

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Nationale Polytechnique
Département Electronique



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique



End of studies project
To obtain the diploma of State Engineer in Electronics

Theme:

**Contribution to the design and implementation of
the Bio-inspired Firefly algorithm on FPGA circuit**

By:

**BENKHIRA Fayçal
DJEMAI Narimane**

Examiners:

- | | | |
|-----------------------------|------------------|------------|
| • Mme Nour El Houda BENALIA | MCB, ENP | President |
| • Mme Sabrina TITRI | MRB, CDTA | Supervisor |
| • Mme Hadjer AZLI | PhD Student, ENP | Supervisor |
| • Mr. Cherif LARBES | Professor, ENP | Supervisor |
| • Mr. Mohamed Oussaid TAGHI | MAT, ENP | Examiner |

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Nationale Polytechnique
Département Electronique



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique



End of studies project
To obtain the diploma of State Engineer in Electronics

Theme:

**Contribution to the design and implementation of
the Bio-inspired Firefly algorithm on FPGA circuit**

By:

**BENKHIRA Fayçal
DJEMAI Narimane**

Examiners:

- | | | |
|-----------------------------|------------------|------------|
| • Mme Nour El Houda BENALIA | MCB, ENP | President |
| • Mme Sabrina TITRI | MRB, CDTA | Supervisor |
| • Mme Hadjer AZLI | PhD Student, ENP | Supervisor |
| • Mr. Cherif LARBES | Professor, ENP | Supervisor |
| • Mr. Mohamed Oussaid TAGHI | MAT, ENP | Examiner |

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Nationale Polytechnique
Département Electronique



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Projet de fin d'Etudes
Pour l'obtention du diplôme d'Ingénieur d'Etat en Electronique

Thème:

**Contribution à la conception et l'implémentation de
l'algorithme bio-inspiré Firefly sur circuit FPGA**

Réalisé Par:

**BENKHIRA Fayçal
DJEMAI Narimane**

Composition du Jury :

- | | | |
|-----------------------------|-----------------|------------|
| • Mme Nour El Houda BENALIA | MCB, ENP | Président |
| • Mme Sabrina TITRI | MRB, CDTA | Rapporteur |
| • Mme Hadjer AZLI | Doctorante, ENP | Rapporteur |
| • M. Cherif LARBES | Professeur, ENP | Rapporteur |
| • M. Mohamed Oussaid TAGHI | MAT, ENP | Examineur |

ملخص: لتعزيز الطاقة الموردة من النظام الكهروضوئي ، فإن تتبع نقطة الحد الأقصى للطاقة أو MPPT هو الطريقة المناسبة للقيام بذلك. هناك العديد من العوامل التي يمكن أن تؤثر على الطاقة المنتجة من الأنظمة الكهروضوئية ، مثل درجة الحرارة و شدة الإشعاع الشمسي والتظليل الجزئي. يتم تغيير خاصية المصفوفة الكهروضوئية عند حدوث تظليل كلي أو جزئي عليها بسبب السحب والأشجار العالية وترسب الغبار والمباني المحيطة أو الظل الناتج عن إحدى المصفوفات الكهروضوئية للأخرى المحيطة. سيؤدي ذلك إلى ظهور العديد من القيم الحدية القصوى في منحى الطاقة. مما يؤدي الى ضياعها. عند حدوث التظليل الجزئي ، طرق تتبع نقطة القيمة الحدية القصوى للطاقة التقليدية غير قادرة على اكتشاف هاته الأخيرة ، وبالتالي ، في هذا المستند ، ستكون خوارزمية البراعة هي الطريقة الأساسية لوحدة التحكم MPPT ، من أجل تحسين خوارزمية تتبع نقطة الحد الأقصى للطاقة. ستكون هنالك العديد من الاختبارات والتحليلات من خلال تغيير درجة الحرارة و قوة الإشعاع ، إلى جانب المقارنة مع خوارزمية استمثال عناصر السرب أو PSO ، التي تنتهي بتحميل الخوارزمية في بطاقة FPGA ، هذا سيبين مدى تفوق الخوارزمية المقترحة المستوحاة من الحياة الحيوية في الطبيعة وكفاءتها.

الكلمات المفتاحية: النظام الكهروضوئي ، تتبع نقطة الحد الأقصى للطاقة (MPPT) ، الخوارزميات المستوحاة من الحياة الحيوية ، خوارزمية البراعة ، التصميم المستند إلى النموذج (MBD) ، FPGA.

Résumé : Pour améliorer la puissance fournie par un système photovoltaïque, le suivi du point de puissance maximale ou MPPT est le moyen d'y parvenir. Plusieurs facteurs peuvent influencer la puissance produite par un système photovoltaïque, tels que la température, l'irradiation et l'ombrage partiel. Les caractéristiques d'un réseau photovoltaïque sont modifiées lorsque la totalité ou certaines parties de ce système sont ombragées par des nuages, de grands arbres, des dépôts de poussière, des bâtiments environnants ou l'ombre créée par l'un des réseaux photovoltaïques sur d'autres dans des environnements extérieurs. Cela entraînera l'apparition de plusieurs maximums dans la courbe photovoltaïque. Ce qui provoque des pertes superflues. Lorsque l'ombrage partiel se produit, les méthodes MPPT conventionnelles seront incapables de détecter le point maximum global de la puissance, par conséquent, dans ce document, l'algorithme de Luciole sera la méthode basée pour le contrôleur MPPT, afin d'améliorer le suivi du point de puissance maximale. Plusieurs tests et analyses en faisant varier la température et l'irradiation, ainsi que la comparaison avec l'algorithme Optimisation par essaims de particules (PSO), et enfin l'implémentation de l'algorithme en FPGA, démontreront la suprématie de l'algorithme bio-inspiré proposé et son efficacité.

Mots-clés : Système photovoltaïque, suivi du point de puissance maximale (MPPT), algorithmes bio-inspirés, algorithme Luciole, conception basée sur un modèle (MBD), FPGA.

Abstract: To enhance the supplied power from a photovoltaic system, the maximum power point tracking or MPPT is the way to do so. there are several factors that can influence the produced power from a Photovoltaic system, such as the temperature, irradiation and the partial shading. Photovoltaic array characteristic gets altered when complete or some parts of this system are shadowed by clouds, tall trees, deposition of dust, and surrounding buildings or shadow created by one of the Photovoltaic arrays to others in outside environments. This will lead to the appearance of several maximums in the Photovoltaic curve. That cause superfluous losses. When the partial shading occurs, the conventional MPPT methods will be incapable to detect the global maximum point of the power, therefore, in this document, the Firefly Algorithm will be the based method for the MPPT controller, in order to improve the maximum power point tracking. Several tests and analysis by varying the temperature and the irradiation, beside the comparison with Particle Swarm Optimization (PSO) algorithm, ending with algorithm implementation in FPGA, will demonstrate the supremacy of the proposed Bio- Inspired Algorithm and its efficiency.

Keywords: Photovoltaic system, Maximum Power Point Tracking (MPPT), bio-inspired algorithms, Firefly Algorithm, Model Based Design (MBD), FPGA.

Acknowledgements

We would like to thank Mr. LARBES Cherif, Professor at the National Polytechnic School of Algiers, and we particularly express our gratitude to him for having made us benefit from his scientific skills, his human qualities and his constant availability.

We would like to thank Mrs. Sabrina TITRI, Doctor and Researcher at the Research Centre for Advanced Technologies, who offered us the possibility to realize this thesis under her direction.

We would also like to thank Miss AZLI Hadjer, doctoral student at the National Polytechnic School of Algiers for her advice and her help during the work period.

We would also like to thank the members of the jury for their interest in this project by agreeing to examine it and to enrich it with their proposals.

Our thanks go to all the teachers of the Electronics Department of the National Polytechnic School.

Dedication

Praise to ALLAH Almighty who gave me the strength and patience to complete this work

I dedicate this modest work to those who gave me everything without asking me anything and to whom I owe a lot and to whom I will never thank never enough:

My dear parents.

My friends during my entire academic career.

To everyone who gave me support.

Fayçal BENKHIRA

Dedication

Every challenging work needs self-efforts as
well as Guidance of elders
especially those who were very close to your
heart.

My humble effort I dedicate to my sweet and
loving

Mother & Father

Two brothers

Whose affection, love, encouragement and
prays

make me able to get such Success and honour.

Along with all my friends, all hard working
and respected Teachers

Narimane DJEMAI

Table of Contents

Table of figures

Tables

Abbreviation List

- General Introduction..... 15**
- 1. Photovoltaic Electrical Energy..... 18**
 - 1.1 Introduction..... 18
 - 1.2 Photovoltaic Systems..... 18
 - 1.3 The Photovoltaic Cell reviews..... 19
 - 1.3.1 Definition 19
 - 1.3.2 Definition 19
 - 1.3.3 Characteristic Curves 20
 - 1.4 Photovoltaic Cell Shading 20
 - 1.5 Effects of Climate Variations on the photovoltaic Module 22
 - 1.5.1 Influence of the illumination..... 22
 - 1.5.2 Influence of temperature..... 23
 - 1.6 MPPT 23
 - 1.7 Conclusion..... 24
- 2. Maximum power point tracking methods. State of the art 26**
 - 2.1 Introduction..... 26
 - 2.2 Performance criteria of MPPT algorithm..... 26
 - 2.3 Classification of MPPT methods 28
 - 2.3.1 Conventional methods..... 28
 - 2.3.2 Advanced methods..... 28
 - 2.3.2.1 AI-Based Methods 29
 - 2.3.2.2 Bio-Inspired (BI) methods 30
 - 2.4 The Firefly algorithm..... 33
 - 2.4.1 Basic concepts..... 34
 - 2.4.2 Steps of the Firefly Algorithm [31] 35
 - 2.4.2.1 Initialization..... 35
 - 2.4.2.2 Exploration 35
 - 2.4.2.3 Checking the convergence criteria..... 38
 - 2.4.3 Algorithm parameters 40
 - 2.5 Application of the Firefly Algorithm (FA) for MPPT controller..... 40

2.6 Application of the PSO algorithm for MPPT controller	44
2.7 Conclusion	45
3. Software Implementation of FA-MPPT Controller	47
3.1 Introduction.....	47
3.2 Simulation Methodology of the proposed MPPT controller	47
3.2.1 Modelling of the photovoltaic system	48
3.2.1.1 PV array.....	48
3.2.1.2 MPPT controller block.....	49
3.2.1.3 Buck-Boost converter	49
3.2.2 Parameterization of FA and PSO	49
3.2.2.1 FA parameterization	50
3.2.2.2 Parameterization of the PSO algorithm.....	52
3.2.3 Simulation of the FA controller	52
3.2.3.1 Study of the convergence criteria.....	53
3.2.3.2 Study of stability, accuracy and robustness criteria	54
a. Rapid variation of the irradiation for a fixed temperature	54
b. Rapid temperature change at fixed irradiation.....	55
3.2.3.3 Effect of partial shading	56
3.3 Conclusion	63
4. Hardware Implementation of FA-MPPT Controller	65
4.1 Introduction.....	65
4.2 Hardware Design Methodologies	66
4.2.1 FPGA.....	66
4.2.2 FPGA Architecture.....	66
4.2.3 Evolution of co-design methodologies.....	67
4.2.3.1 Bottom-Up methodology	67
4.2.3.2 Top-Down methodology	67
4.3 Model based design conception	71
4.3.1 HDL Coder.....	72
4.3.1.1 HDL Workflow Advisor.....	73
4.3.1.2 Fixed-Point Conversion	73
4.3.1.3 HDL Verification.....	74
4.3.1.4 HDL Synthesis.....	74
4.3.2 Vivado Xilinx.....	75
4.4 Application of the MBD approach for the FA-MPPT controller.....	76
4.4.1 Introduction.....	76

4.4.1 Workflow Advisor in MATLAB.....	76
4.4.1.1 Launching HDL Coder	77
4.4.1.2 Targeting .m Function and Test Bench	77
4.4.1.3 Defining input types	78
4.4.1.4 Converting to fixed point and targeting the device.....	78
4.4.1.5 Generate HDL Code.....	79
4.4.1.6 Synthesis HDL Code	79
4.4.2 Workflow Advisor in Simulink.....	79
4.4.2.1 Setting HDL Library	79
4.4.2.2 Launching HDL Workflow Advisor.....	80
4.4.2.3 Set target device and Synthesis Tool	81
4.4.2.4 Set target frequency.....	81
4.4.2.5 Check model settings.....	81
4.4.2.6 Set HDL Options	82
4.4.2.7 Generate RTL code and testbench.....	82
4.4.2.8 Create project	83
4.4.2.9 Perform Synthesis and P/R	83
4.4.3 The generated HDL.....	84
4.4.4 Synthesis & implementation	84
4.4.5 Generating bitstream file using Xilinx Vivado.....	87
4.4.6 FPGA in the loop.....	88
4.5 Conclusion	90
General conclusion	92
References	94

Table of figures

Figure 1-1 Four-inch polycrystalline silicon photovoltaic cell [7].	19	
Figure 1-2 Characteristic Curves [9].	20	
Figure 1-3 Shaded PV Cell [10].	21	
Figure 1-4 Shaded PV Cell with Bypass Diode Protection [10].	21	
Figure 1-5 Illumination influence [13].	22	
Figure 1-6 Temperature influence [13].	23	
Figure 1-7 Schematic diagram of the MPPT converter [15].	24	
Figure 2-1 Classification of Soft-Computing methods [27].	29	
Figure 2-2 Bio-inspired methods used with MPPT techniques in a PV system [4].	31	
Figure 2-3 Fireflies.	33	
Figure 2-4 Initial locations for 12 fireflies.	35	
Figure 2-5 Firefly position update shown with vectors.	37	
Figure 2-6 After one iteration	Figure 2-7 After five iterations	38
Figure 2-8 Pseudo code of firefly algorithm.	39	
Figure 2-9 Flow chart of the firefly algorithm [2].	43	
Figure 2-10 Flow chart of the PSO algorithm [2].	44	
Figure 3-1 Block diagram for PV system simulation.	48	
Figure 3-2 Simulation for different values of the controller parameters. FA_MPPT.	50	
Figure 3-3 Evolution of the output power and the duty cycle of the FA_MPPT under standard conditions ($I_{rr}=1000 \text{ W/m}^2, T=25 \text{ }^\circ\text{C}$).	53	
Figure 3-4 Evolution of the output power of FA and PSO MPPT under standard conditions.	53	
Figure 3-5 power evolution of the FA and PSO MPPT controllers for rapid changes in irradiation.	54	
Figure 3-6 power evolution of the FA and PSO MPPT controllers for rapid changes in temperature.	55	
Figure 3-7 P-V characteristic for different shading patterns.	57	
Figure 3-8 The extracted power during the shading pattern #1 with, (a) P-V Trajectory tracking, (b) FA, (c) PSO.	58	
Figure 3-9 The extracted power during the shading pattern #2 with, (a) P-V Trajectory tracking, (b) FA, (c) PSO.	59	
Figure 3-10 The extracted power during the shading pattern #3 with, (a) P-V Trajectory tracking, (b) FA, (c) PSO.	60	
Figure 3-11 The extracted power during the shading pattern #4 with, (a) P-V Trajectory tracking, (b) FA, (c) PSO.	61	
Figure 4-1 Architecture of an FPGA [32].	66	
Figure 4-2 design approaches.	68	
Figure 4-3 New Design Approach.	70	
Figure 4-4 HDL Coding methods for FPGA Programming [34].	72	
Figure 4-5 Method to generate HDL Code from MATLAB and Simulink [35].	72	
Figure 4-6 FPGA design flow based on VIVADO Design Suite.	75	
Figure 4-7 HDL Workflow Advisor.	76	

Figure 4-8 Launching HDL Coder in MATLAB.....	77
Figure 4-9 Specifying the required files.....	77
Figure 4-10 Input types	78
Figure 4-11 running fixed-point conversion	78
Figure 4-12 The target device.....	78
Figure 4-13 HDL Code Generation.....	79
Figure 4-14 Synthesis and Analysis	79
Figure 4-15 HDL Documentation Blocks	80
Figure 4-16 target device and synthesis tool	81
Figure 4-17 Target frequency	81
Figure 4-18 Check model settings	81
Figure 4-19 Set HDL Options.....	82
Figure 4-20 Generate RTL code and testbench.....	82
Figure 4-21 Create Project.....	83
Figure 4-22 Synthesis task	83
Figure 4-23 The generated VHDL in Vivado	84
Figure 4-24 Vivado's report	85
Figure 4-25 The synthesized FA-MPPT	85
Figure 4-26 Synthesis design	86
Figure 4-27 Artix7	87
Figure 4-28 PROGRAM AND DEBUG	87
Figure 4-29 auto connect option in Vivado	87
Figure 4-30 Programmed Device	88
Figure 4-31 Verify Code with HDL Test Bench.....	89
Figure 4-32 HDL testbench result	90

Tables

Table 1 Parameter and notation of FA	34
Table 2 Terminologies in FA and PV system	40
Table 3 GPV characteristics	48
Table 4 Simulation results for different parameter values	51
Table 5 Irradiation setup of the four shading patterns (W/m ²)	56
Table 6 Simulation results of the tracking speed, accuracy and efficiency	63

Abbreviation List

PVG: Photovoltaic Generator
PSO: Particle Swarm Optimization
PV: Photovoltaic
FA: Firefly Algorithm
MPP: Maximum Power Point
GMPP: Global Maximum Power Point
LMPP: Local Maximum Power Point
MPPT: Maximum Power Point Tracking
D: Duty Cycle
DC: Direct Current
P: Output Power of the photovoltaic module
FLC : Fuzzy Logic Controller
AFLC : Adaptative Fuzzy Logic Controller
INC:Incremental Conductance
BSO: Bat Swarm Optimization
GWO: Gray Wolf Optimization
GPS:Generalized Pattern Search
SFL:Shuffled Frog Leaping
PC:Parasitic Capacitance
P&O:Perturb And Observe
ANN: Artificial Neural Network
PWM: Pulse Width Modulation.
AI: Artificial Intelligence
ACO: Ant Colony Algorithm
ABC: Artificial Bee Colony
CSA: Cuckoo Search Algorithm
FPA: Flower Pollination Algorithm
PSC:Partial Shading Conditions
SI:Swarm Intelligence
HDL: Hardware Description Languages
DUT: Design Under Test
FPGA: Field-Programmable Gate Array
MBD: Model Based Design
FIL: FPGA In the Loop

General Introduction

General Introduction

The modern civilization with all its manifestations is based on the consumption of energy, a large part of which is from fossil sources. This consumption, which is reaching frighteningly high levels, is reducing the reserves of this type of energy for future generations, leading to the intensification of efforts and calls to seek new sources of inexhaustible energy. Hence the alternative solution of renewable energies that has been emerging since the sixties of the twentieth century. Among these renewable energies we find solar energy, wind energy, hydrogen and biomass.

Considering its geographical location, Algeria has an exceptional solar deposit, thanks to its large surface which exceeds ten hours per day during several months. The duration of insolation on almost the entire national territory exceeds 2000 hours annually and reaches 3900 hours (highlands and Sahara). The energy received daily on a horizontal surface of 1 m² is about 5 kWh on most of the national territory, that is to say nearly 1700 kWh/m²/year in the North and 2263 kWh/m²/year in the South of the country. Therefore, the installation of large-scale solar panels in the south of the country can provide a sufficient amount of energy that fulfil to public needs and also contributes in reducing carbon footprint and the greenhouse gases (GHG).

The discovery of the photovoltaic effect, which consists of the conversion of light into electricity through sensors or photovoltaic cells, has allowed the development of this axis. This new source of electricity can produce energy ranging from a few milliwatts to megawatts.

Among the major problems encountered within photovoltaic (PV) systems is the random changes in solar energy received by the cells, which is due to the variation of atmospheric conditions (temperature and radiation) or to the non-uniform radiation. This latter is called partial shading condition PSC, and it is the result of having a shadowed panel. This shadow comes from moving clouds, a flock of birds moving, building or dust on the surface of the panels. During those conditions the photovoltaic system power generation decreases.

These different findings have motivated us to work on the optimization and control of the photovoltaic system using maximum power point tracking MPPT. This latter ensures a better exploitation of the solar energy source while ensures a robust control of the fluctuation encountered due to the partial shading effect and the temperature variations. The phenomena of partial shading in recall, causes high power losses. From an economic view point, these hindrances reduce the profitability of the production of photovoltaic systems.

Many works found in the literature have proposed technical solutions to reduce, or even eliminate, these phenomena. The ultimate goal is to operate photovoltaic systems at their maximum power points under any environmental conditions.

Several peaks occur in the power-voltage curve under partial shading conditions. Under such conditions, many conventional maximum power point tracking methods such as the P&O may be invalid due to blocking in a local maximum power point. These methods lack the intelligence to distinguish between a local maximum (LMPP) and a global maximum (GMPP) when the P-V characteristic is multimodal. AI methods such as the ANN, can achieve satisfactory results in global MPP tracking, but in the other hand they suffer from drawbacks. They need a lot of adjustments and training to obtain the optimal MPP.

In order to overcome this problem, during the last decade, several researchers have proposed new methods based on algorithms inspired by nature: bio-inspired algorithms. This latter showed an astonishing result in finding MPP compared to conventional methods and AI, with regards of their simplicity, low computational time and less necessity of external equipment. We can mention some of popular bio-inspired algorithms in the field MPPT are: Ant Colony Algorithm (ACO) [1], Particle Swarm Optimization (PSO) [2], Artificial Bee Colony (ABC) [3], Firefly Algorithm (FA) [4]...

Thus, in our final project, we will present firefly (FA) algorithm for maximum power point tracking MPPT in a photovoltaic system. This algorithm is inspired by the social behaviour of Fireflies, namely the firefly (FA) algorithm.

In order to highlight the performance of the proposed algorithm, it will be compared with a Particle Swarm Optimisation (PSO) algorithm using MATLAB software. The simulation results will be detailed and discussed

To get the targeted algorithm in presence, the implementation of the generated HDL of the FA algorithm will be achieved using the MATLAB HDL Coder software, beside Xilinx Vivado in order to deploy it onto an ARTIX7 board.

This brief is organised into four chapters.

- Chapter 1 presents a brief description of the electrical energy in the world generally, and Algeria specifically, it also presents the basic concepts related to photovoltaic systems and the maximum power point tracking.
- Chapter 2, introduces the different maximum power point tracking methods and then presents the main features of the Firefly (FA) algorithm as a bio-inspired optimization method. The application of the proposed algorithm and the Particle Swarm Optimization (PSO) algorithm as MPPT is discussed. The PSO technique is used to perform a comparative study in terms of performance.
- Chapter 3 presents a MATLAB-SIMULINK simulation of the realized photovoltaic system. The tracking and search performance of the MPP using FA and PSO algorithms are evaluated and compared in terms of tracking speed, accuracy and the ability to handle partial shading conditions.
- Chapter 4 presents the model-based design (MBD) conceptions and the different approaches of HDL generation for the FA-MPPT, mainly by targeting the HDL Coder and presenting the required steps until the deployment in the target device, beside simulating the generated HDL Code [5].

