youssef, M, El Telbany, et al, "Reconfigurable generic FPGA implementation of fuzzy logic controller for MPPT of PV systems", Renewable and Sustainable Energy Reviews, Vol.82, pp1313-1319, <http://dx.doi.org/10.1016/j.rser.2017.09.093>.
- [180] D.Gajski, R.H. Kuhn, "Guest Editors" Introduction : New VLSI Tools", IEEE Computer, Vol.16, N°12, 1983, pp. 11-14.
- [181] H.Kaeslin, "Top-down Digital VLSI Design: From Architectures to Gate-level Circuits and FPGAs", Morgan Kaufmann, 2014.
- [182] Standard VHDL Language Reference manual, New York, The institute of electrical and Electronics engineers, Inc., 1988.
- [183] D.E.Thomas, P.Moorby, "The Verilog Hardware Description Language", Kluwer Academic Publishers, 1991.
- [184] D.Gajski, N.Dutt, et al, "High Level Synthesis: Introduction to Chip and System Design", Kluwer Academic Publishers, 1992, ISBN0-7923-9194-2

- [185] Wakabayashi, Kazutoshi, “C-based behavioral synthesis and verification analysis on industrial design examples”, Proceedings of the 2004 Asia and South Pacific Design Automation Conference. IEEE Press, 2004.
- [186] Cong, Jason, et al., “High-level synthesis for FPGAs: From prototyping to deployment”, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 30.4 (2011): 473-491.
- [188] ARTISAN Software Tools, “Artisan real-time studio support for Model-Driven Architecture (MDA)”, ARTiSAN Software Tools, Whitepaper, 2002, http://www.omg.org/mda/mda_files/ArtisanRealtimestudio.pdf
- [189] N.Razvan,V.Sima, “A Survey and Evaluation of FPGA High Level Synthesis tools”, IEEE Transactions on Computer Aided Design of Integrated Circuits and systems, Vol.35,N°.10,Octobre 2016.
- [190] W.Meeus,K.Beeck, et al, “An overview today’s of high level synthesis tools”, Des Autom Embed Syst, DOI.10.1007/s10617-012-9096-8.
- [191] www.mathworks.com/hdlCoder
- [192] <http://www.xilinx.com/products/design-tools/vivado/integration/sysgen.html>
- [193] www.ni.com/LabView
- [194] www.altera.com/DSPBuilder
- [195] <http://www.calypto.com>
- [196] <http://www.forteds.com>
- [197] <http://www.synopsys.com>
- [198] Stream-C
- [199] <http://www.xilinx.com/products/design-tools/vivado/integration/esl-design/index.htm>
- [200] <http://www.xilinx.com/products/design-tools/vivado>
- [201] <http://www.xilinx.com/products/design-tools/vivado/HLS>