Chapter 1

Photovoltaic Electrical Energy

1. Photovoltaic Electrical Energy

1.1 Introduction

Currently in the world, electrical energy is produced, for the most part, through the combustion of fossil fuels (coal, oil and natural gas) or radioactive elements. The harmful impact of these processes of electricity production on the terrestrial environment is very important due to the emission of greenhouse gases and the generation of radioactive waste. For ecological considerations and to remedy the harmful effects of pollution due to fossil fuels, mankind is increasingly moving towards the use of clean and renewable resources, including solar energy.

The production of electricity by photovoltaic systems using solar energy is attracting greater interest, is growing significantly and is finding more and more new applications. Hence, the interest to study in detail this energy resource including its photovoltaic component. The constituent elements of photovoltaic systems and their functions.

Algeria, given its geographical position, has one of the highest solar deposits in the world. On the whole national territory, the global solar energy received per day on a horizontal surface of one square meter varies between 5.1 KWh in the North and 6.6 KWh in the Great South.

1.2 Photovoltaic Systems

A photovoltaic panel transforms the Sun's energy into renewable electricity.

In concrete terms, a solar panel is composed of photovoltaic cells (made from crystalline silicon) that have the ability to transform the Sun's rays into direct current.

However, the appliances work with alternating current. That's why the electricity produced by the solar panel will be sent to an inverter whose role is to transform the direct current into alternating current.

A grid-connected PV system is composed of a set of components starting from the PV panels and ending with the grid. Several factors affect the performance of the grid-connected PV system. These factors can be: [6]

- Natural: such as weather conditions like solar radiation, ambient temperature and wind speed.
- Artificial: based on the different topologies of the conditioning devices (DC/DC converters and DC/DC converters and DC/AC converters), input and output filters and transformers.

1.3 The Photovoltaic Cell reviews

1.3.1 Definition

Photovoltaic cells (photon: grain of light and volt: unit of voltage) are semiconductor electronic components (usually made of silicon in its different forms).

They directly convert light energy into low voltage direct current electricity (photovoltaic effect). Since the light energy is the sun, they are referred to as solar cells.

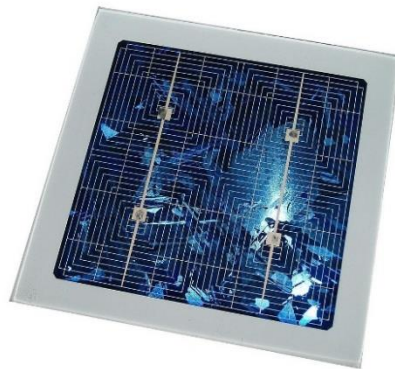


Figure 1-1 Four-inch polycrystalline silicon photovoltaic cell [7].

1.3.2 Definition

The solar cell, round or square, is the basic element of a solar system. A set of cells forms a solar module, in a module the cells are electrically connected to each other and encapsulated, thus protected from external agents. Several modules form a solar panel. Several panels form a system or solar field, to which are added protections, a regulator, an energy storage system (battery), control and measurement devices, an inverter [8].

The term PVG photovoltaic generator is used to refer to a PV module or panel, depending on the application. However, it can refer to the entire PV system.

1.3.3 Characteristic Curves

Photovoltaic panels can be wired or connected together in either series or parallel combinations, or both to increase the voltage or current capacity of the solar array. If the array panels are connected together in a series combination, then the voltage increases and if connected together in parallel then the current increases.

The electrical power in Watts, generated by these different photovoltaic combinations will still be the product of the voltage times the current, ($P = V \times I$). However, the solar panels are connected together, the upper right-hand corner will always be the maximum power point (MPP) of the array.

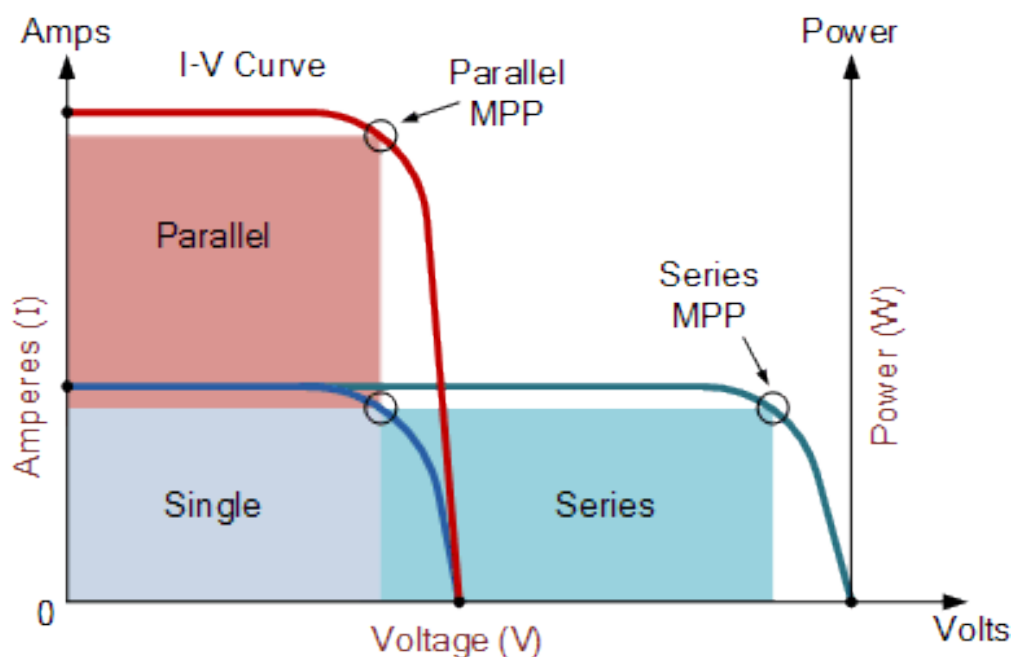


Figure 1-2 Characteristic Curves [9]

1.4 Photovoltaic Cell Shading

The solar Cell No.2 in the figure below in the string is either partially or fully shaded while the remaining two cells in the series connected string are not, that is they remain in full sun. When this occurs, the output of the series connected string will reduce dramatically with less current as shown.

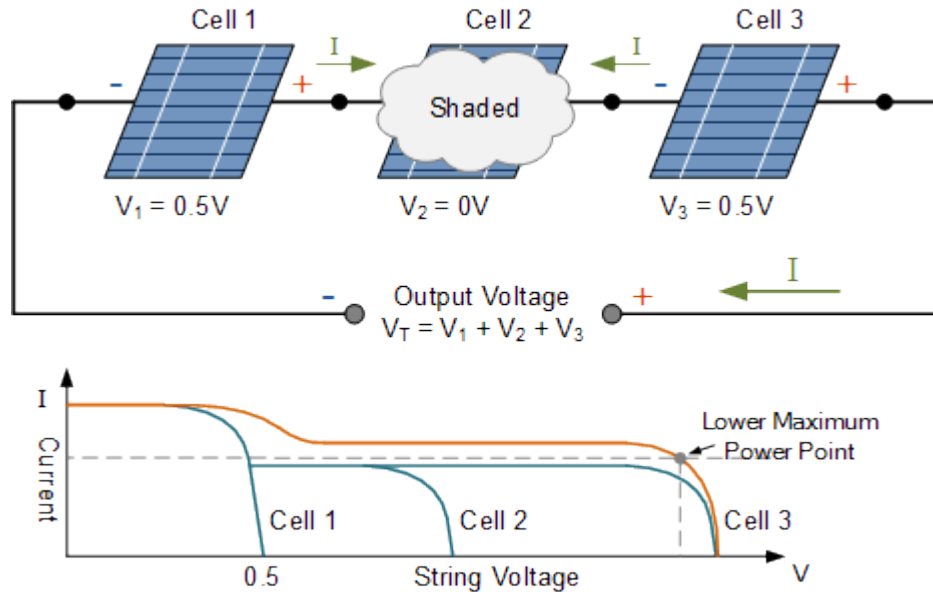


Figure 1-3 Shaded PV Cell [10]

The forced current is therefore "pushed", so that the shaded solar cells heat up enormously. This can damage the solar cells, or even deform or shatter the glass plate of the panels. To prevent this, bypass diodes are incorporated into a solar panel, when there are solar cells that produce less energy, the current that must flow through these cells is diverted by a bypass diode. Thus, the panel is protected when there is shade on a part because the panel has several bypass zones, the electricity from the solar cells that have enough light is not lost.

Under this condition, cell two stops producing electrical energy and behaves like a semiconductive resistance. Due to the shaded cell generating reverse power, it forward biases the parallel connected bypass diode (i.e., it turns it "ON") diverting current flow of the two good cells through itself as shown [10].

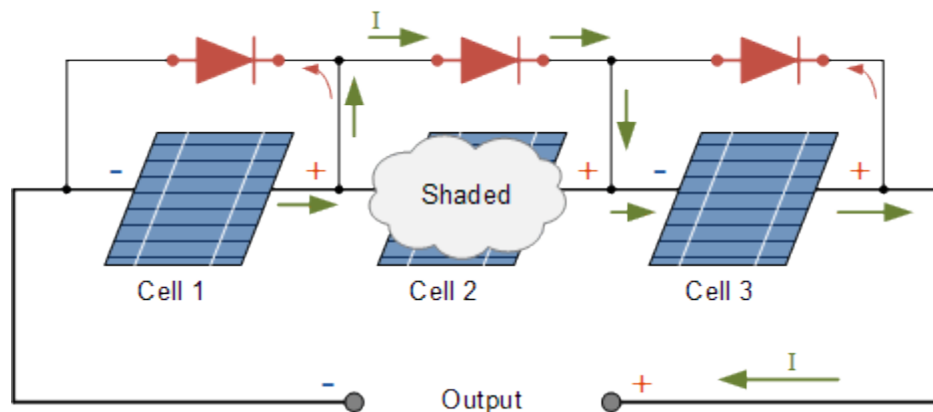


Figure 1-4 Shaded PV Cell with Bypass Diode Protection [10]

As we have said PV array characteristic gets altered when complete or some parts of this system are shadowed by clouds, tall trees, deposition of dust, and surrounding buildings or shadow created by one of the PV arrays to others in outside environments. This

consequently leads to nonuniform irradiance conditions and results in superfluous losses in cells of the module in the form of overheating commonly known as hot spot. Hot spot phenomenon causes detrimental effects on partially shaded cells/modules, reducing the overall efficiency greatly. However, shunt resistance is taken as one of the parameters correspond to the shading effect in the mathematical modeling of a solar cell. All the three modules of the string receive uniform irradiance, thus corresponding P-V characteristics exhibit one MPP as depicted in the next Figure with the blue line. PSC occurred due to all the three modules receive different insolation. P-V characteristic gets altered and is not as smooth as in case of the uniform condition and exhibits multiple peaks under partial shading condition as shown in Figure below with the green line. PV characteristic shows a very complex behavior as a bypass diode (BD) is connected across each module to suppress the hot spot effect. Available output power across string under PSC is reduced as conventional MPPT gets trapped in one of the local peaks [11].

1.5 Effects of Climate Variations on the photovoltaic Module

1.5.1 Influence of the illumination

The brightness has a considerable influence on the performance of the cells.

As shown in this graph, the short-circuit current (I_{sc}) increases proportionally with the illumination, while the open-circuit voltage (V_{oc}) varies very little (about 0.5 V), thus, the greater the cloud cover, the lower the current generated [12].

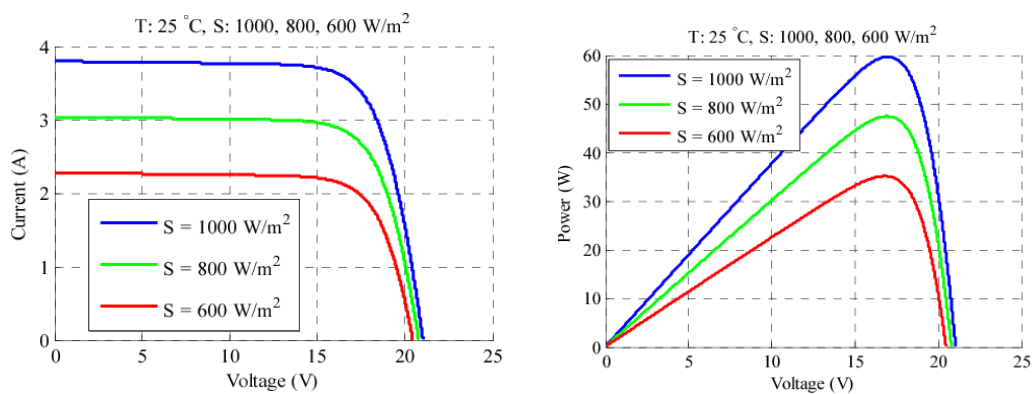


Figure 1-5 Illumination influence [13]

1.5.2 Influence of temperature

The temperature has a considerable influence on the behavior of the cell and therefore on its performance. This influence is mainly reflected by a decrease in the voltage generated (and a very slight increase in current) [14].

According to the models, this behavior induces, per degree, a loss of 0.5 % of the output compared to the maximum output of the cell. We can therefore understand the importance of a correct ventilation at the back of the panels!

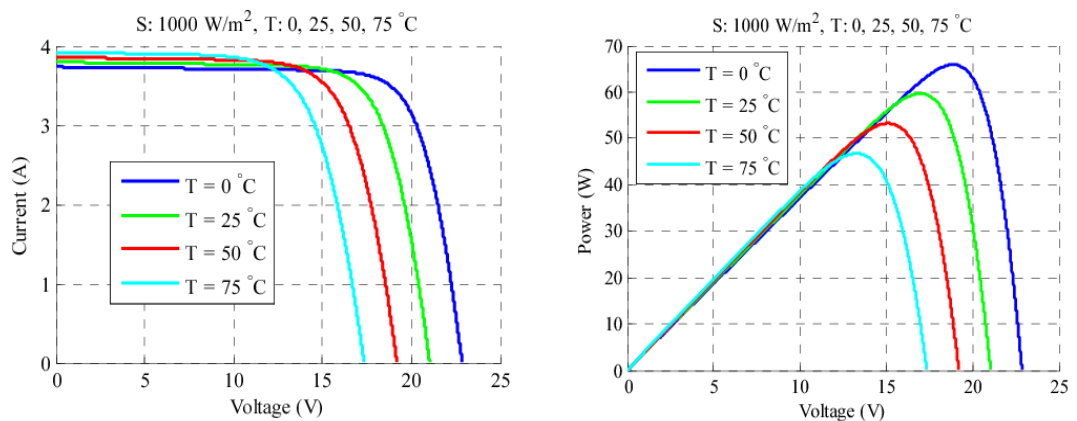


Figure 1-6 Temperature influence [13]

1.6 MPPT

The power delivered by a PVG depends strongly on the level of sunlight, the temperature of the cells, the shading and also on the nature of the load fed. It is therefore very unpredictable. As we have already seen previously, the PV power characteristic curve has a maximum power point (MPP) corresponding to a certain operating point with coordinates V_{MPP} for the voltage and I_{MPP} for the current (see figure below). Since the position of the MPP depends on the level of sunlight and the temperature of the cells, it is never constant over time.

Figure 1-22 shows the schematic diagram of a typical MPPT converter. The MPPT control varies the duty cycle of the DC/DC converter which has three types, Buck-Boost converter, Buck converter: and Boost converter: in order to extract at each moment, the maximum power available at the PVG terminals and to transfer it to the load, this matching stage DC/DC converter is used. This stage acts as an interface between the two elements. It ensures, through a control action, the transfer of the maximum power supplied by the generator. The MPPT algorithm can be more or less complicated to find the MPP. In general, it is based on the variation of the duty cycle of the converter according to the evolution of the input parameters of the latter (I and V and consequently of the power of the PVG) until the MPP is reached. Several algorithms are present in the literature, such as the Bio-Inspired Algorithms whose are beside others: Practical Swarm Optimization (PSO), Differential Evolution (DE), Artificial Bee Colony (ABC) and the Firefly Algorithm (FA) which will be our main subject in the coming chapters.

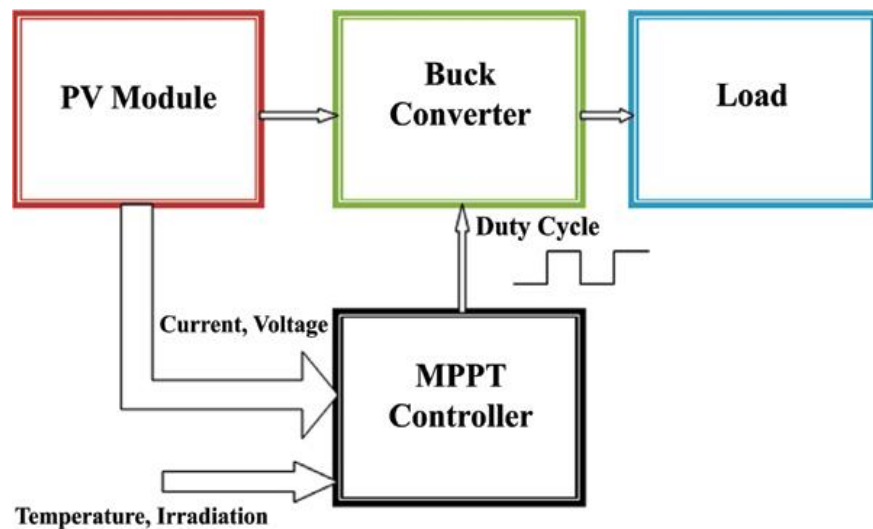


Figure 1-7 Schematic diagram of the MPPT converter [15]

1.7 Conclusion

In this chapter, we began by giving a brief description of the state of the art of electricity generation in general and renewable electricity generation in particular in the world and Algeria. Then, we introduced the general terms and concepts related to the field of photovoltaics and the principle of conversion of solar energy into electricity through the photovoltaic effect. It describes the role and characteristics of the photovoltaic cells. Finally, the effects of varying climatic conditions on the operation of these systems are described. This leads to the principles of finding the maximum power point MPP used for better solar energy conversion. And in the following sections, we will present the firefly algorithm by analyzing and simulating it in different conditions.

Chapter 2

Maximum power point tracking
methods. State of art

2. Maximum power point tracking methods. State of the art

2.1 Introduction

The output power produced by a solar PV is not stable, it depends strongly on solar irradiation, temperature and load. As explained in Chapter 1, this variation is nonlinear and the power generated from the PV array is hardly predictable. When the solar panel is coupled directly to the load, it does not operate at the maximum power point (MPP) unless the load is properly matched to the PV system, but this matching won't last because of the continuous variation of environmental conditions.

Therefore, an MPPT control algorithm is needed to get superior efficiency and continuously extract the maximum power available from the PV module under static, varying and partial shading weather conditions.

To solve this MPPT optimization problem, many algorithms have been proposed in the literature. These methods react differently in different situations, each has its own advantages and disadvantages and varies in terms of complexity, efficiency, cost, convergence speed, hardware implementation etc.

In this chapter, we will first present the performance criteria of an MPPT controller. Then, we will adapt the Firefly bio-inspired algorithm to be implemented in an MPPT controller, present the basic concepts, describe the mathematical model and the different steps of algorithm. Moreover, in order to compare the performances of the studied controller, we have chosen to use a controller based on another bio-inspired algorithm, the PSO algorithm.

2.2 Performance criteria of MPPT algorithm

There are large numbers of the applied MPPT control techniques available to date that have been developed by researchers to maintain the PV array at its MPP. In order to compare between these techniques and to know which one better to implement for particular application, the followed specifications should be taken in consideration [16].

It should be noted that in the literature, it is difficult to find standard evaluation factors, some of which cannot even be measured accurately, which is an approximation.

- **Tracking accuracy and efficiency:**

High-performance MPPT algorithm has the ability to find the global MPP (GMPP) precisely at the right time and under rapidly changing working conditions. To demonstrate the performance of a technique, tracking accuracy and efficiency are important factors to evaluate.

Tracking accuracy is defined as the difference between the estimated power (P_{MPP_est}) of the PV array tracked by the algorithm and the theoretical power (P_{MPP_theor}) during the same period. It is considered as the error (ERR), which is expressed as [17]:

$$ERR (\%) = \left(1 - \frac{P_{MPP_est}}{P_{MPP_true}} \right) * 100$$

Tracking efficiency is assessed in terms of the percentage of energy generated from the PV array during the simulation period, the equation below mathematically defined the tracking efficiency as [18]

$$\eta_{MPPT} (\%) = \frac{E_{MPPT}}{E_{MPP_true}} = \frac{\int P_{MPPT} dt}{\int P_{MPP_true} dt} * 100$$

- **Tracking speed (convergence time)**

MPPT control algorithms must converge fast to MPP under rapidly changing weather conditions. The faster the dynamic response of the MPPT algorithm, the more solar energy is used.

- **Steady-state error**

In steady-state, the algorithm should set the duty cycle to a constant value and minimize the oscillation of the operating point around the found GMPP. High steady state oscillations cause a loss of power and presumably reduce the PV system efficiency.

2.3 Classification of MPPT methods

In literature, MPPT controllers have been classified as conventional methods and advanced (soft computing) methods. Conventional methods use very simple algorithms with low cost and they were initially employed for the PV systems. Contrary to the conventional methods, advanced methods are relatively complex and require high computation capacity [27].

2.3.1 Conventional methods

MPPT conventional techniques are widely used due to their simplicity, low cost and easy hardware implementation. Under uniform insolation conditions or slow irradiance changes, they provide a good tracking performance. However, these methods fail to find the GMPP under partial shading conditions (PSCs), even if they converge to the GMPP, they take a long time with low accuracy and with large steady state oscillation.

The most applied conventional MPPT methods are:

- Perturb and Observe (P&O) [19] [20]
- Incremental Conductance (INC) [19] [20]
- Parasitic Capacitance (PC) [21]
- Constant Voltage [22]
- Constant Current [23]

2.3.2 Advanced methods

Advanced methods known as soft computing methods have been proposed as alternative approaches to solve the conventional methods obstacles. They have shown superior tracking performance under dynamic weather and partial shading conditions due to their robustness, flexibility, and reliability. Advanced methods can be classified into two subcategories as artificial intelligence (AI) and Bio-inspired (BI) methods. The figure below illustrates a classification of soft computing methods [27].

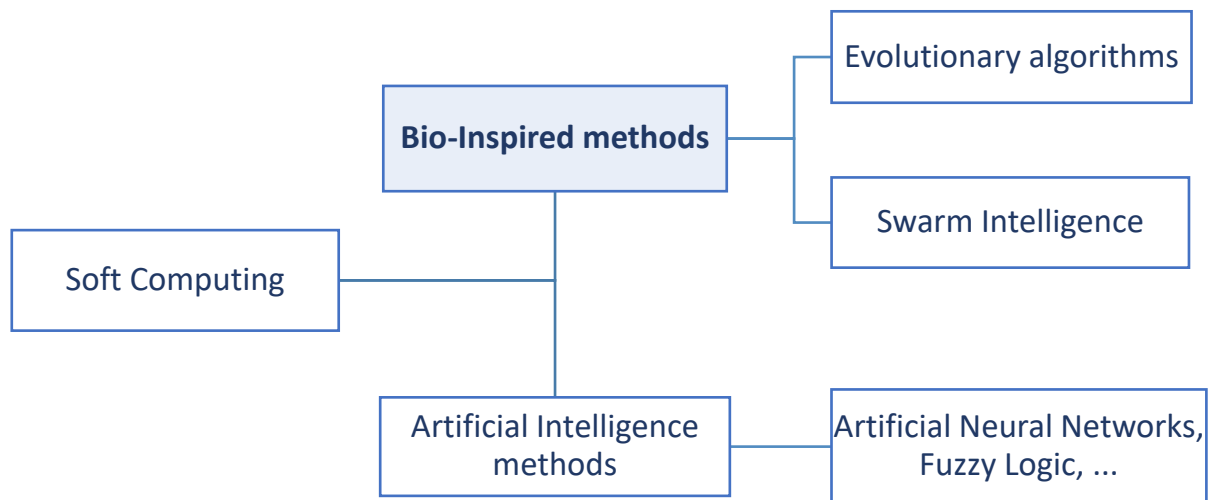


Figure 2-1 Classification of Soft-Computing methods [27].

As illustrated in the figure above, soft computing techniques include a number of methods, ranging from those based on artificial intelligence, such as neural networks, fuzzy logic, hybrid algorithms, etc., to newly introduced methods, namely bio-inspired methods.

These last ones have been very popular in the scientific community. In the following we will list the existing MPPT methods based on AI and bio-inspired.

2.3.2.1 AI-Based Methods

Several studies have been proposed in the literature working on MPPT methods based on artificial intelligence. Among these, the most commonly used in MPP research, we can cite:

- Artificial neural networks (ANN) [24]
- Fuzzy logic controller (FLC_MPPT) [25]
- Adaptive Fuzzy logic Controller (AFLC_MPPT) [25].
- Genetic Algorithms combined with Fuzzy Logic (GA_FLC_MPPT), or with Artificial Neural Networks (GA_ANN_MPPT) [26]

AI methods achieve satisfactory results in accurately tracking GMPPs under PSCs. But they suffer from drawbacks such as the fact that the ANN requires a lot of effort to be adjusted to obtain the optimal MPP, and also needs to be well trained using a large number of measurements before its actual operation in the MPPT control unit to get accurate results. FLC controllers depend on rule base training and membership functions. There is no standard way to precisely define the controller parameters. FLC with constant parameters is inappropriate in an application where the operating conditions change over a wide range and

the available expert knowledge is unreliable. The solution to this problem is the use of an adaptive fuzzy logic controller (AFLC). The AFLC has the ability to readjust its parameters to achieve the best performance.

Moreover, these methods are very computationally intensive. This is a major drawback for them. This has led the scientific community to propose new intelligent methods to overcome the limitations of traditional artificial intelligent methods. These methods are based on the concept of bio-inspired. In what follows, we will focus on these new methods.

2.3.2.2 Bio-Inspired (BI) methods

Recently, BI methods have attracted much attention due to their potential to find optimal solutions in complex optimization problems (such as multimodal objective functions). These methods are based on iterative improvement of a population of solutions or a single solution (e.g., Tabu Search) and mainly use randomization and local search to solve a given optimization problem [27].

As illustrated in Figure 2.2, the two most predominant classes of bio-inspired algorithms are evolutionary algorithms and swarm intelligence-based algorithms [4]. These algorithms are derived from the study of natural evolution and the swarming behaviour of living beings.

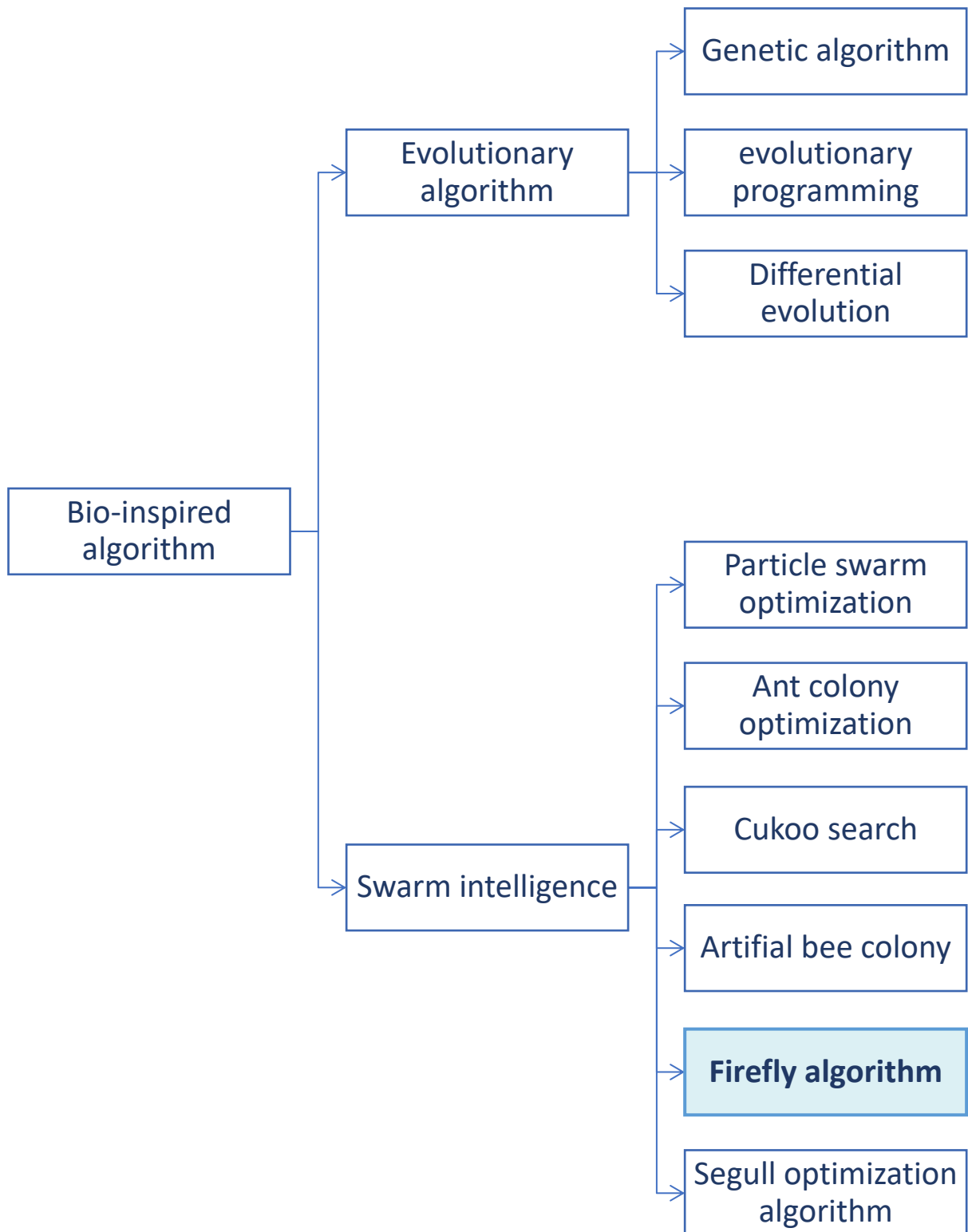


Figure 2-2 Bio-inspired methods used with MPPT techniques in a PV system [4].

In view of the advantages of Bio-Inspired methods, and the enthusiasm of the scientific community, they have a promising future in the field of renewable energies, given the number of existing works that still remains below.

Among the controllers designed based on bio-inspired algorithms, we can mention:

- MPPT controller based on the Ant Colony Algorithm (ACO), [1]
- MPPT controller based on the Particle Swarm Optimization (PSO) algorithm, [2]
- MPPT controller based on the Artificial Bee Colony (ABC) algorithm, [3]
- MPPT controller based on the Cuckoo Search Algorithm (CSA), [28]
- MPPT controller based on the Firefly Algorithm (FA), [4]
- MPPT controller based on the Bat Swarm Optimization Algorithm (BSO), [29]
- MPPT controller based on the Flower Pollination Algorithm (FPA), [29]
- MPPT controller based on the Gray Wolf Optimization (GWO) algorithm, [29]
- MPPT controller based on the Generalized Pattern Search (GPS) algorithm, [29]
- MPPT controller based on the Shuffled Frog Leaping Algorithm (SFL) [29].

Through the various works proposed in the literature on MPPT controllers designed around bio-inspired methods, and in view of their advantages, this motivated us to choose one of these methods in the context of our PFE, namely, Firefly algorithm (FA).

This latter has shown a good performance to find optimal solutions to complicated optimization issues faced in many applications, such as, optimization routing in wireless sensor networks, optimization antenna design, optimization scheduling, feature extraction in digital image processing, feature selection and detection of fault diagnostics and localizing the defects, etc.

In the context of power optimization in photovoltaic systems, this algorithm is recently being developed. In what follows, we will target mainly this algorithm in order to optimize the produced power. In the following section, we will focus on applying this algorithm in Maximum Power Point Tracking (MPPT) controller for PV systems through the implementation of its mathematical model to find the MPP of a photovoltaic generator.

2.4 The Firefly algorithm

The firefly algorithm, which was first developed by Xin-She Yang in late 2007 and 2008 at Cambridge University [30], is a population-based optimization that belongs to metaheuristic and stochastic swarm intelligence (SI) techniques.

This nature-inspired algorithm depends on the group behaviour of lightning bugs- commonly known as fireflies, it mimics their flashing pattern in the summer sky in the tropical temperature regions.



Figure 2-3 Fireflies

The basic idea of formulation of the FA is the attractiveness of fireflies, that fireflies attract each other according to the brightness (the less bright one will be attracted to the brighter one). Based on that, FA simulates the movement of fireflies to perform the position iteration, finds the optimal position and completes the optimization process.

2.4.1 Basic concepts

To simplify the description of FA, various aspects of fireflies in nature must be idealized. First, real fireflies flash in discrete patterns, whereas modelled fireflies will be considered always bright. Second, the three following rules are considered to govern the algorithm and create the behaviour of a modelled firefly [30].

- All fireflies are unisex so the attraction between them is not according to their gender.
- Attractiveness is proportional to their brightness and they both decrease as their distance increases. Thus, the lower bright firefly will move towards the brighter one. If there is no brighter one than a particular firefly will move randomly.
- Objective function which being optimized determines the brightness of Firefly.

When comparing the brightness of any two fireflies, the locations of the fireflies must be considered. It is known that the intensity of light (I) keeps on decreasing as the distance (r) increases from the light source, it follows the inverse square law ($I \propto 1/r^2$). In addition, the air absorbs the light and affects its intensity at a given distance from the light source. Due to these limiting factors, fireflies are visible within limited range, usually within several hundreds of meters that they can communicate.

Table 1 Parameter and notation of FA

Sr. No.	Parameter	Notation in algorithm
1	Brightness	Objective Function
2	Beta (β)	Attractiveness
3	Alpha (α)	Randomization parameter
4	Gamma (γ)	Absorption coefficient
5	Number of generations (G)	Iteration
6	Number of Fireflies (N)	Population
7	Problem Dimension (D)	Dimension

2.4.2 Steps of the Firefly Algorithm [31]

2.4.2.1 Initialization

FA starts by assigning the fireflies to random positions, the position of a firefly refers to the values of the parameters of the objective function to be optimized. The objective function is then evaluated from the newly acquired position of each firefly. The light intensity (or brightness) is related to the evaluation. For a maximization problem, the brightness of a firefly I at a particular position x can be defined as $I(x) \sim f(x)$, where $I(x)$ is the brightness and $f(x)$ is the value of objective function at this position.

For a demonstration of initialization, shown in figure 2.4 is the initial location of twelve fireflies, in the 2-dimensional sphere function problem. The global minimum for this problem is 0 at the location (0,0), the origin.

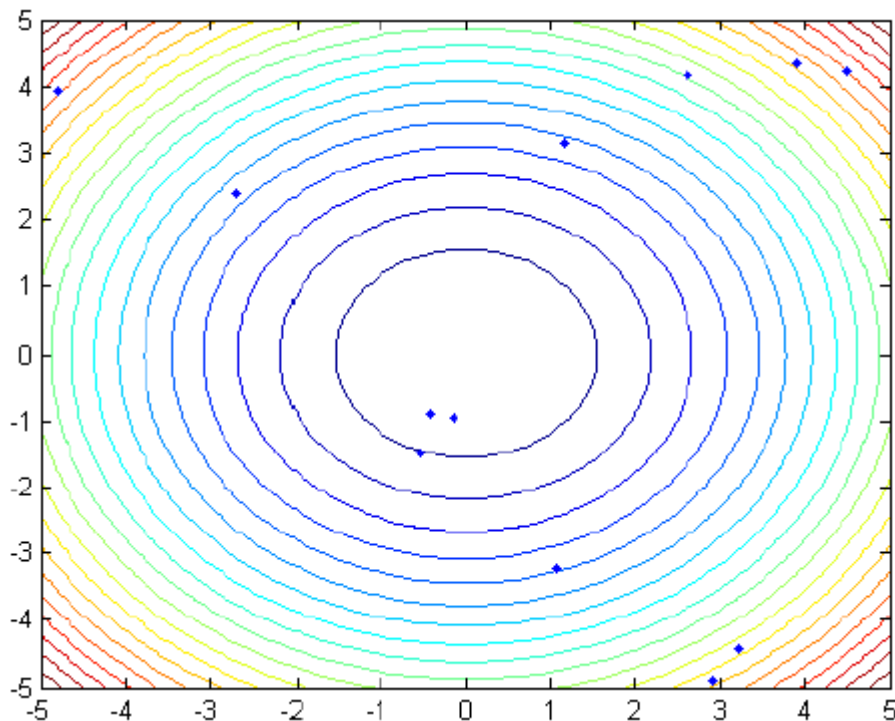


Figure 2-4 Initial locations for 12 fireflies

2.4.2.2 Exploration

After initialization, each firefly's brightness is compared against all the rest in the population, and the movement will be towards every brighter firefly it encounters. First, the distance between the fireflies, r , has to be calculated. For the general case the Cartesian distance is appropriate. The Cartesian distance that separates any two fireflies i and j at x_i and x_j , respectively, in d -dimensional space is given as:

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

The attractiveness β is relative to the position of other fireflies. So, after the distance between two fireflies is known, the attractiveness β can be determined. To simplify the formulation, the attractiveness of a firefly is determined as a function of the light intensity. Thus, the attractiveness will vary with the degree of absorption of light [31]. Light intensity $I(r)$ changes according to the inverse square law shown in equation (2.1).

$$I(r) = I_s/r^2 \quad (2.1)$$

where I_s is the source intensity. Given a medium with a constant light absorption coefficient γ , the light intensity I will change with distance r as shown in equation (2.2) [31].

$$I = I_0 e^{-\gamma r} \quad (2.2)$$

where I_0 is the original light intensity. In the expression I_s/r^2 singularity at $r = 0$ can be avoided by the combined effect of both equation (2.1) and absorption can be estimated by the Gaussian form that controls the decrease of light intensity

$$I(r) = I_0 e^{-\gamma r^2} \quad (2.3)$$

Since the firefly attractiveness varies linearly to the intensity experienced by adjacent fireflies. Attractiveness of a firefly can now be obtained by

$$\beta(r) = \beta_0 e^{-\gamma r^2} \quad (2.4)$$

where β_0 is the initial attractiveness at $r = 0$. Equation (2.4) defines a characteristic distance $r = 1/\sqrt{\gamma}$ over which the attractiveness varies significantly from β_0 to $\beta_0 e^{-1}$.

The attractiveness function $\beta(r)$ is selected to be a monotonically decreasing function. Equation (2.5) shows the basic form.

$$\beta(r) = \beta_0 e^{-\gamma r^m}, (m \geq 1) \quad (2.5)$$

where m is an integer greater than 1. For a constant γ , the characteristic length is:

$$r = \gamma^{-1/m} \rightarrow 0, m \rightarrow \infty \quad (2.6)$$

The movement of a firefly towards more attractive (brighter) firefly is determined by the attractiveness $\beta(r)$ and a random parameter. The random parameter is key for all metaheuristic algorithms; it allows the algorithm to escape from local optimums and to find the global optimal solution. A simple way to create a random walk distance is to use a uniform distribution in the range of $[0.5, 0.5]$. The movement can be either forward or backward, as both

positive and negative values are possible. However, if two parameters of the objective function have different ranges of possible values, a fixed range of random numbers would cause different relative randomness for each dimension. To solve this problem, the random numbers generated can be multiplied by the scale of the function dimension, S , to produce a vector of scaling values. With all the terms together, the position update equation for a firefly being attracted to firefly becomes [20]:

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_i - x_j) + \alpha S(rand - 0.5) \quad (2.7)$$

Where the first term is the current position of a firefly, the second term is used for considering a firefly's attractiveness to light intensity seen by adjacent fireflies, and the third term is used for the random movement of a firefly in case there are no brighter ones. Where $rand$ is a set of uniformly distributed random numbers in the range of $[0, 1]$, and α is a parameter controlling the amount of randomness. The randomness parameter α is typically in the range $[0, 1]$, where 0 corresponds to no randomness and 1 corresponds to being highly random.

Figure 2.5 shows the position update of the fireflies from their initial locations as shown in figure 2.1.

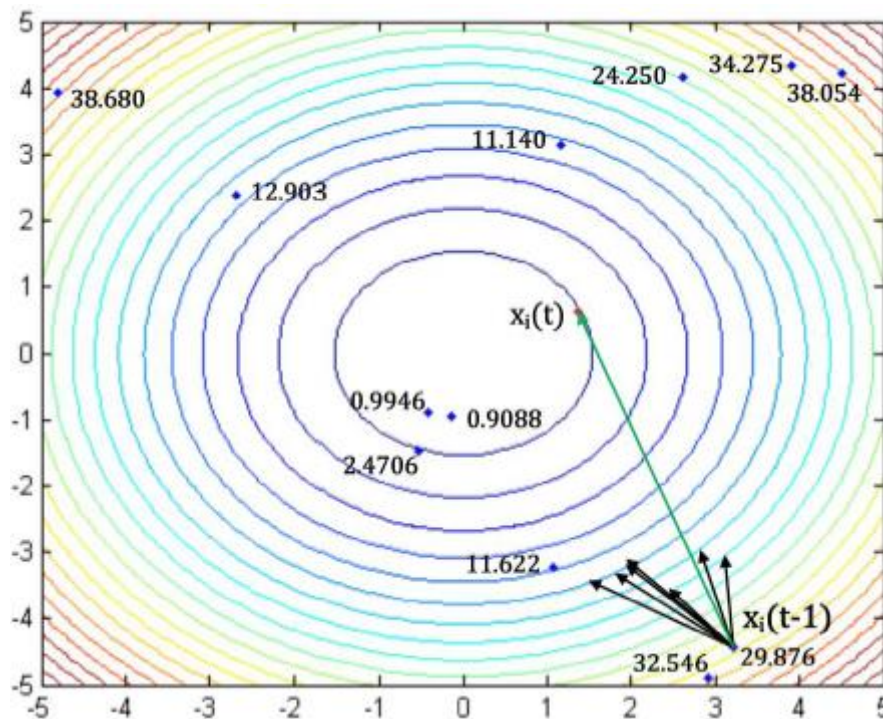


Figure 2-5 Firefly position update shown with vectors

After a firefly has been moved towards all brighter fireflies, its brightness is updated by evaluating the objective function in the new position. The new evaluation is also compared against the best found so far. If it is better, that position becomes the new best. In this way, if a firefly passes through a location better than any other found, but for whatever reason it or any other firefly does not end up there by the last iteration of the algorithm, that best location is still

recorded. After re-evaluation, the given firefly is updated, and the algorithm moves on to the next firefly. One generation of the algorithm is performed after this continues for every firefly.

Figure 2.6 shows the sphere function example after one complete generation.

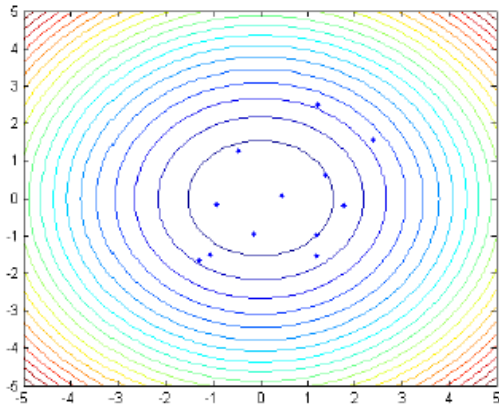


Figure 2-6 After one iteration

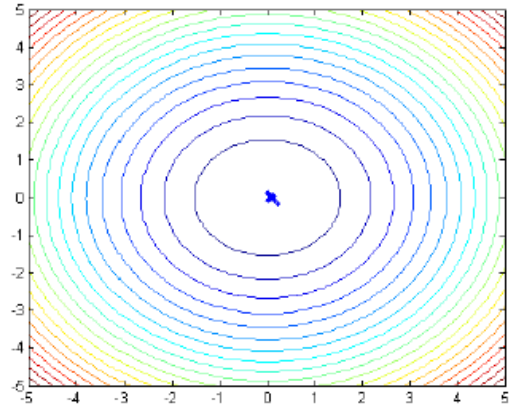


Figure 2-7 After five iterations

2.4.2.3 Checking the convergence criteria

After each generation, it must be determined if the algorithm is complete or not. The simplest method for deciding when to stop is to run for a set number of generations and see if it is enough for fireflies to converge to the optimal position. This method is commonly used for its simplicity, but it can be inefficient. Another method is to check how much better the results are getting. Since the best firefly is stored, the previous best can also be stored. If the difference between the best and previous best is very small, then likely the fireflies have converged on the optimal location. This method is more complicated, however, since the value cannot be too large, as the global optimum may not have been reached yet, but it cannot be too small, because even when the fireflies converge, there are still small movements being made, and thus the algorithm would never end. A third approach to ending the algorithm is to set a value that is “good enough,” so if the best firefly is as good or better than a given threshold value, we do not need to proceed any farther. This method is also easy to use, but it requires the knowledge of what is a good solution for the given problem.

Now that the entire algorithm has been covered, a summary in pseudocode is given in figure 2.8 below.

Firefly Algorithm

Given:

Objective function $f(x)$, where $x = (x_1, x_2, \dots, x_d)$ for d dimensions.

Light intensity I_i at x_i is determined by $f(x_i)$

Define $\alpha, \beta, \gamma, \delta$

Give random locations for N fireflies in D dimensions

Initial evaluation of all N fireflies

while (End condition not met)

 Increment t

 for $i = 1$ to NumFireflies

 for $j = 1$ to NumFireflies

 if ($I_j > I_i$)

 Move firefly i towards j in d -dimension

 end if

 end for j

 Evaluate new solution and update light intensity

 Check if the best position found so far

 end for i

 Reduce α

 Sort the fireflies

end while

Find the best firefly

Figure 2-8 Pseudo code of firefly algorithm

2.4.3 Algorithm parameters

Proper tuning of the FA parameters is very necessary for global optimum searching. The most important parameter in firefly algorithm is γ , it characterizes the variation of the attractiveness and it plays a very crucial role of how fast the algorithm converges and how the algorithm behaves. The parameter α allows the search process to escape the local search and search on the global scale.

As we will see in the next chapter, the implementation of this algorithm needs to set these parameters that guarantees a quick convergence to the optimal solution.

2.5 Application of the Firefly Algorithm (FA) for MPPT controller

The performance of the FA is exploited for the design of an MPPT controller to search for the optimal MPP in different working conditions.

In the implementation of this algorithm in MPPT, the position of the firefly (the regulated variable) represents the duty cycle that controls the DC-DC converter, and the objective function is chosen as the PV output power. The table 2 below shows how the FA terminologies match those of the PV systems.

Table 2 Terminologies in FA and PV system

Firefly Algorithm	PV system
Firefly position	Duty cycle of DC-DC converter (d)
Distance	Duty cycle difference (Δd)
Attractiveness	Exponential function of distance (Δd)
Brightness	Power (P_{pv})
Brightest Firefly	Global maximum power (P_{Gbest})
Dimension (number of variables)	one

The proposed method works as follow:

Step 1: Initialization

First, initialize the parameters of the FA, namely, β, γ, α and population size N .

In order to start the optimization, the duty cycle of the PWM signal is chosen to be the optimization variable. Thus, it is adjusted directly by the MPPT controller. Initially, a solution vector of duty cycles with a number of N particles is defined. The number of particles in the population must be chosen carefully. A larger number of particles allows for more accurate MPP tracking even under complicated shading patterns but the tracking speed decreases. As the number of particles increases, the computation time also increases.

Therefore, the population size should be chosen to ensure good tracking speed and accuracy. To find a compromise, five particles are considered in this research.

The FA particles are usually randomly initialized in the search space.

$$d = d_i = (d_1, d_2, d_3, d_4, d_5)$$

The goal of the optimization process is the maximization of the power extracted from the PV system, which is defined as the objective function (P). The fitness value evaluation function is defined as:

$$P(d_i^k) > P(P_i^{k-1})$$

Step 2: Update the best global duty cycle.

The global best duty cycle, d_{best} , is determined by comparing the fitness values of the actual population with the global best PV power obtained, P_{best} .

Step 3: Update of the parameters

At each iteration the attractiveness β and B parameters are updated using the equations (2.4)

Step 4: Update the position of each particle

After the evaluation process, the positions of each particle in the swarm are updated for each iteration.

For each iteration, the duty cycle value that corresponds to a maximum power doesn't change, whereas those which correspond to less power update their values of duty cycle using equation (2.7)

Step 5: Convergence criteria

The algorithm continues to calculate the new duty cycles until the convergence constraint is satisfied. If the absolute difference between each of the two different duty cycles is less than a threshold Δd , then the algorithm stops the optimization process and brings out *d_{best}*:

$$|d_i^k - d_j^k| > \Delta d$$

Step 6: Re-initialization

Due to varying weather conditions and loads, the global MPP is usually changing.

The MPPT algorithm should have the ability to detect the change in the shading pattern and search for the new GMPP. In this brief, the search process is initialized if the following condition is met:

$$\frac{P_{PV_new} - P_{PV_last}}{P_{PV_last}} > \Delta P$$

Where P_{PV_new} and P_{PV_last} are the power values of the photovoltaic panel in two successive sampling periods and ΔP is the power tolerance.

The flow chart below shows how the FA is implemented for the MPPT.

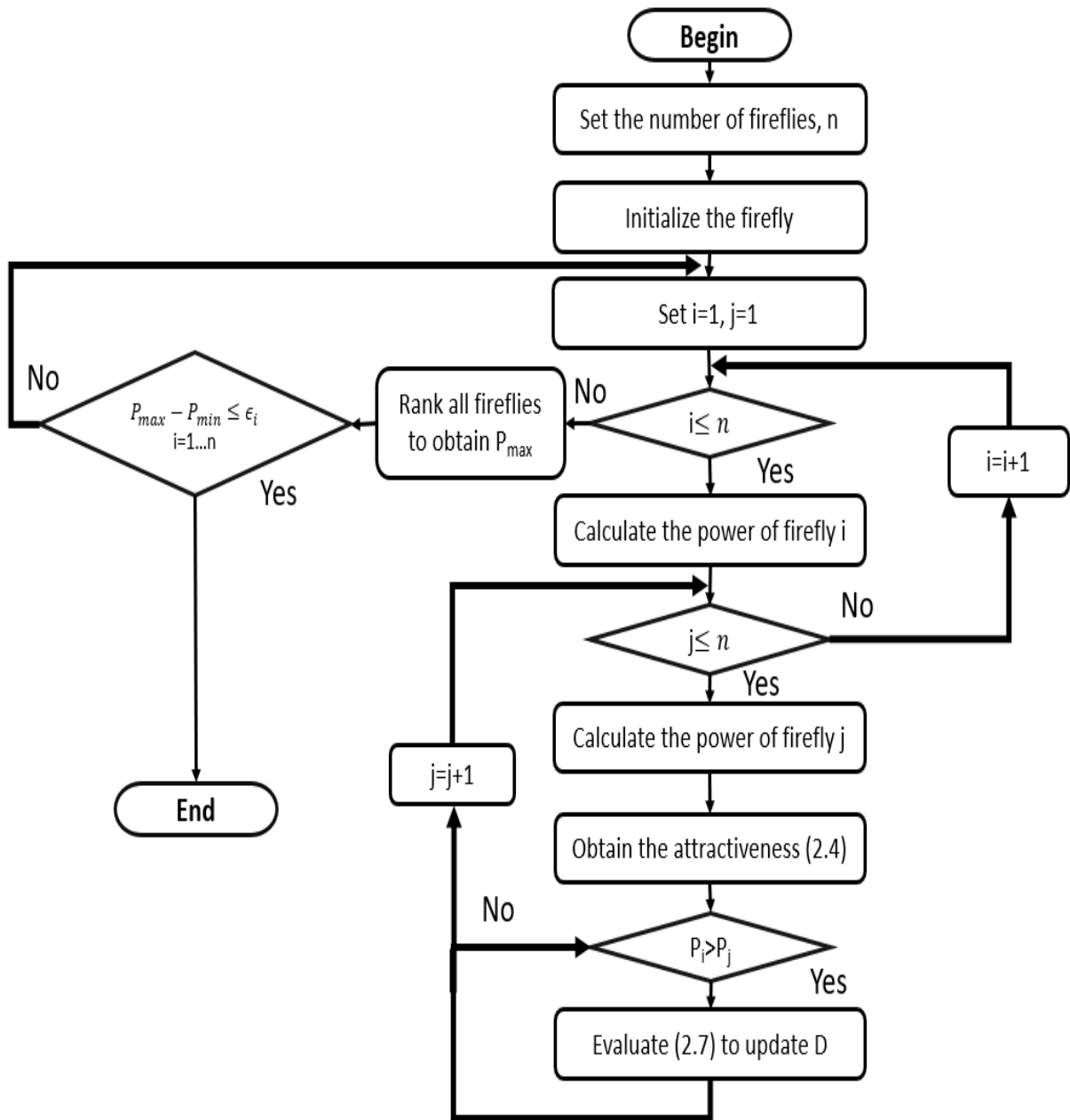


Figure 2-9 Flow chart of the firefly algorithm [2]

2.6 Application of the PSO algorithm for MPPT controller

As mentioned earlier, in order to compare the performance of the chosen controller, in this section we will describe the use of the PSO method to address the optimization problem related to MPPT controller in a PV system.

The flowchart of the PSO algorithm for MPPT control is presented below [2].

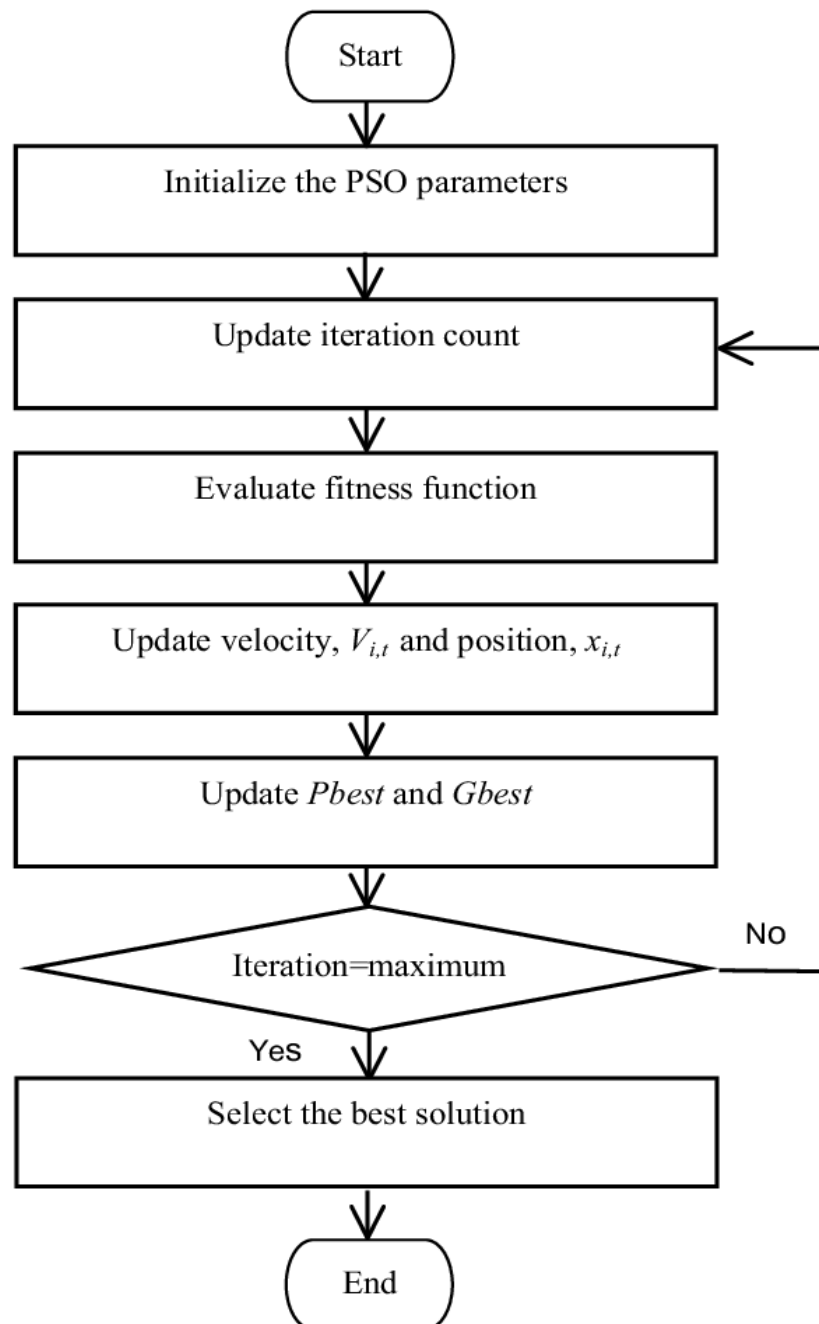


Figure 2-10 Flow chart of the PSO algorithm [2]

2.7 Conclusion

In this chapter we have described the basic principles and design steps of two MPPT controllers. The first one, namely, the proposed controller based on the Firefly Algorithm (FA), as well as the second one based on the PSO algorithm developed in order to be able to compare it with the proposed controller.

In the next chapter, we will present the implementation of these two MPPT controllers. These techniques will be evaluated and compared in terms of convergence time, accuracy and the ability to handle partial shading conditions (i.e., finding the GMPP).

Chapter 3

Software Implementation of FA- MPPT Controller

3. Software Implementation of FA-MPPT Controller

3.1 Introduction

In order to highlight the MPPT controller thus developed from bio-inspired method, namely, FA_MPPT, we will present in what follows the methodology adopted for its simulation. This consists, first, in modelling the PV system, then in a second step and after several tests and simulations, we proceeded to the selection of the FA_MPPT parameters that optimize the controller efficiency.

Following this, taking into account the performance criteria mentioned earlier in Section 2.2.1, we have developed several scenarios to study the evolution and response of the proposed controller under different atmospheric variations and in terms of GMPP tracking under partial shading conditions. In this context, we have chosen to take as reference the criteria related to the convergence time, steady-state oscillations, the accuracy and the robustness against atmospheric variations

Finally, we have carried out a comparative study in terms of performance with another bio-inspired method, which is PSO_MPPT. For this controller, we present only the selected optimal control parameters.

The different test scenarios are developed under MATLAB/Simulink environment.

3.2 Simulation Methodology of the proposed MPPT controller

In order to perform a comparative study between the two bio-inspired based controllers, we adopted the methodology cited above.

The methodology consists, first of all, in modelling the PV system, which will be followed by the parameterization step of the suggested algorithm, then in inserting the two MPPT controllers, one by one, in the PV system modelled under MATLAB/Simulink during each simulation phase, taking into account the selected criteria and the different scenarios considered.

3.2.1 Modelling of the photovoltaic system

Figure 3.1 shows the block diagram developed for the simulation of the PV system. The overall simulation model of the PV system is a combination of different blocks modelling the elements that constitute it.

The PV system is modelled under MATLAB/Simulink environment. It is composed of the following blocks: a PV generator, a static Buck-Boost converter, a resistive load block and a MPPT controller block.

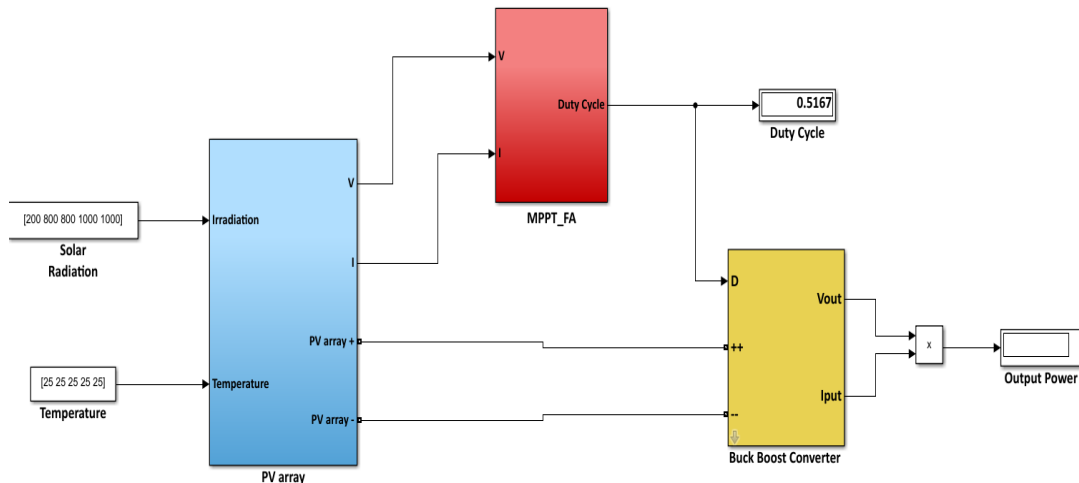


Figure 3-1 Block diagram for PV system simulation

3.2.1.1 PV array

The PV array is composed of five modules (Modules #1–#5), with their parameters being given in Table 3. Each module is a SunPower SPR-X20-250-BLK which consists of 72 cells connected in series and protected by a bypass diode mounted in antiparallel.

Table 3 GPV characteristics

Maximum power	$P_{max} = 1250 (W)$
Voltage at maximum power	$V_{MPP} = 214 (V)$
Current at maximum power	$I_{MPP} = 5.84 (A)$
Open-circuit voltage	$V_{OC} = 254.65 (V)$
Short-circuit current	$I_{sc} = 6.2 (A)$

3.2.1.2 MPPT controller block

Two different MPPT algorithms, the proposed FA and PSO were implemented in the MPPT block to compare and evaluate tracking accuracy and speed. The operating voltage and current of the PV array were measured as input data of the MPPT algorithms.

3.2.1.3 Buck-Boost converter

The buck-boost topology is used for several reasons, notably because it has superior characteristics, compared to other types of converters, regarding the MPP tracking performance of the photovoltaic generator, and it also allows the MPP tracking at any time, regardless of the PV panel temperature, solar irradiance and connected load.

The DC-DC converter was controlled by a PWM signal of a duty cycle (D) adjusted by the MPPT algorithms to change the operating point of the PV array.

The switching frequency is chosen equal to 10kHz. The determination of the different components constituting the buck-boost circuit is a critical step in the implementation. The choice of components is made considering that the converter is in continuous conduction mode.

$$L \geq R * \frac{1 - D^2}{2 * f} \rightarrow L \geq 0.0005H$$

$$C_{in} \geq \frac{I_{pv_max}}{f * \Delta V_{pv}} \rightarrow C_{in} \geq 0.0001F$$

$$C_o \geq \frac{D_{max}}{ripple * f * R_o} \rightarrow C_o \geq 0.0001F$$

The values of the capacitors, inductance and output resistance used are:

$$C_{in} = 476\mu F, C_o = 116\mu F, L = 903\mu H, R_o = 17.2\Omega$$

3.2.2 Parameterization of FA and PSO

To get the best results from any optimization algorithm, it is necessary to select the appropriate parameters, as these parameters are crucial and can have a big impact on the efficiency of the algorithm (i.e., slow convergence, stuck in a local maximum, etc.).

For example, if the size of the population (fireflies) is large, this will extend the search space for each iteration. However, this will increase the computation time.

Moreover, the number of iterations also has a great influence, as an example: if the number of iterations is small, it can block the search process prematurely, while for a high number of iterations, it can increase the computing time.

To do this, taking into account these factors and in order to evaluate the influence of the parameters during the execution of the algorithm FA and to validate the proposed controller FA_MPPT, several tests are performed.

These consist in evaluating the controllers by changing the values of the respective parameters, namely: (number of fireflies, the maximum of attractiveness, absorption coefficient, etc.), thus making it possible to select the most appropriate ones so as to reduce the execution time and to improve the effectiveness of the controllers, i.e., to reach the MPPT.

While for the PSO_MPPT controller, we present only the selected optimal control parameters.

The different tests are performed under standard conditions (1000W/m², 25°C).

3.2.2.1 FA parameterization

Figure 3.2 shows the simulation results for different values of the FA controller parameters.

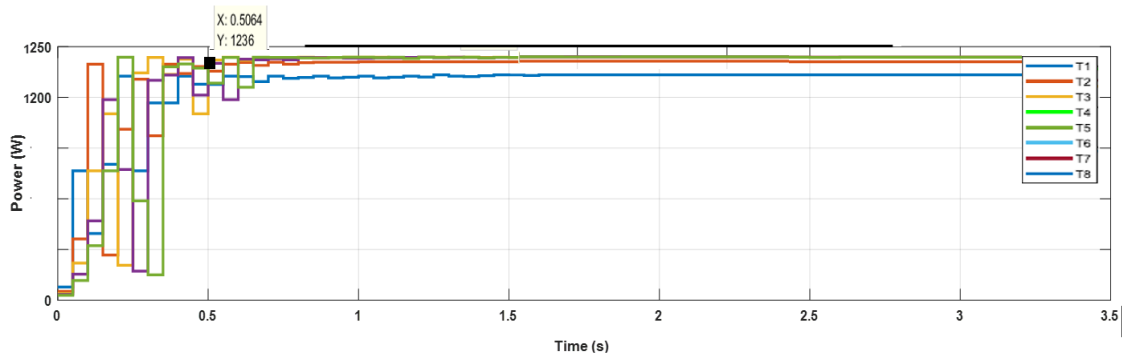


Figure 3-2 Simulation for different values of the controller parameters. FA_MPPT.

The table below shows the tests conducted for different settings of the parameters:

N : the number of fireflies;

β_0 : the maximum attractiveness;

γ : the absorption coefficient of light.

P_{pv} : Power of the PV module;

T_c : convergence time;

Table 4 Simulation results for different parameter values.

Tests Parameters	T1	T2	T3	T4	T5	T6	T7	T8
Max_iter	40	20	10	20	20	20	20	20
N	6	6	6	6	5	5	4	5
β_0	1	0.7	0.5	0.3	0.5	0.5	0.5	0.5
γ	1	1	1	1	1	0.9	1	0.8
$Ppv(W)$	1243.3	1243.6	1201.9	1232.2	1243.9	1243.7	1236.1	1240.5
$T_c(s)$	2	0.6	0.6	0.5	0.5	0.6	0.6	0.6

According to the various tests carried out, the use of 20 iterations and 5 fireflies is sufficient to ensure the convergence of the controller; these values are the minimum values that can be used. Indeed, for values lower than the number of iterations “Max_iter: the maximum number of iterations”, and a lower number of fireflies “N: number of fireflies”, the method can converge towards a local maximum and not reach the optimal solution as shown by the first tests in table.4.

The performance of the FA algorithm is tested under different partial shading scenarios. The parameters used are Max_iter=20, N=5, $\beta_0=0.5$ and $\gamma=1$. These parameters were selected after a series of simulations. Therefore, it can be confirmed that the combination (Max_iter ,N , β_0,γ) is optimal.

3.2.2.2 Parameterization of the PSO algorithm

The performance of the PSO algorithm is also tested under different partial shading scenarios. The parameters used for the PSO algorithm are $W=0.4$, $C1=1.2$, $C2=1.6$, $\Delta P=0.5$ and $\Delta d=0.01$. These parameters were selected after a series of simulations. Therefore, it can be confirmed that the combination (W , $C1$, $C2$) is the most optimal.

3.2.3 Simulation of the FA controller

In order to visualize the position of the maximum power point from the proposed controller, different scenarios have been developed to follow the evolution and the behaviour of this controller in regards to different atmospheric variations during the different simulations.

The behavioural study of the different controllers will be done on the basis of the previously selected criteria, namely:

- Convergence (response time,);
- Steady-state oscillations;
- Accuracy;
- Robustness.

In order to highlight each of these criteria, we have adopted the following scenarios:

- Scenario 1 consists in simulating the performance of the PV system under standard conditions ($25\text{ }^{\circ}\text{C}$, $1000\text{ W}/\text{m}^2$) and visualizing the response time of our controller, thus allowing to evaluate the criteria related to the convergence speed;

- Scenario 2 consists in simulating the performance of the PV system by modifying the atmospheric conditions (Irradiance or Temperature) in order to observe the dynamic behaviour of the FA_MPPT controller, thus allowing to study the criteria related to stability, oscillation, precision and robustness. The controllers are subjected to the following scenarios:

- Simulations of the PV system under varying solar irradiation conditions where two cases are considered: decrease and increase of the irradiance;
- Simulation of the PV system under varying temperature conditions.

In the following, we will apply each of these scenarios for the simulation of the FA_MPPT controller. Then, in order to validate the high performance of the proposed controller, a comparative study with PSO_MPPT controller in terms of convergence, steady-state oscillations, accuracy and robustness is established.

3.2.3.1 Study of the convergence criteria

As mentioned previously, in order to observe the response time of the developed FA_MPPT controller, scenario 1 is adopted. It consists in simulating the different controllers under standard conditions ($Irradiation = 1000 W/m^2$, $Temperature = 25\text{ }^\circ C$) and observing the evolution of the output power supplied from the controller to the load as a function of time.

Figure.3.3 illustrates the evolution of the output power supplied to the load for the FA_MPPT controller.

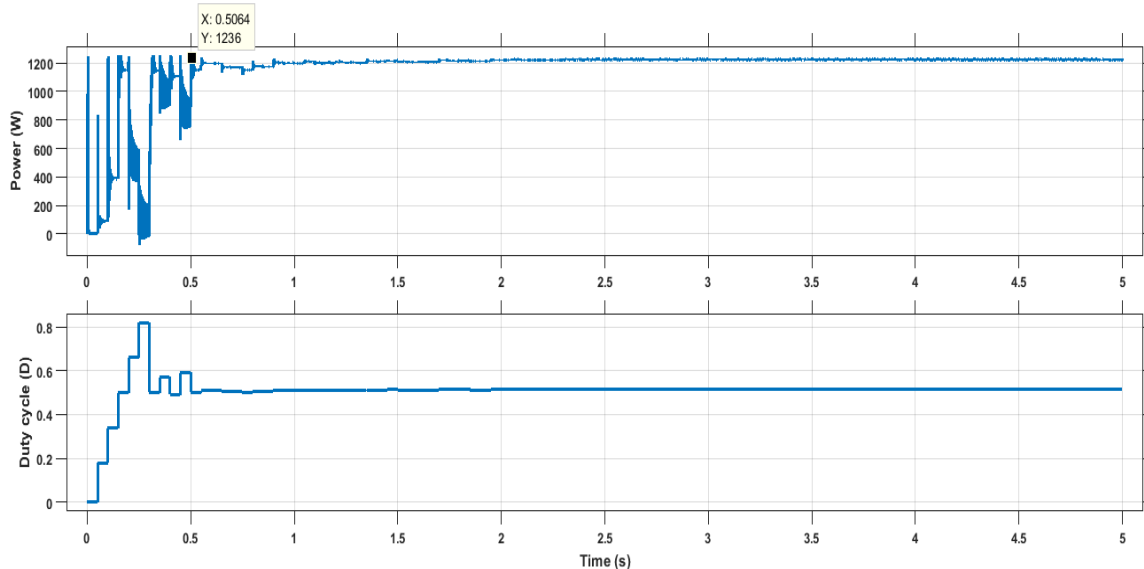


Figure 3-3 Evolution of the output power and the duty cycle of the FA_ MPPT under standard conditions ($Irr=1000 W/m^2, T=25\text{ }^\circ C$)

Figure.3.4 shows the time response of the FA_MPPT and PSO_MPPT controllers under standard environmental conditions as a comparative study of their performances.

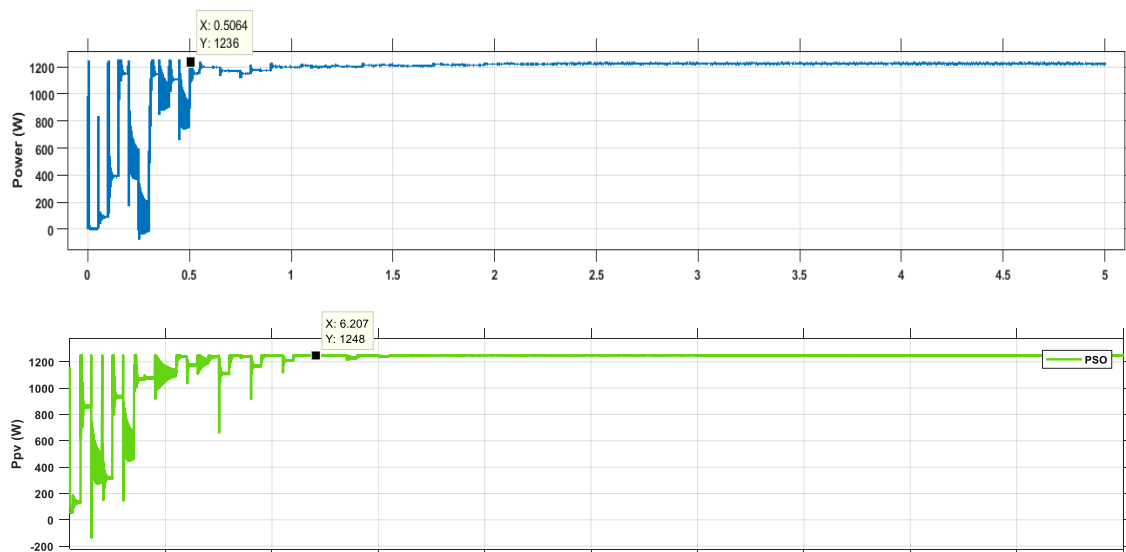


Figure 3-4 Evolution of the output power of FA and PSO MPPT under standard conditions.

Discussion of results:

The proposed controller FA_MPPT is much faster than the PSO_MPPT controller with a convergence time of 0.56 s for FA_MPPT and 1.2 s for PSO_MPPT. The two bio-inspired methods provide almost zero steady-state oscillation.

3.2.3.2 Study of stability, accuracy and robustness criteria

The efficiency of MPPT controllers is evaluated with respect to their ability to track maximum power, their robustness, their stability against oscillations and their accuracy.

To estimate the accuracy, stability, robustness and efficiency of the developed controller, several scenarios were conducted to observe the dynamic response of the controller. The PV system equipped with the controller is tested for different variations of environmental conditions (temperature or irradiation). We present in the following the results obtained.

a. Rapid variation of the irradiation for a fixed temperature

In this test, a rapid increase in illuminance from 600 to 1000 W/m² is observed at time $t=5s$. Then a decrease in illuminance occurs at time $t=10s$ from 1000 to 400 W/m². During the whole test, the temperature is maintained at 25°C.

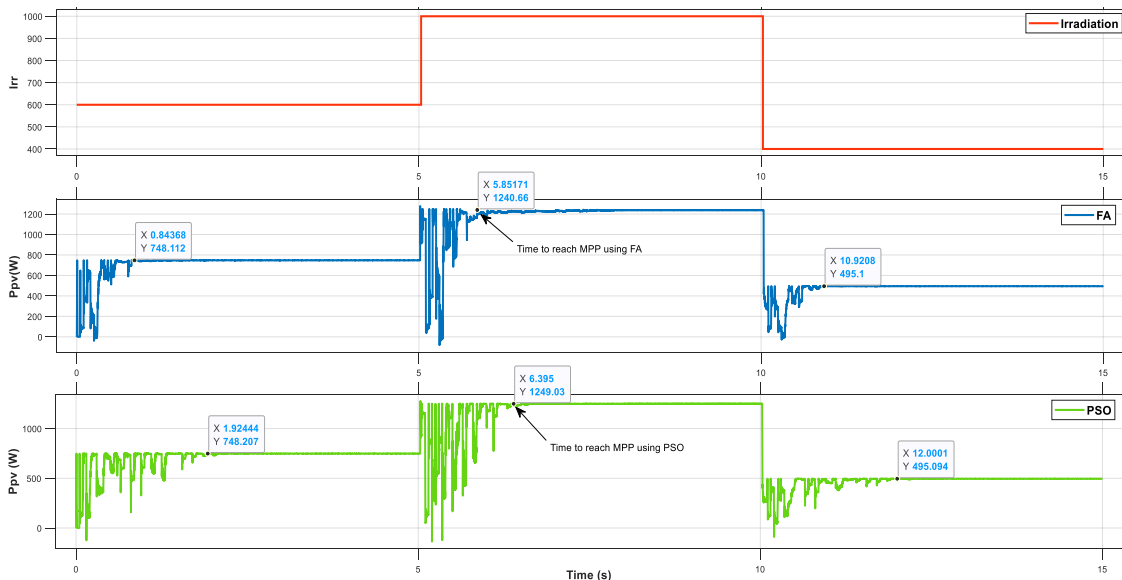


Figure 3-5 power evolution of the FA and PSO MPPT controllers for rapid changes in irradiation.

Figure 3.5 shows the power output of the PV module when the irradiation conditions change rapidly for the two controllers. The results highlight that the tracking efficiency under all irradiation conditions was 100% with the FA_MPPT controller and slightly above 99% with the PSO_MPPT controller.

To notice that the FA_MPPT presents also better performance in terms of convergence time, the MPP is reached after 0.85s-0.92s from the beginning of the sudden variations at t=5s and t=10s and 1.39-2s for the PSO_MPPT controller.

But even though, we can say that the two controllers based on bio-inspired algorithms follow perfectly the sudden variations of the irradiation and show no oscillation around the MPP.

b. Rapid temperature change at fixed irradiation

In this test, a sudden decrease of the temperature from 45°C to 25°C is observed at the time t=5s. Then another decrease of temperature occurs at time t=10s from 25°C to 10°C. During the whole test the irradiation is maintained at 1000 W/m².

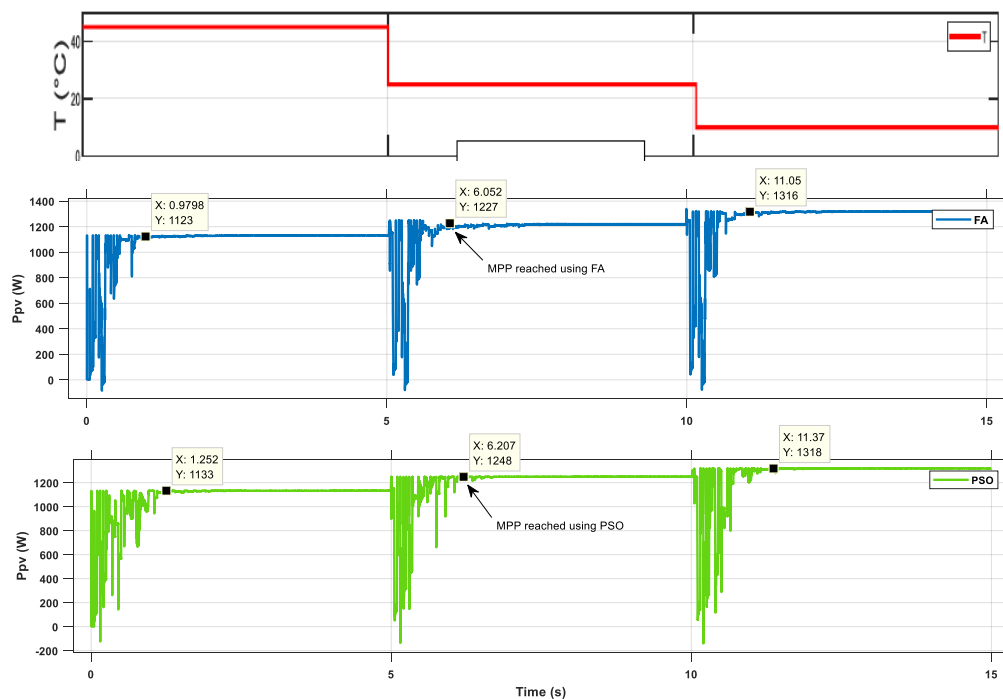


Figure 3-6 power evolution of the FA and PSO MPPT controllers for rapid changes in temperature.

Figure 3.6 shows the power output of the PV module when temperature conditions change rapidly for both controllers. The results highlight that the tracking efficiency under all temperature conditions was 100% with the FA_MPPT controller and slightly above 99% with the PSO_MPPT controller.

To notice that the FA_MPPT also shows better performance in terms of convergence time, the PPM is reached after 0.85s-0.92s from the beginning of sudden variations at t=5s and t=10s and 1.39-2s for the PSO_MPPT controller.

Despite this, we can say that both controllers based on bio-inspired algorithms react perfectly to the sudden temperature variations and show no oscillation around the MPP.

3.2.3.3 Effect of partial shading

The purpose of this scenario is to observe the performance of the proposed method FA_MPPT against the effect of partial shading on the PV system and to evaluate the performance of the latter.

Since partial shading is a stochastic phenomenon, several scenarios can occur. Therefore, it is difficult to test all occurrences of partial shading.

Therefore, we opted for the design of a model under Simulink to simulate the behaviour of the PV system. The latter is composed of five PV modules. The temperature of the modules (T1–T5) was fixed at 25 °C. The irradiation setups (G1–G5) for each module could be individually programmed to generate different PSCs. Table 2 lists the 4 different PSCs (Patterns #1–#4), the purpose of which was to simulate patterns that have 1–4 LMPPs, with their respective GMPP located at different intervals.

Table 5 Irradiation setup of the four shading patterns (W/m²)

Irradiance setup	LMPPs	Module #1	Module #2	Module #3	Module #4	Module #5
Pattern #1	2	1000	1000	1000	500	500
Pattern #2	3	1000	600	600	200	200
Pattern #3	4	1000	800	600	400	400
Pattern #4	5	1000	200	800	600	600

The figure below illustrates the P-V curves of the PV array under the 4 different patterns and their corresponding GMPPs (#1–#4).

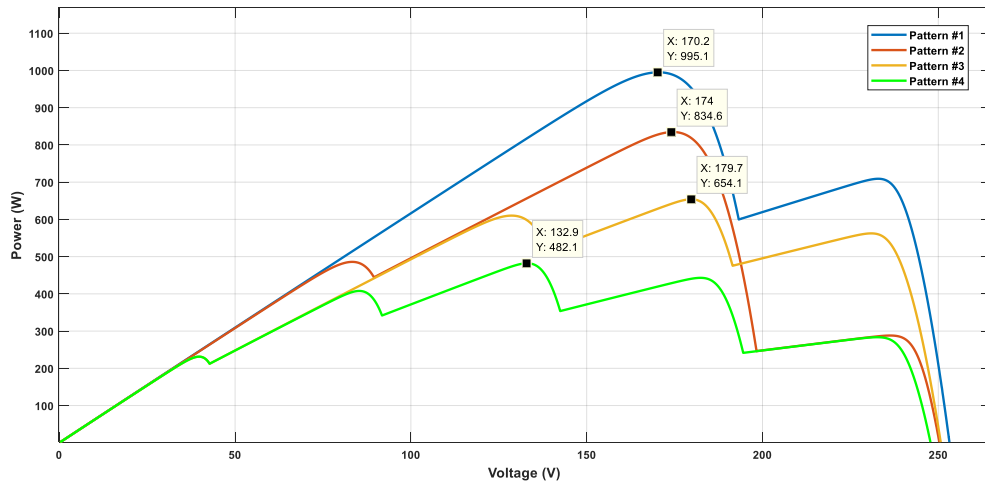
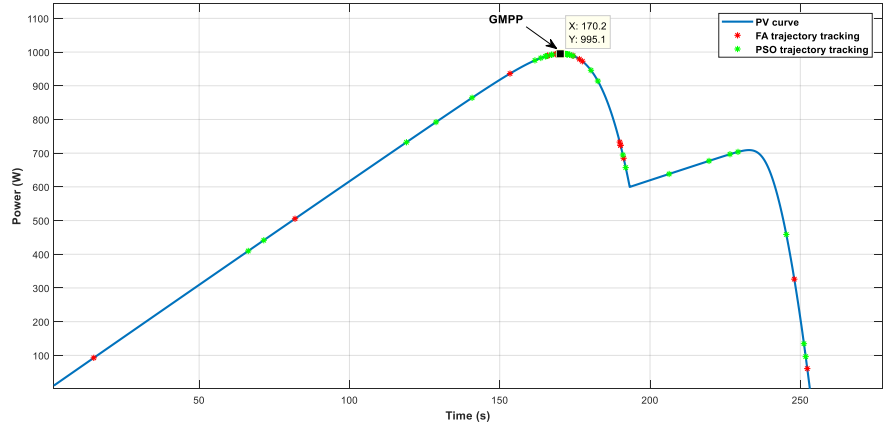
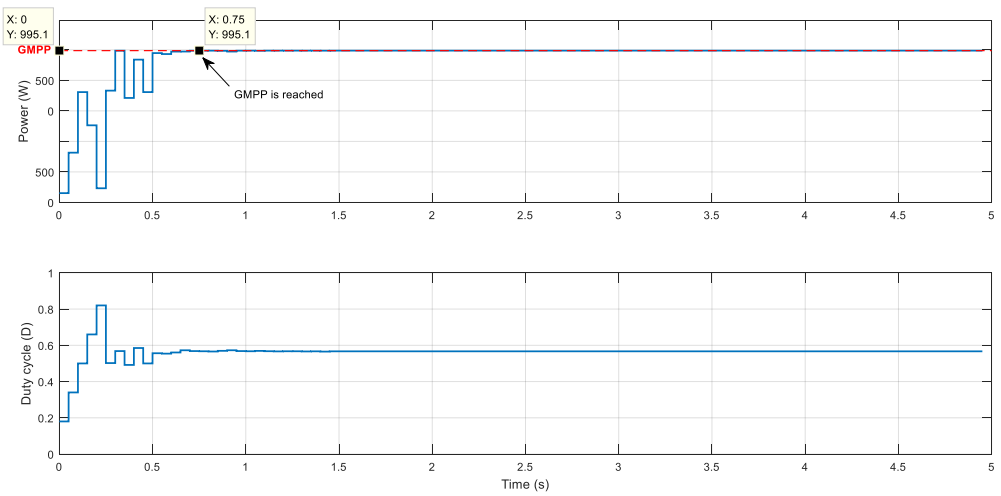


Figure 3-7 P-V characteristic for different shading patterns.

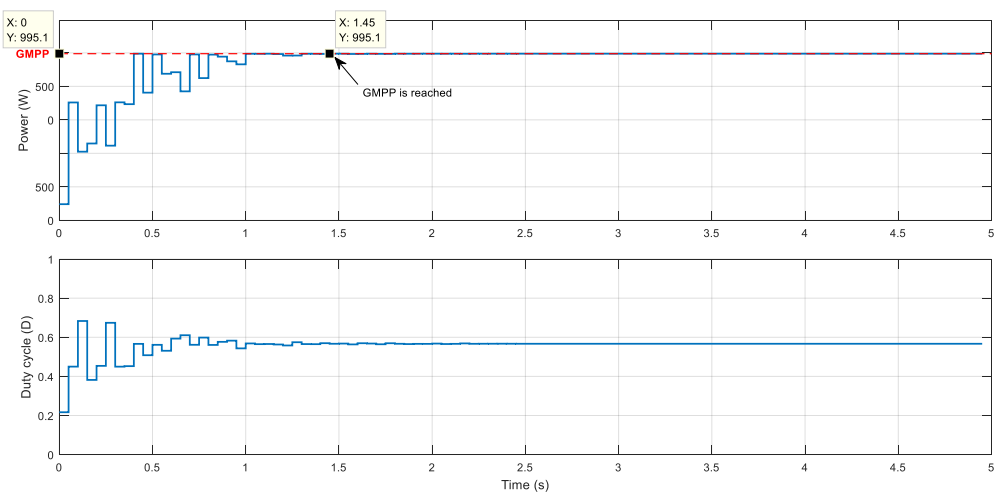
The four figures from 3.8 to 3.11 illustrate the power simulation results extracted from the proposed FA_MPPT and PSO MPPT bio-inspired controllers.



(a)

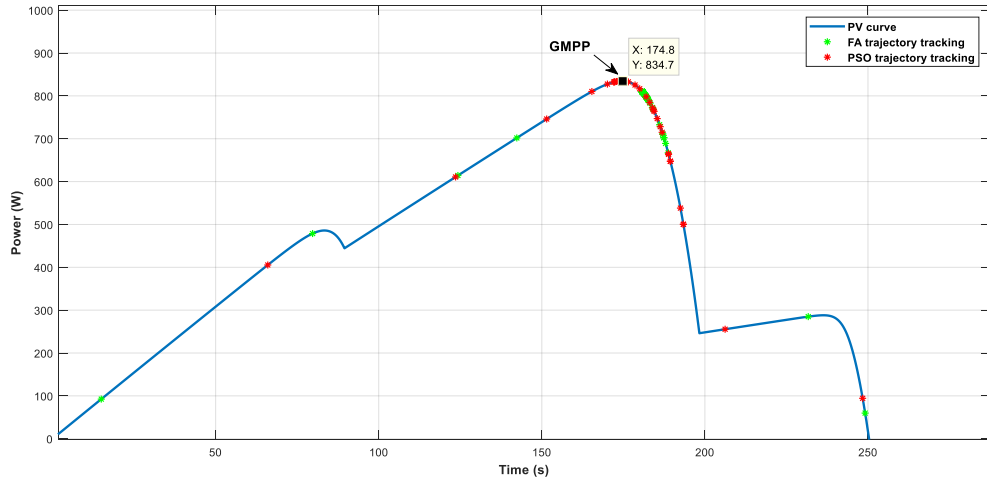


(b)

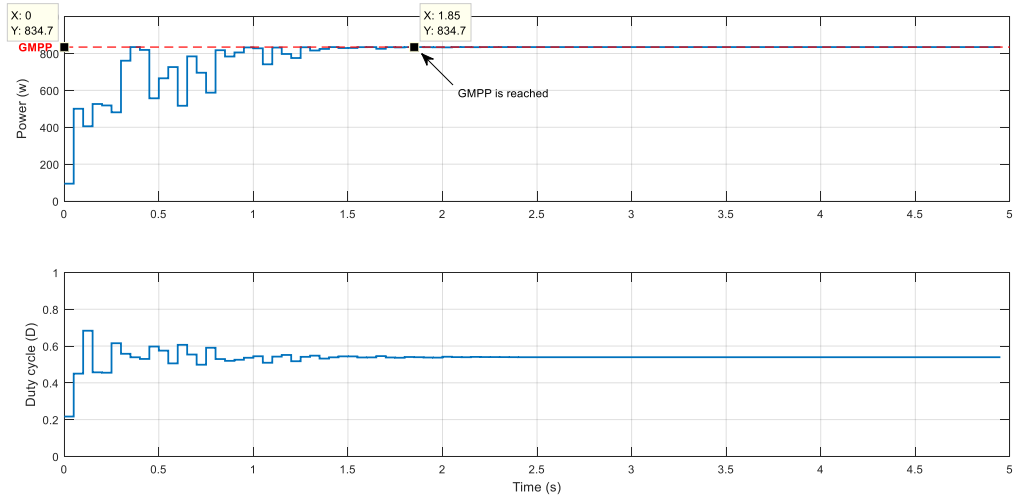


(c)

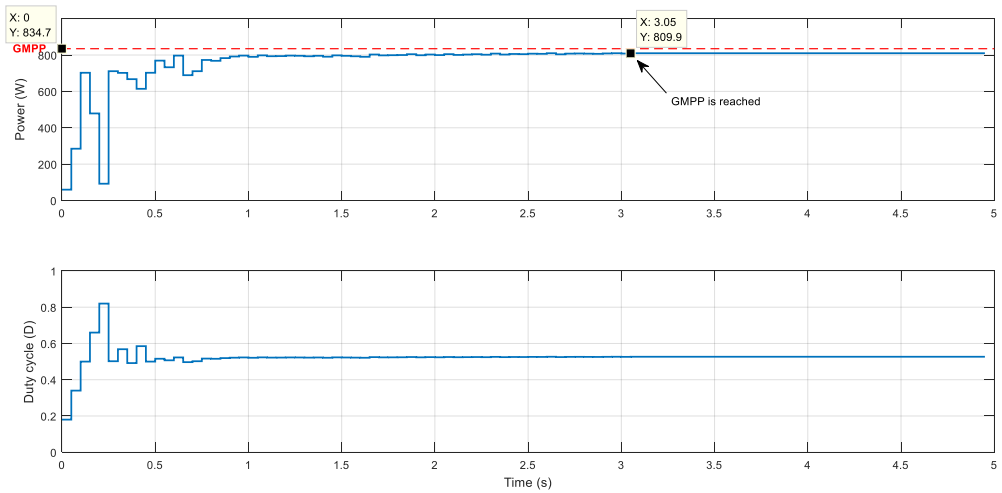
Figure 3-8 The extracted power during the shading pattern #1 with, (a) P-V Trajectory tracking, (b) FA, (c) PSO.



(a)

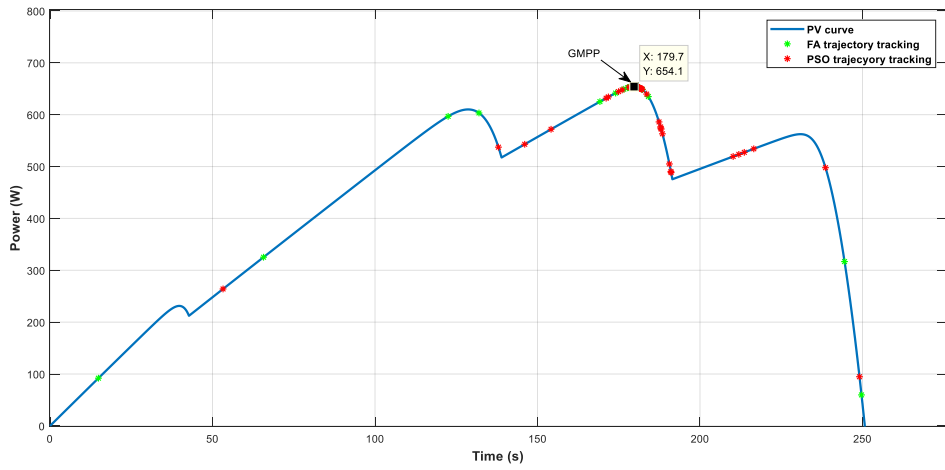


(b)

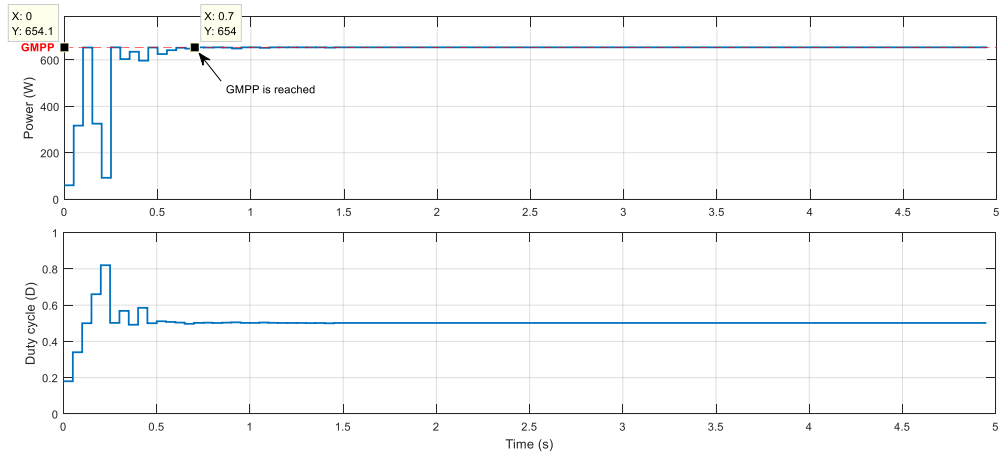


(c)

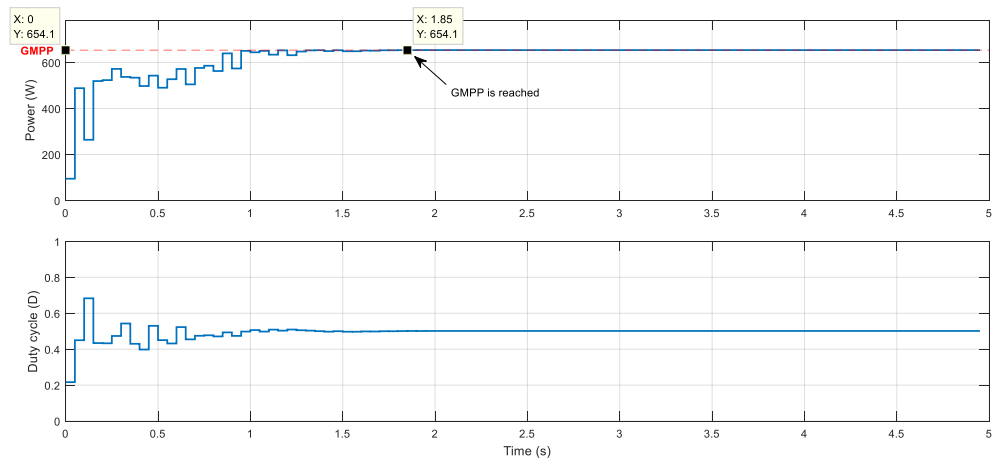
Figure 3-9 The extracted power during the shading pattern #2 with, (a) P-V Trajectory tracking, (b) FA, (c) PSO.



(a)

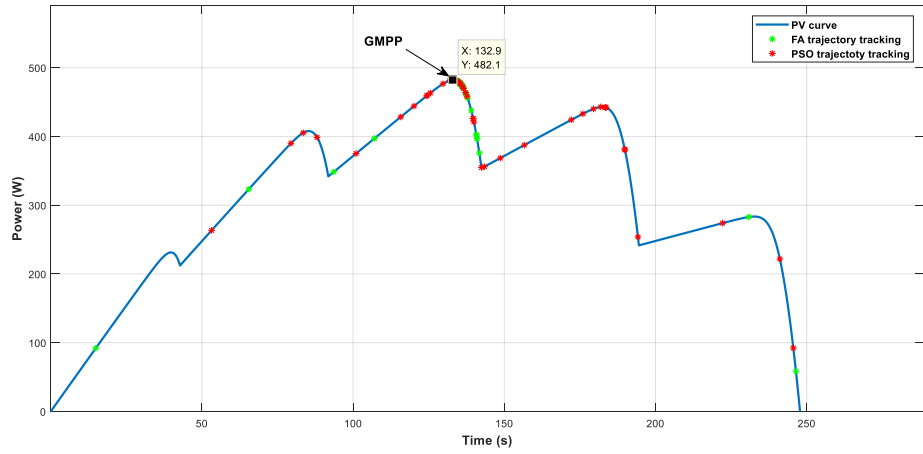


(b)

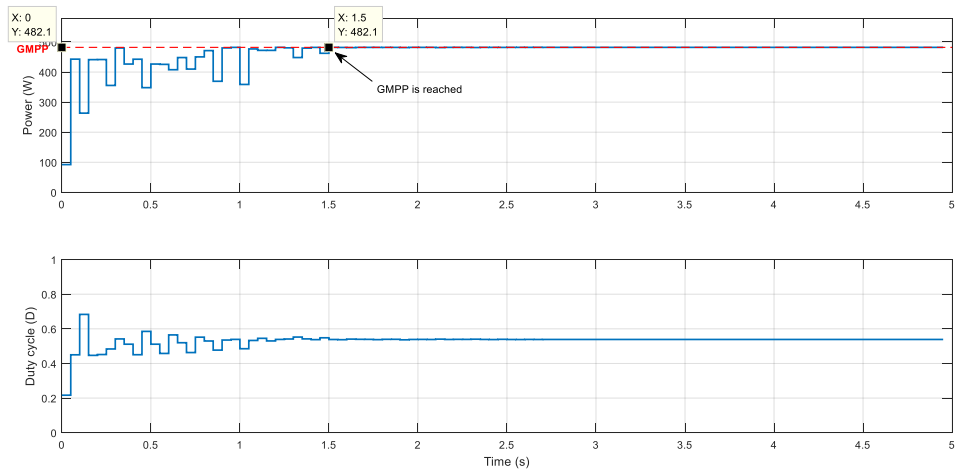


(c)

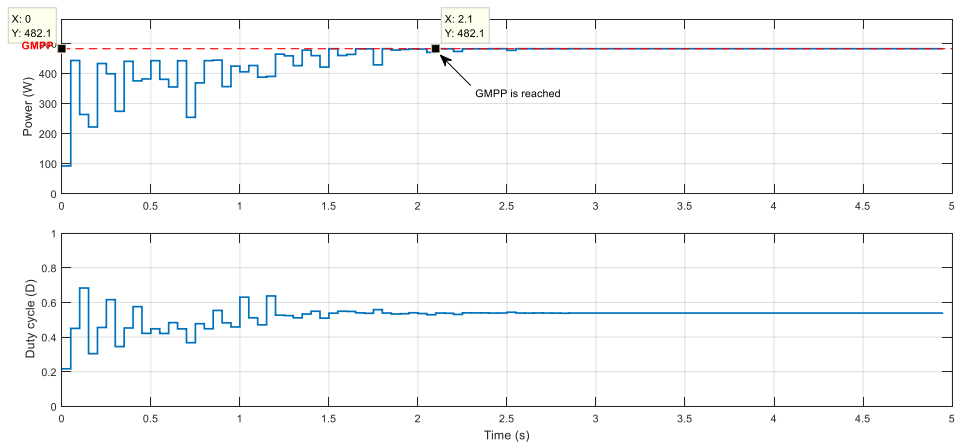
Figure 3-10 The extracted power during the shading pattern #3 with, (a) P-V Trajectory tracking, (b) FA, (c) PSO.



(a)



(b)



(c)

Figure 3-11 The extracted power during the shading pattern #4 with, (a) P-V Trajectory tracking, (b) FA, (c) PSO.

Discussion of results:

From the obtained simulation results, we can conclude that when the shading phenomenon occurs, the FA and PSO MPPT controllers show good performances to locate the GMPP under different partial shading conditions. For each shading pattern, the global maximum is reached as shown in the four figures corresponding to the four partial shading patterns.

To prove once more that the proposed FA_MPPT controller overcomes the PSO_MPPT controller in tracking the global power point in some partial shading scenarios. In what follows, we will compare FA_MPPT with PSO_MPPT in terms of convergence time, tracking accuracy and efficiency.

Table 3.6 shows the steady state tracking results of the two algorithms.

- Tracking efficiency and accuracy:

The FA_MPPT controller produces a tracking efficiency of 100% in all the partial shading cases studied. From this we conclude the superiority of our controller in this study.

- Convergence time:

For all scenarios, the FA algorithm achieved the GMPP tracking faster than the PSO algorithm. For the first and third shading scenario, FA method has well located the GMPP in only 6 to 7 iterations, the PSO algorithm on the other hand performs 8 to 12 iterations to reach the steady state. For the second and fourth scenario. FA takes around 3 iterations to reach the GMPP, while PSO takes 6 to 7 iterations to meet the convergence criterion. As can be seen, the FA algorithm is superior in terms of convergence time compared to the PSO technique.

Table 6 Simulation results of the tracking speed, accuracy and efficiency

Patterns	Pmax available (W)	Algorithm	P_found (W)	Convergence time (s)	Sampling times	Iterations	Accuracy error (%)	Efficiency (%)
Pattern #1	995.1029	FA	995.1022	0.75	15	3	7.03×10^{-5}	100
		PSO	995.1017	1.45	29	5.8	1.206×10^{-4}	100
Pattern #2	834.7154	FA	834.7134	1.85	37	7.4	2.396×10^{-4}	100
		PSO	809.8934	3.05	61	12.2	2.974	97
Pattern #3	654.0535	FA	654.0534	0.7	14	2.8	1.529×10^{-5}	100
		PSO	654.0517	1.85	37	7.4	2.752×10^{-4}	100
Pattern #4	482.134	FA	482.132	1.5	30	6	4.148×10^{-4}	100
		PSO	482.012	2.1	42	8.4	0.0253	99

3.3 Conclusion

The simulations described in this chapter are purely theoretical because in order to evaluate the performance of a digital control before establishing a real prototype, it is necessary to perform simulations first.

We have analysed the performance of two MPPT controllers, based on FA and PSO algorithms, for finding the GMPP in a photovoltaic system. The MATLAB / Simulink environment is used to simulate various I-V and P-V characteristic curves under various partial shading scenarios. A robustness and performance analysis of the proposed MPPTs has been performed. A comparative study between the MPPT based on the proposed FA algorithm and based on PSO algorithm has been performed. From the obtained simulation results, the proposed controller is able to reach the optimal MPPTs with high accuracy and low convergence time under different partial shading conditions. The MPPT controller based on the FA algorithm produces 100% tracking efficiency in all the studied scenarios while the MPPT based on the PSO algorithm failed in dealing with some partial shading scenarios, so provides less tracking efficiency.

In the next chapter, we will propose a new conception for the rapid prototyping of MPPT controllers, using Model Based Design approach in order to implement it onto an FPGA with the HDL Coder.

Chapter 4

Hardware Implementation of FA-MPPT Controller

4. Hardware Implementation of FA-MPPT Controller

4.1 Introduction

With the technological advancement in the field of microelectronics, new digital solutions such as FPGAs (Field Programmable Gate Array) or ASICs (Application Specific Integrated Circuit) are available and can be used as digital targets for the implementation of control algorithms.

The inherent parallelism of these new solutions as well as their great computing capacities make that the delays of calculation are reduced in spite of the complexity of the algorithms to be implemented. Indeed, in addition to the improvement of the performances through the reduction of the calculation times, the parallelism of the material solutions makes it possible to integrate, on a single circuit, several algorithms which ensure various functionalities and which can work independently of each other.

In view of the proven advantages of FPGA circuits, they have over time become essential in the design of digital systems in the many fields where these systems are used. The field related to renewable energies is not left out. Indeed, the use of MPPT controllers for power optimization in PV systems has become an essential technological solution.

It is with this in mind, and as part of our work, that we have targeted FPGA circuits for the hardware implementation of the bio-inspired firefly algorithm.

In this chapter, we will discuss, first, the Field Programmable Gate Array (FPGA) circuits, with their main characteristics and their architecture.

Then in a second step, we will examine the contribution of Model Based Design MBD methods for MPPT controller implementation, where we will target mainly the Hardware Description HDL Coder method with Vivado as a third-party software.

4.2 Hardware Design Methodologies

4.2.1 FPGA

The Field Programmable Gate Arrays (FPGAs) have been developed since 1985. Thanks to the technological evolution these circuits have not stopped developing and many manufacturers have since embarked on the manufacture of FPGAs (Xilinx, Altera, Actel...). Taking advantage of technological change, an FPGA can now contain several million logical gates. These circuits are composed of logic elements interconnected with one another by routing resources. The special feature of an FPGA is its reconfigurability. Logical elements as well as the interconnect network can be reconfigured to support another application. There are several types of FPGAs (SRAM-based FPGAs, Flash-based FPGAs, and Antifuse-based FPGAs), classified according to how they are programmed. The majority of FPGAs use static memory and are called SRAM-based FPGA [32].

4.2.2 FPGA Architecture

An FPGA consists of an array of logical elements called Configurable Logic Block (CLB) connected to each other by interconnection resources.

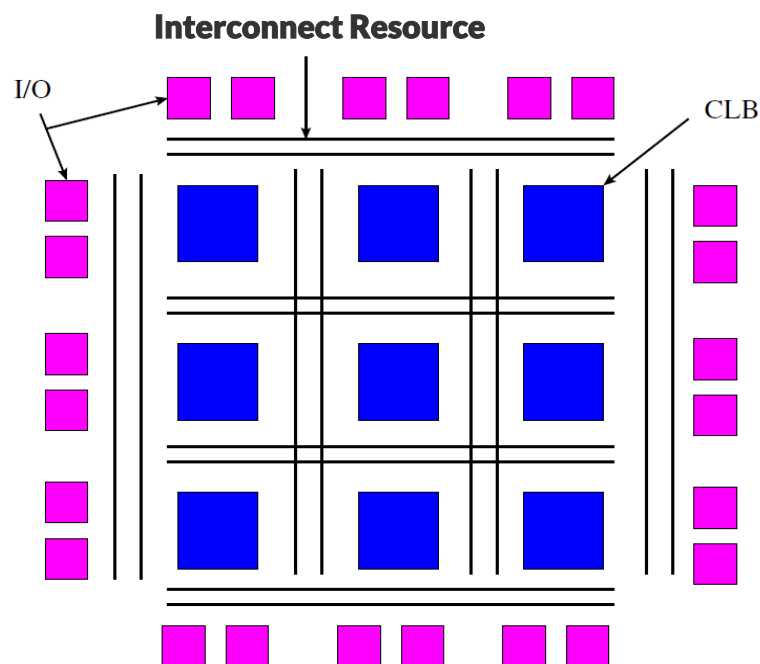


Figure 4-1 Architecture of an FPGA [32]

Each CLB consists of the Basic Logic Element (BLEs) used to implement the logical part of the circuit. A CLB is therefore characterized by the number of its I input and by the number N of the BLE it contains. In modern FPGAs this number of BLE can vary from 3 to 12 and each BLE can be connected to any I input of the CLB or to another BLE. An BLE consists of a set of transcoding tables (Look Up Table: (LUT) and a D-toggle to implement the basic functions with the memory block (SRAM), followed by a multiplexer.

4.2.3 Evolution of co-design methodologies

Generally, design methodologies are based on very structured approaches. They are based on the validation of representations by simulation, at different levels. They use component libraries in a CAD environment.

4.2.3.1 Bottom-Up methodology

As its name indicates, this methodology starts from the low level of abstraction to the high level. The methodology consists in consolidating and synthesizing the basic elements initially identified. In other words, it favors the design and validation of elementary modules before those of the higher hierarchical level. Thus, each level generates libraries for the higher level of abstraction. The advantage of this methodology lies in the fact that the levels of abstraction are clearly separated and each has its own library.

4.2.3.2 Top-Down methodology

Top-Down methodology starts with the highest level of abstraction to go from the functional description of the system to its structure. Thus, the system is built as a hierarchy of objects, where the details of realization become clearer as one goes down in this hierarchy.

Traditionally, the design and implementation of an "algorithm" architecture on an FPGA circuit uses top-down design methodologies (approaches) known as classical, very close to the design of dedicated digital circuits ASIC (Applied Specific Integrated Circuit).

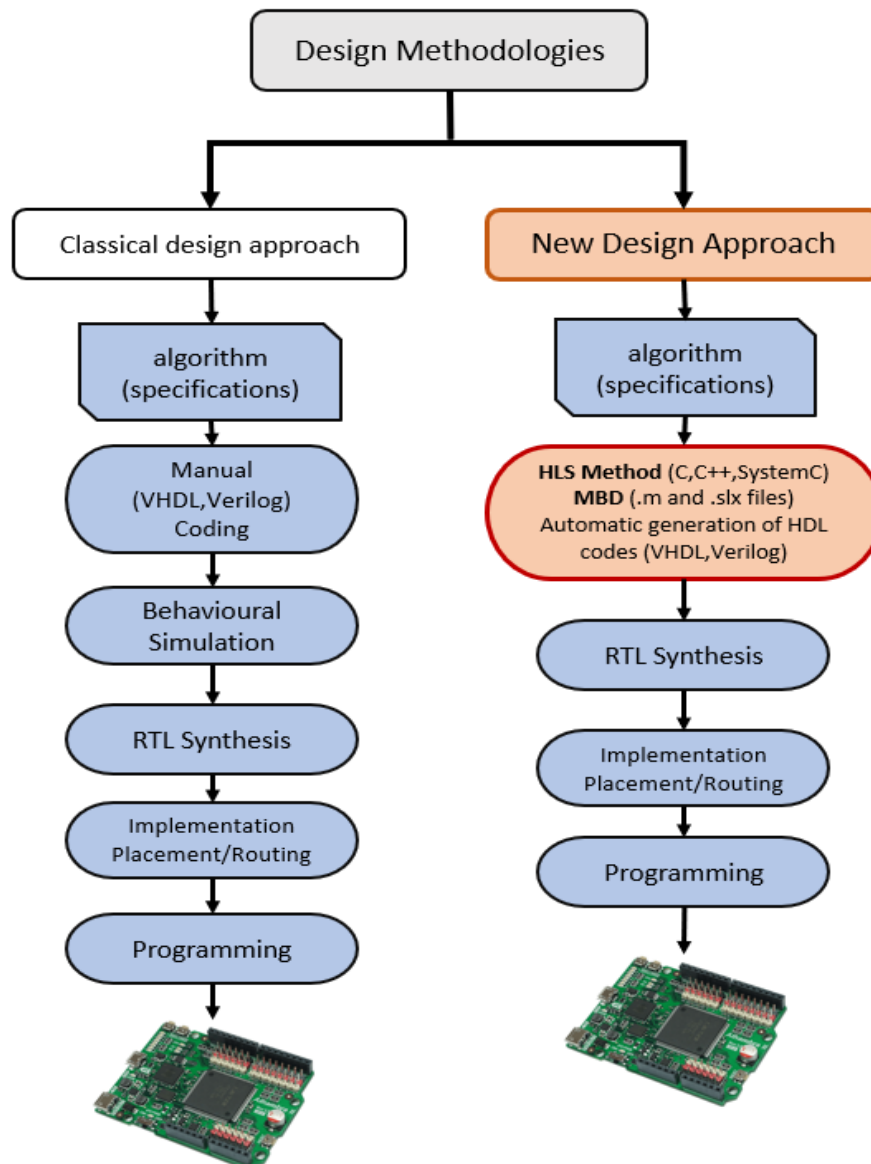


Figure 4-2 design approaches

As depicted in figure 4.3, in the classical approach, the flow allows the passage from hand writing HDL language (vhdl Verilog.) description to the RTL level until arriving at the generation of the "bitstream" file for programming the circuit.

It is useful to specify that during the design phase of the algorithm, the choice of the type of description as well as the set of parameters of the synthesis tool strongly affect the performance of the application in the end.

The tool generates the synthesis that translates the transcription of the HDL description introduced by the designer into a representation at RTL (Register Transfer Level). This is done by first generating the basic elements such as flip-flops, LUTs and state machines. Then, from the source code, it extracts the macros corresponding to specific behaviours

or operations such as those related to RAM, shift registers, multiplexers, etc., and replaces them with the manufacturer's own components.

After that, the tool performs a regrouping of the combinatorial and sequential resources to minimize the number of logic blocks used. As soon as the synthesis process is finalized, a functional simulation of the algorithm thus described is already possible to verify the behaviour of the system with respect to the expected functionality. If the specifications are not met, it is possible to make the necessary changes by modifying the code to remedy the problem.

From the synthesis files, the tool first transforms the RTL representation into a "logic gate" level representation, then proceeds to the placement and routing on the targeted FPGA circuit.

Generating the bitstream file (.bit) is the last step that allows the generation, from the obtained representation, of a "bitstream" configuration file that will be used to program the internal connections and those of the logic blocks of the FPGA circuit.

However, the classical design methodology, which goes from the algorithmic specifications to the generation and implementation of the bitstream file (.bit), has seen the emergence of new design approaches.

These new design approaches were created to shorten the design time and to overcome the increasing difficulty of programming complex architectures with HDL (Hardware Description Language).

As depicted in Figure 4.4, the new design approaches consist of:

- The C-like language-based approach, which consists of describing the algorithm with programming languages of higher abstraction level than HDLs, such as C, C++, SystemC, SpecC, Handel-C, etc...
- The Model Based Design (MBD) approach consists of a set of tools with graphical interfaces that allow describing the behavior and structure of a digital system with graphical models instantiated from predefined libraries. In this category we find MATLAB Simulink HDL Coder, Xilinx System Generator, LabView, Altera DSP Builder, etc

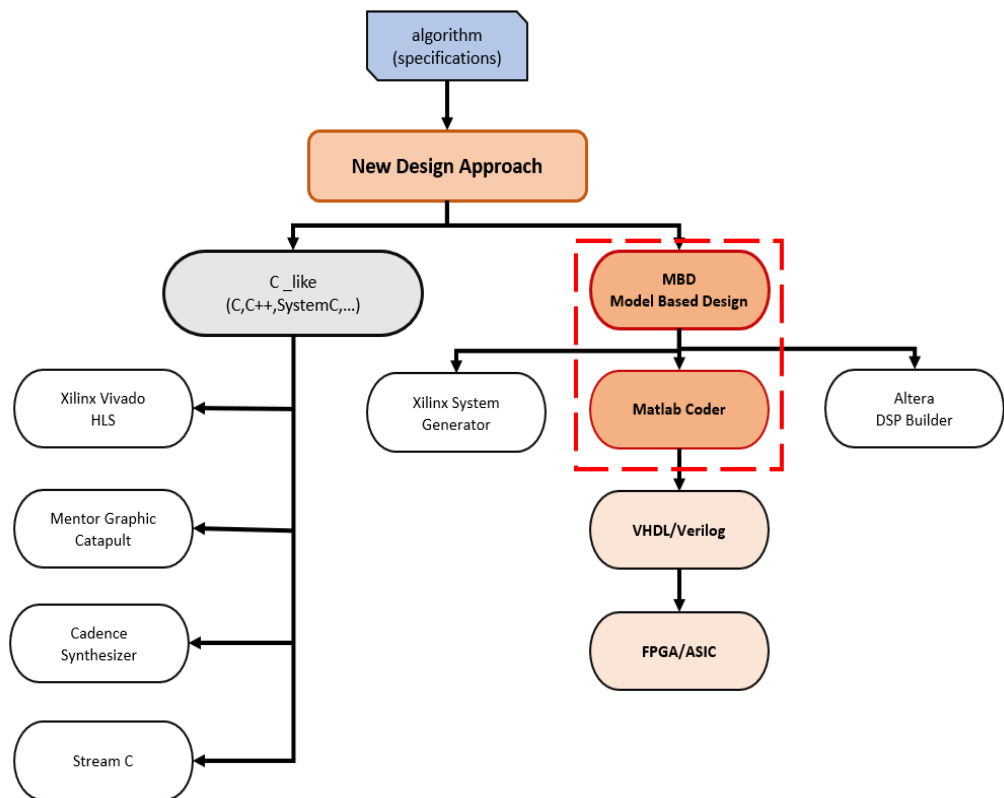


Figure 4-3 New Design Approach

In our work, we are interested in the MBD-based approach with the exploitation of the MATLAB HDL Coder design flow.

This new design methodology is based on the combination of the two approaches, namely the design approach based on high-level synthesis combined with the classical approach. The introduction of the HDL coder tool in the classical design flow allows to automate part of the generation of the RTL description. Thus, the designer will be able to easily go from the algorithmic domain to the generation of an IP and then implement it on FPGA target.

This new design methodology will be used for the hardware development of the bio-inspired FA_MPPT.

4.3 Model based design conception

As mentioned previously in a traditional work environment, the embedded code must be written by hand from a system model or from scratch. The software engineer compiles the control algorithm according to the specifications written by the control system engineer. Each step in this process (writing specifications, hand-coding algorithms, and debugging handwritten code) can be time consuming and error prone. With model-based design approach, the designer doesn't need to manually write thousands of lines of code, but generate code directly from the model, and the model acts as a bridge control between software and system engineers. The generated code can be used for rapid prototyping or production.

Rapid prototyping is a fast and inexpensive way to test algorithms in real time on hardware and perform design iterations in minutes instead of weeks. You can use prototype hardware or production Electronic Control Unit. Using the same rapid prototyping hardware and design templates, you can perform hardware-in-the-loop testing and other test and verification activities to verify hardware and software designs before production. Production code generation converts your design into actual code that will be implemented on the production embedded system. The generated code can be optimized for a specific processor architecture and integrated with traditional handwritten code.

4.3.1 HDL Coder

The conversion from MATLAB to HDL CODE is made using HDL Coder but it is also necessary to follow Model-Based Design to make it work properly. Model-Based Design is a mechanism that will show how to bridge the gap between the algorithmic description and the hardware implementation. One easy way to verify that hardware implementation matches system specifications is using Model-Based Design and HDL Coder (MATLAB application) together. It can also be useful to iterate designs faster.

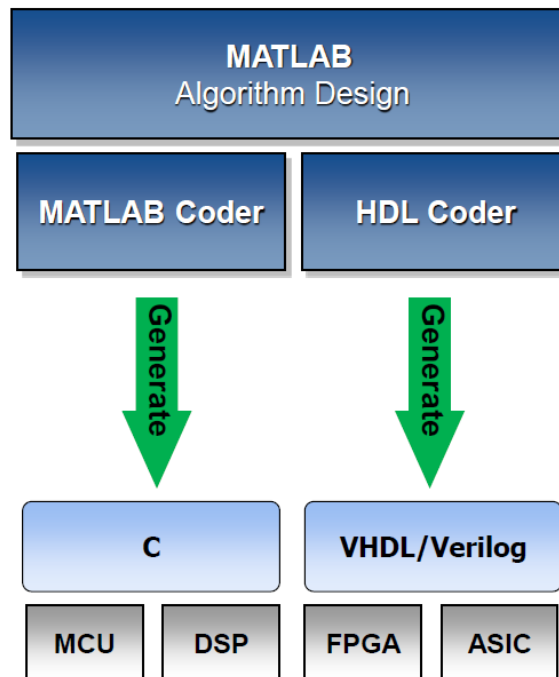


Figure 4-4 HDL Coding methods for FPGA Programming [34]

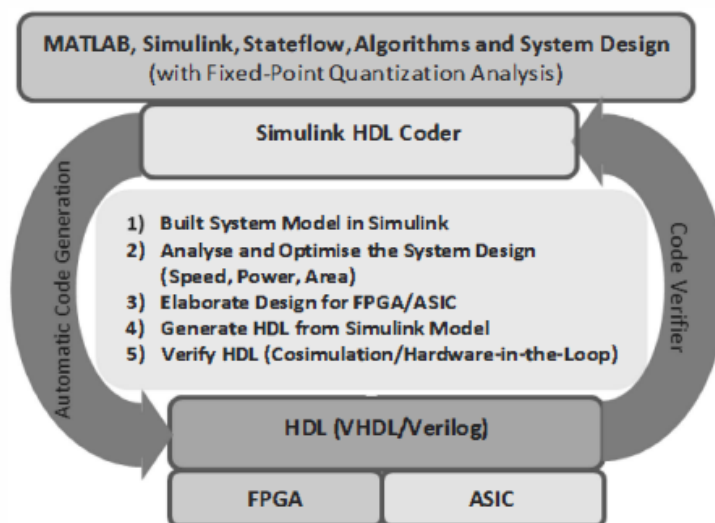


Figure 4-5 Method to generate HDL Code from MATLAB and Simulink [35].

The process of translating MATLAB designs to hardware consists of the following steps: [36]

- Model the algorithm in MATLAB - use MATLAB to simulate, debug, and iteratively test and optimize the design.
- Generate HDL code - automatically create HDL code for FPGA prototyping.
- Verify HDL code - reuse the MATLAB test bench to verify the generated HDL code.
- Create and verify FPGA prototype - implement and verify the design on FPGAs.

There are some unique challenges in translating MATLAB to hardware. MATLAB code is procedural and can be highly abstract; it can use floating-point data and has no notion of time. Complex loops can be inferred from matrix operations and toolbox functions.

Implementing MATLAB code in hardware involves:

- Converting floating-point MATLAB code to fixed-point MATLAB code with optimized bit widths suitable for efficient hardware generation.
- Identifying and mapping procedural constructs to concurrent area- and speed-optimized hardware operations.
- Introducing the concept of time by adding clocks and clock rates to schedule the operations in hardware.
- Creating resource-shared architectures to implement expensive operators like multipliers and for-loop bodies.
- Mapping large persistent arrays to block RAM in hardware

HDL Coder simplifies the above tasks through workflow automation.

4.3.1.1 HDL Workflow Advisor

The HDL Workflow Advisor helps automate the steps and provides a guided path from MATLAB to hardware. we can see the following key steps of the workflow in the left pane of the workflow advisor:

- a. Fixed-Point Conversion
- b. HDL Code Generation
- c. HDL Verification
- d. HDL Synthesis and Analysis

4.3.1.2 Fixed-Point Conversion

The floating-point to fixed-point conversion workflow consists of the following steps:

- a. Verify that the floating-point design is compatible with code generation.
- b. Propose fixed-point types based on computed ranges, either through the simulation of the testbench or through static analysis that propagates design ranges to compute derived ranges for all the variables.
- c. Generate fixed-point MATLAB code by applying proposed fixed-point types.
- d. Verify the generated fixed-point code and compare the numerical accuracy of the generated fixed-point code with the original floating-point code.

Note that this step is optional. we can skip this step if the MATLAB design is already implemented in fixed-point.

4.3.1.3 HDL Verification

HDL Coder generates VHDL and Verilog test benches from the MATLAB scripts for rapid verification of generated HDL code.

HDL Coder also works with HDL Verifier to automatically generate two types of co-simulation testbenches:

- HDL co-simulation verification works with Mentor Graphics ModelSim and QuestaSim, where MATLAB and HDL simulation happen in lockstep.
- FPGA-in-the-Loop simulation allows you to run a MATLAB simulation with an FPGA board in strict synchronization. You can use MATLAB to feed real world data into your design on the FPGA, and ensure that the algorithm will behave as expected when implemented in hardware.

4.3.1.4 HDL Synthesis

Apart from the language-related challenges, programming for FPGAs requires the use of complex EDA tools. Generating a bitstream from the HDL design and programming the FPGA can be daunting tasks. HDL Coder™ provides automation here, by creating project files for Xilinx and Altera that are configured with the generated HDL code. You can use the workflow steps to synthesize the HDL code within the MATLAB environment, see the results of synthesis, and iterate on the MATLAB design to improve synthesis results.

4.3.2 Vivado Xilinx

VIVADO is a part of tool suite from Xilinx. This latter is a software development environment for digital systems that allows you to go through all the development stages of a design project on an FPGA target. Designs can be described in three main forms: as schematics, HDLs, or state diagrams. In addition, the new versions facilitate the integration and reuse of designs in the form of IP (Intellectual Property) blocks generated from streams such as Vivado HLS, HDL Coder, ...etc., as well as improving design productivity with implementations on the latest families of programmable circuits from Xilinx. These include several features that not only optimize the delivered products, but also considerably reduce the development time.

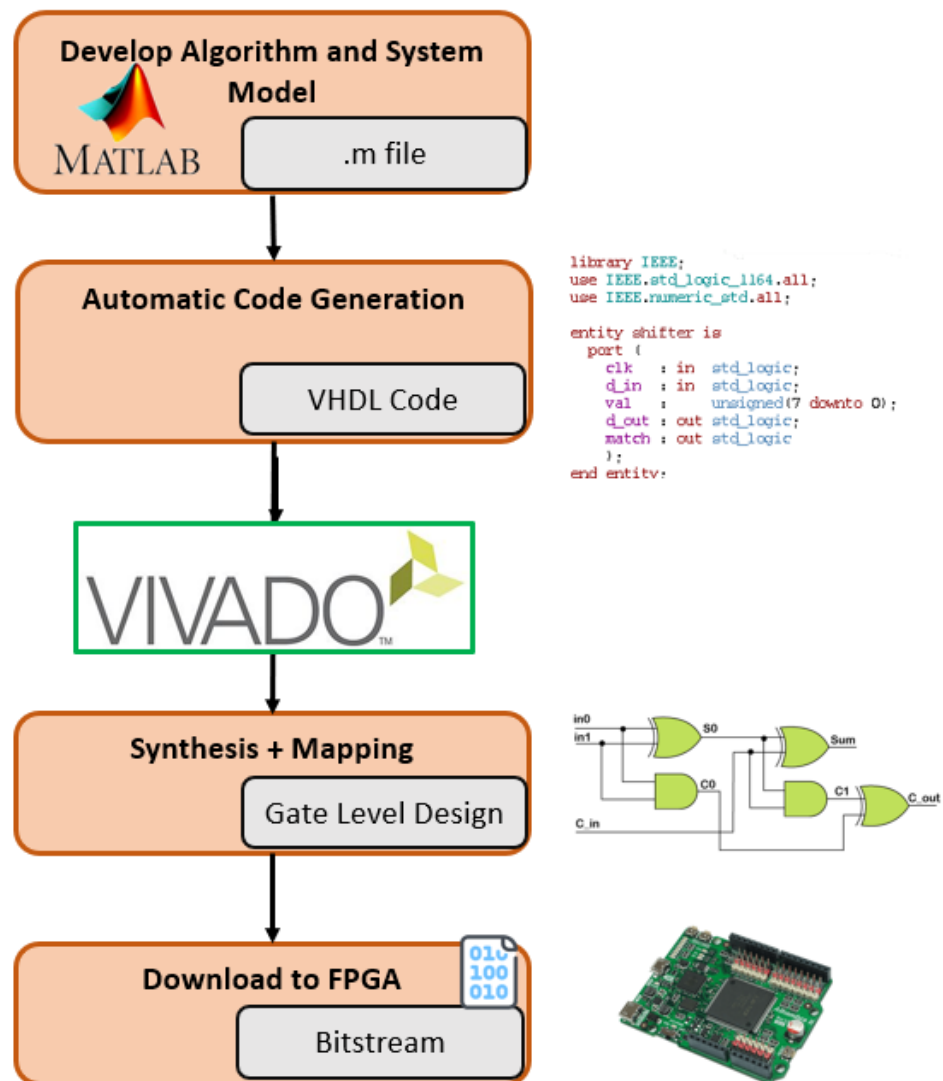


Figure 4-6 FPGA design flow based on VIVADO Design Suite.

4.4 Application of the MBD approach for the FA-MPPT controller

4.4.1 Introduction

After validation of the performances of the bio-inspired controllers developed in the software part (Chapter.3), during this section, the design methodology described in the previous chapter, is adopted for a fast prototyping and an efficient hardware implementation of these controllers from their ".m" description to the creation and hardware implementation of their respective IPs: FA_MPPT. The MATLAB_hdlcoder_Wokflow adviser tool will be used to generate the codes. The latter will be implemented on a Xilinx FPGA target with the Vivado ISE Design Suite tool.

4.4.1 Workflow Advisor in MATLAB

The HDL Workflow Advisor is a tool that checks the HDL compatibility of the algorithm, makes the HDL generation, verifies and gives reports of the generated HDL, and then uploads the code to the target device.

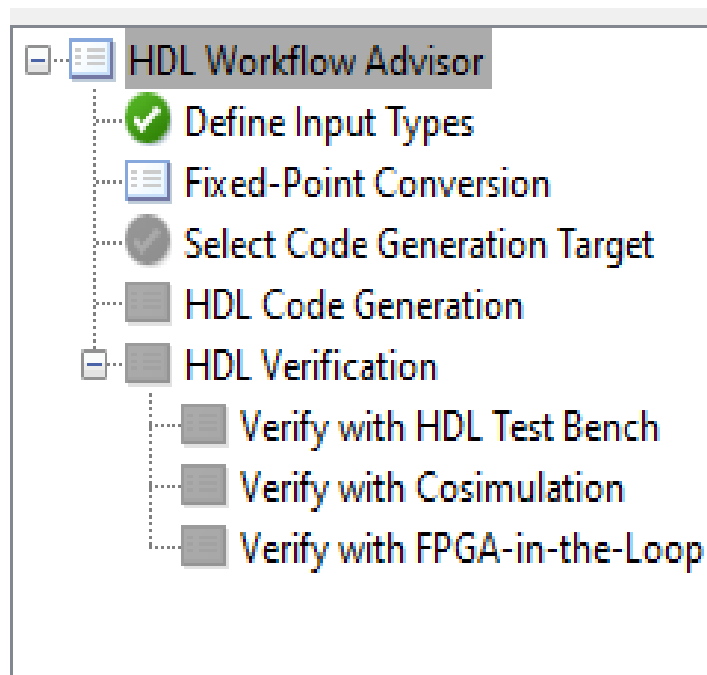


Figure 4-7 HDL Workflow Advisor

4.4.1.1 Launching HDL Coder

To run the workflow advisor in MATLAB, first from the toolstrip, we need to go to the APPS tab, and then choose the HDL Coder APP from the popped apps, like it's shown the following figure:

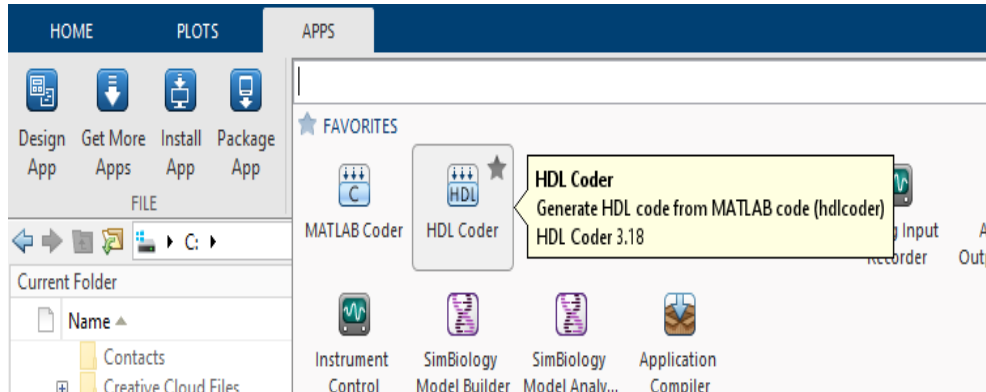


Figure 4-8 Launching HDL Coder in MATLAB

4.4.1.2 Targeting .m Function and Test Bench

After creating the project, in the right side, a menu will pop out, which in it we will add the main design function, beside its testbench (optional),

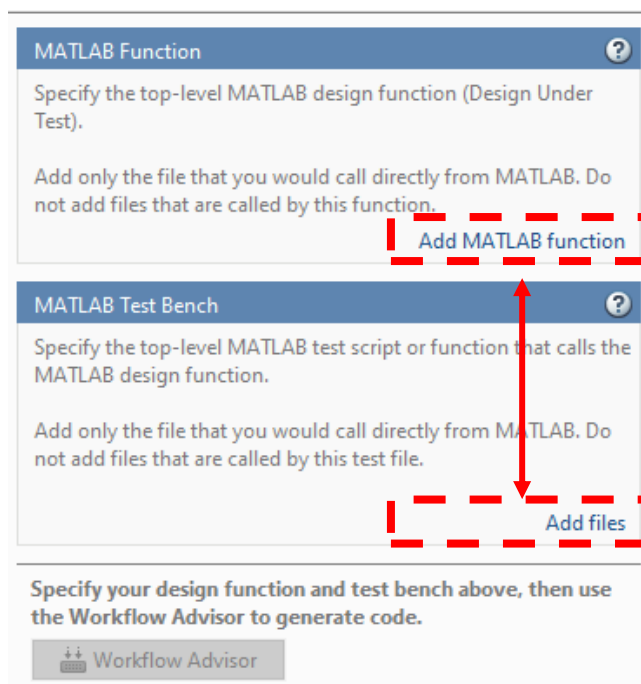


Figure 4-9 Specifying the required files

4.4.1.3 Defining input types

The next step after the addition, is to define the input types:

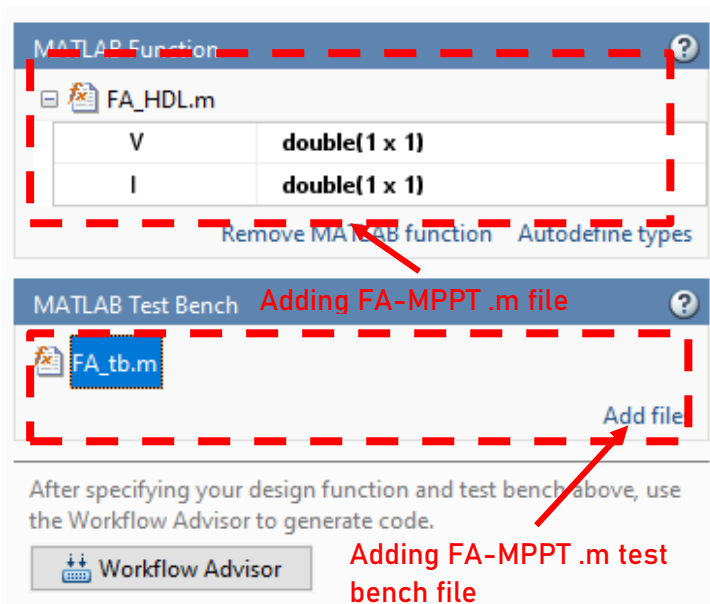


Figure 4-10 Input types

4.4.1.4 Converting to fixed point and targeting the device

After opening the Workflow Advisor, we need to choose fixed-point conversion:

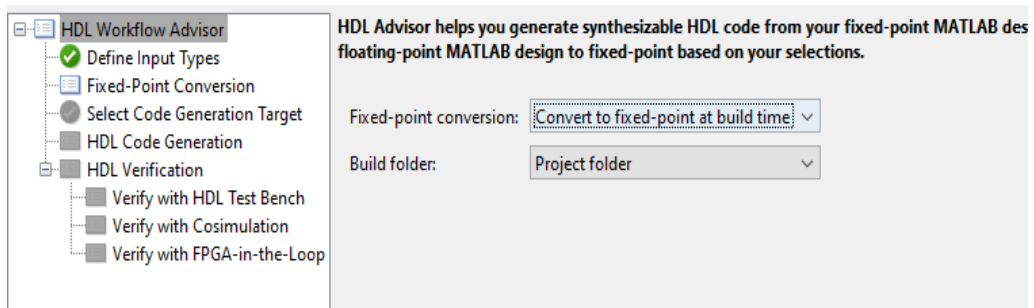


Figure 4-11 running fixed-point conversion

The next step, after the Fixed-Point Conversion, is to specify the target FPGA device, in our case, we will deploy the FA-MPPT controller in FPGA type Artix7 xc7a0100t.

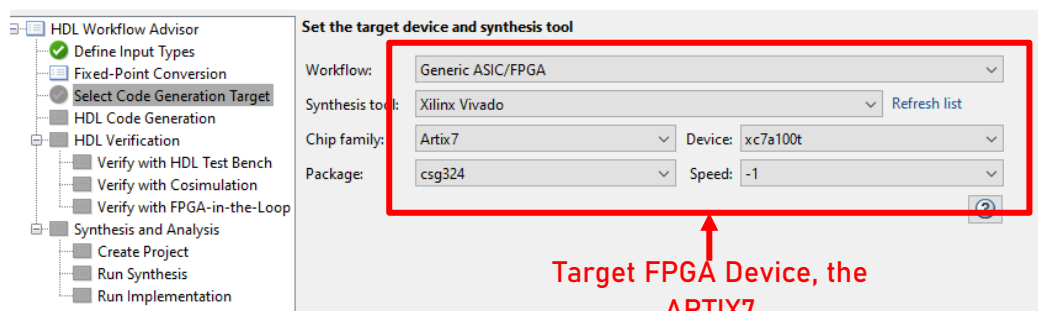


Figure 4-12 The target device

4.4.1.5 Generate HDL Code

After running the HDL code generation step, and if it was successfully finished, we need to jump to the next step, the HDL verification, this step is different and it depends on the desired parameters and the target workflow, we will skip it for now.

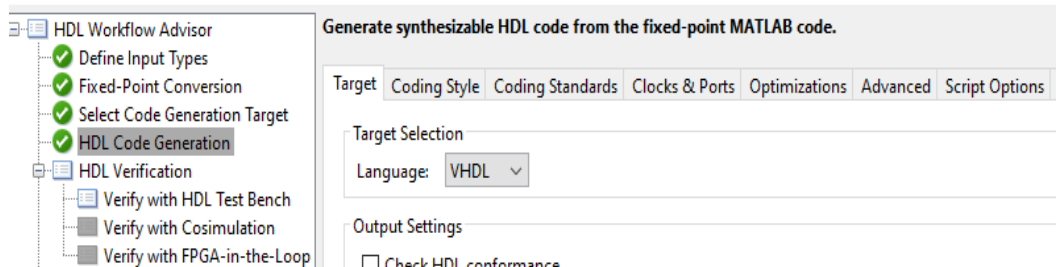


Figure 4-13 HDL Code Generation

4.4.1.6 Synthesis HDL Code

And finally, in order to implement the code in the FPGA, we have to run the Synthesis and Analysis step which will be discussed in details later.

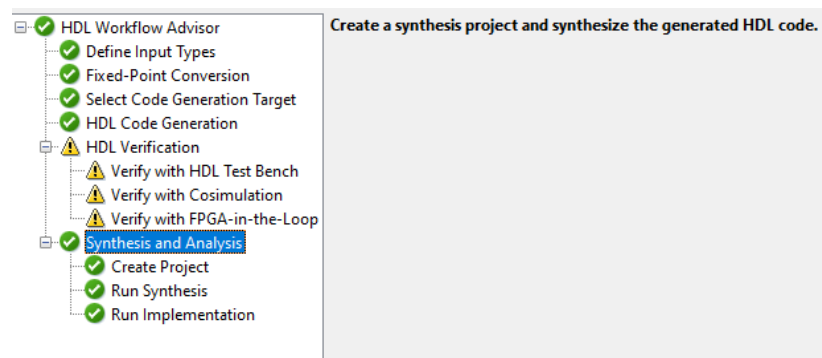


Figure 4-14 Synthesis and Analysis

4.4.2 Workflow Advisor in Simulink

4.4.2.1 Setting HDL Library

Before getting started with the HDL workflow adviser, it's necessary to check the supported block for HDL generation in Simulink, to find blocks that support HDL code generation, we need to enter the following line" **hdllib**" [37] to pop out the HDL library.

Another way to find libraries of blocks supported for HDL code generation, in the documentation, by filtering the block reference list. Click Blocks in the blue bar at the top of the Help window, then select the HDL code generation check box at the bottom of the left column. The blocks are listed in their respective products.

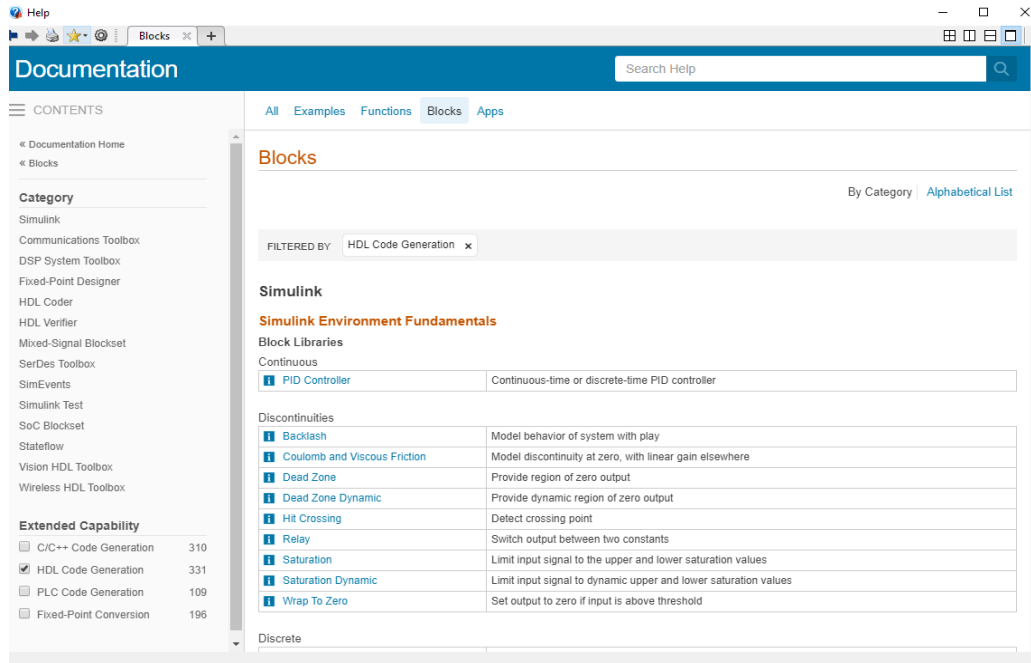


Figure 4-15 HDL Documentation Blocks

4.4.2.2 Launching HDL Workflow Advisor

To open the HDL workflow advisor in Simulink, we should either open it from the APPS tab, or by selecting the FA-MPPT DUT block, right click on it, HDL Code, then Workflow Advisor. the DUT is the Subsystem that contains the algorithm for which you want to generate code.

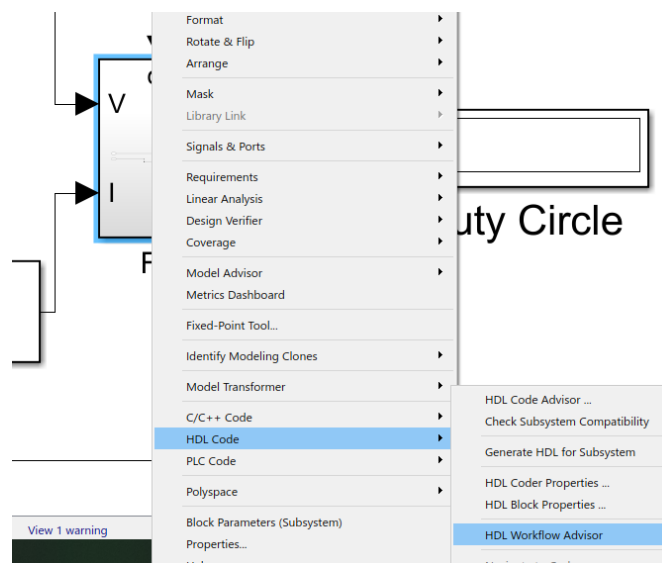


Figure 2 16 HDL Coder in DUT

4.4.2.3 Set target device and Synthesis Tool

Now for its tasks, we will begin with setting target device and synthesis tool, this task permits to choose the third-party software (here we have used Vivado 2019.2), and it allows us to specify the desired FPGA device in order to deploy the HDL code of FA-MPPT.

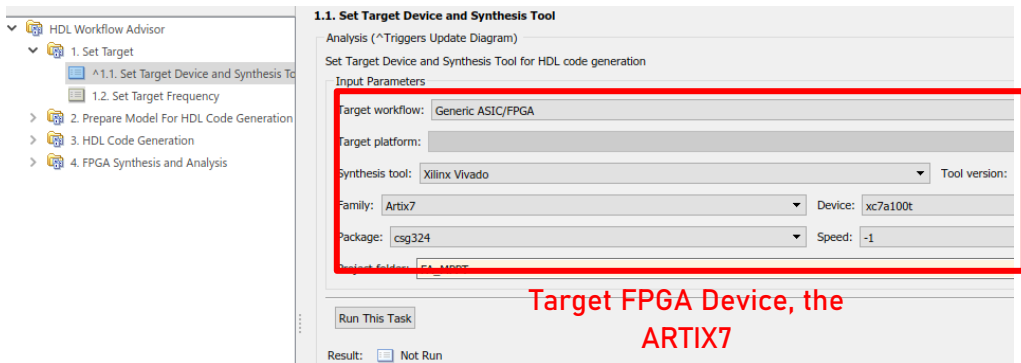


Figure 4-16 target device and synthesis tool

4.4.2.4 Set target frequency

After that, we should insert the target frequency, here we have used for example 100MHz for the target FPGA.

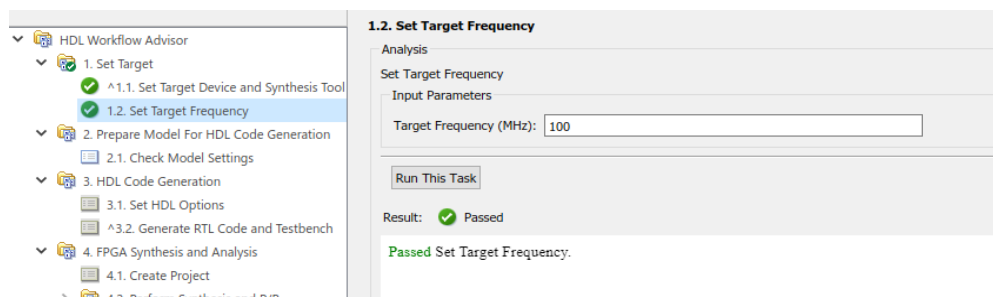


Figure 4-17 Target frequency

4.4.2.5 Check model settings

The tasks in this category check your model for HDL code generation compatibility. The tasks also report on model settings, blocks, or other conditions .it permits to modify the settings that doesn't fit the model in order to make the HDL generation step with less errors (such as setting the algebraic loop from warning to error), that would impact code generation, and provide advice on how to fix such problems, simply the Modify All button will fix them.

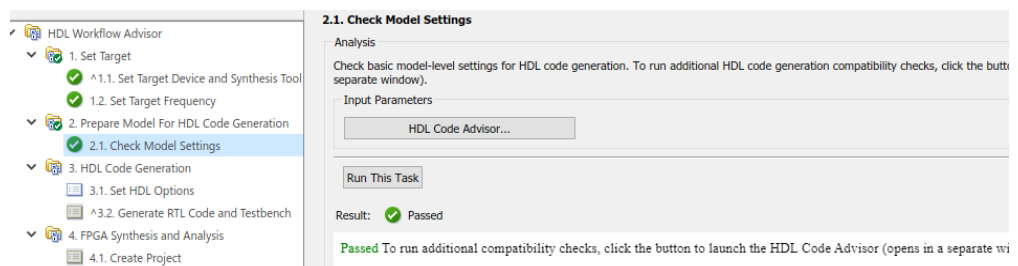


Figure 4-18 Check model settings

4.4.2.6 Set HDL Options

The HDL Workflow Adviser provides the option of setting HDL options; from there we can set up and configure the desired HDL parameters, such as setting the target of the synthesis tool (The tasks in this category enable you to select the desired target device and map its I/O interface to the inputs and outputs of your model), the package name and the family of it, it allows us also to set the parameters, the floating-point IP library, the Global setting and the test bench report.

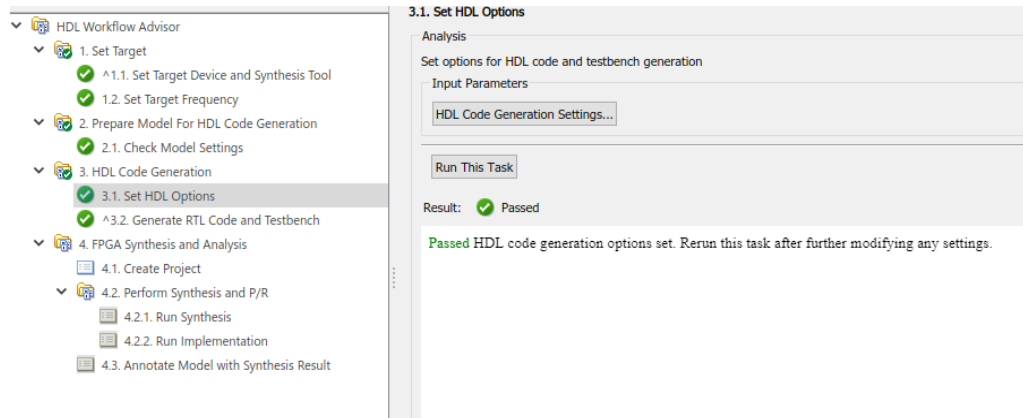


Figure 4-19 Set HDL Options

4.4.2.7 Generate RTL code and testbench

This task generates for the DUT subsystem, the RTL code. It generates a constraint file that contains pin mapping information and clock constraints, and can also generate the test bench of the system, beside generating the validation model which spots the generated delays and other differences between the original model and the generated co-simulation model.

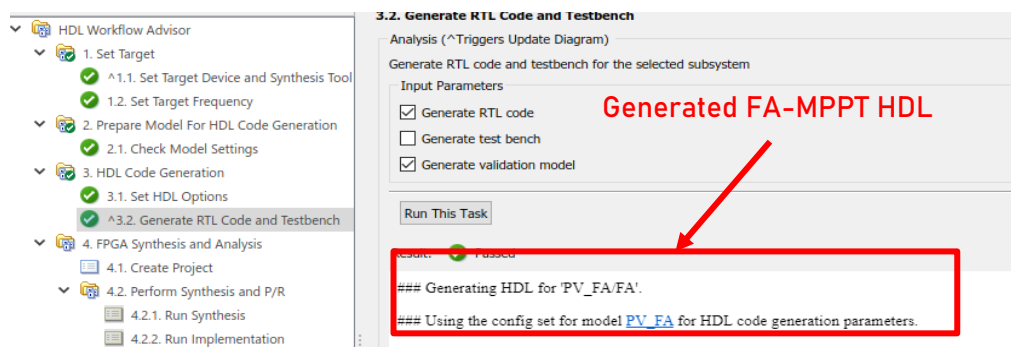


Figure 4-20 Generate RTL code and testbench

Once the code is generated, it will be used in the VIVADO flow for a hardware implementation through the creation of a hardware project.

4.4.2.8 Create project

This option makes a new project using Xilinx Vivado for the generated HDL code, beside the option of choosing the Synthesis objective, which are: Area Optimized, Compile Optimized, None and Speed Optimized.

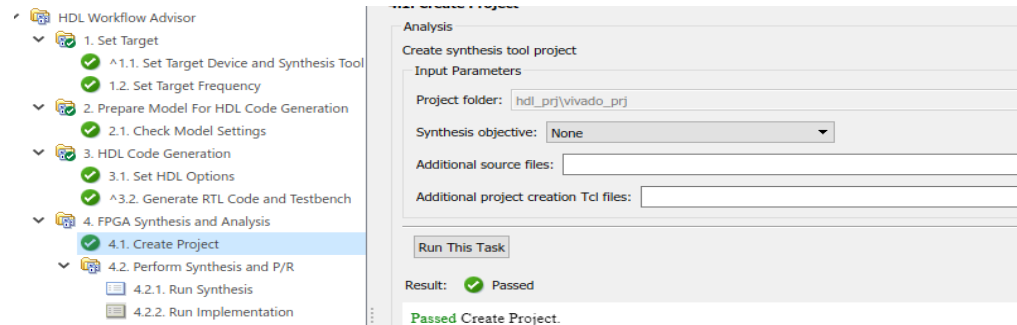


Figure 4-21 Create Project

4.4.2.9 Perform Synthesis and P/R

For Run synthesis, the task gives reports showing the amount of resources that have been consumed by the design and the data path delay, and information about LUTs and slices....

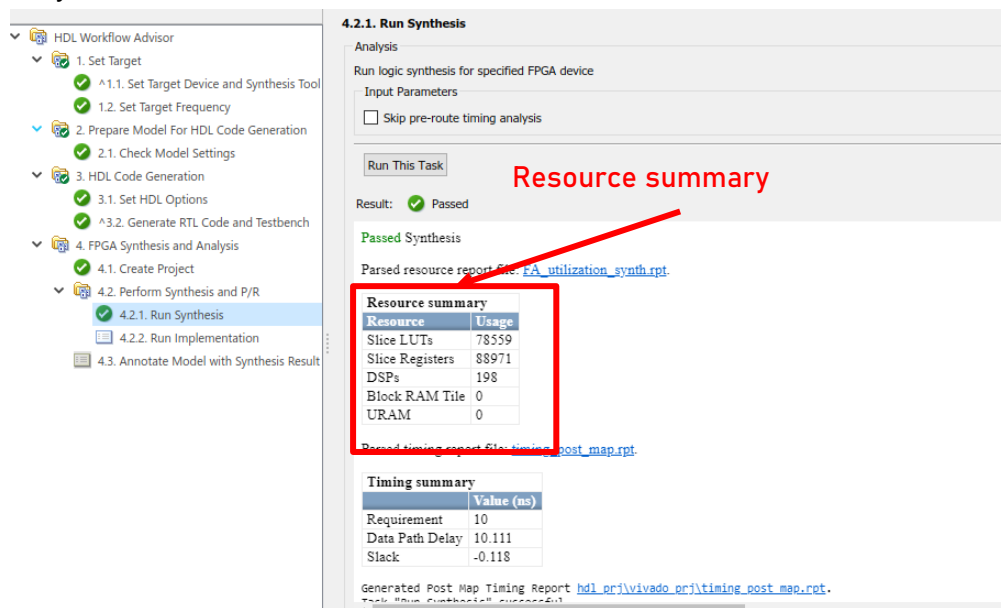


Figure 4-22 Synthesis task

For the task Run implementation, Simulink (using Vivado as a Third-Party software) does the necessary steps to do the placement and routing the netlist onto target resources, within the logical, physical, and timing constraints of the design.

4.4.3 The generated HDL

Finally, as depicted in figure bellow, the HDL code of FA algorithm was successfully generated, in order to verify, we opened the created project using vivado, and the result was like in the following figure:

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;
USE IEEE.numeric_std.ALL;
USE work.FA_pac.ALL;

ENTITY FA IS
PORT(
  clk      : IN    std_logic;
  reset    : IN    std_logic;
  clk_enable : IN  std_logic;
  V        : IN    std_logic_vector(63 DOWNTO 0); -- double
  I        : IN    std_logic_vector(63 DOWNTO 0); -- double
  S        : IN    std_logic_vector(63 DOWNTO 0); -- double
  ce_out   : OUT   std_logic;
  D        : OUT   std_logic_vector(63 DOWNTO 0) -- double
);
END FA;
```

Inputs and Outputs of the FA-MPPT

Main File

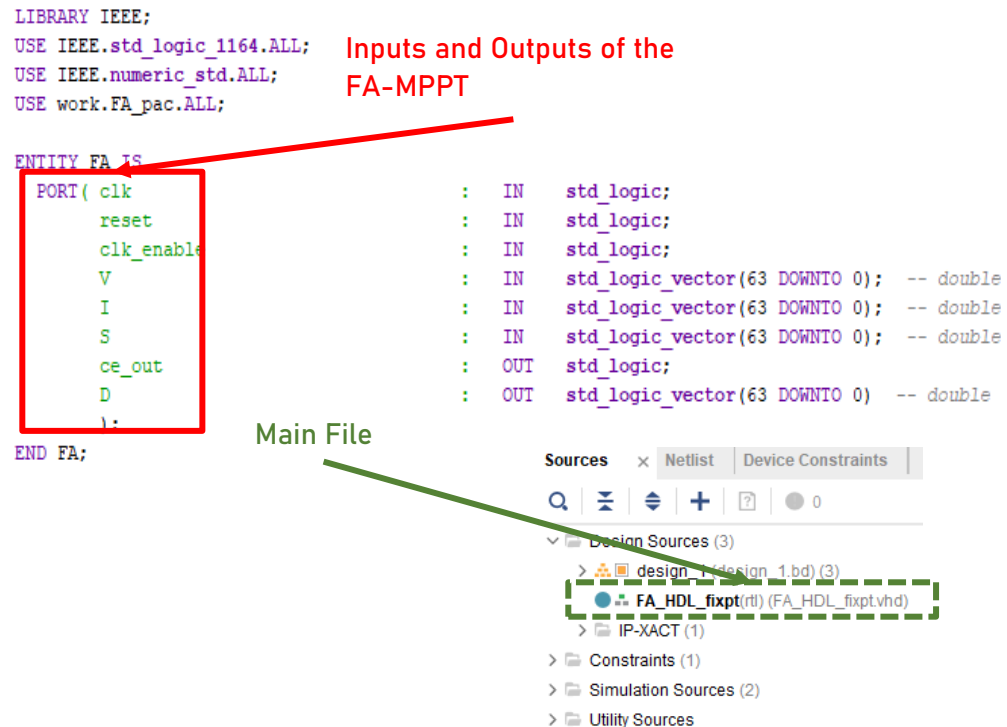


Figure 4-23 The generated VHDL in Vivado

As we can see, VHDL code of FA controller, with the inputs Voltage V, Current I, the Clock clk, with its pulse S, and the Duty cycle output D.

4.4.4 Synthesis & implementation

The FA generated code has been synthesized using the vivado tool, figure bellow depicted the results. In the Vivado's report, and fortunately, we can see that this time the code consumed only 15% of the available LUT, which means 9672/63400 only in the design.

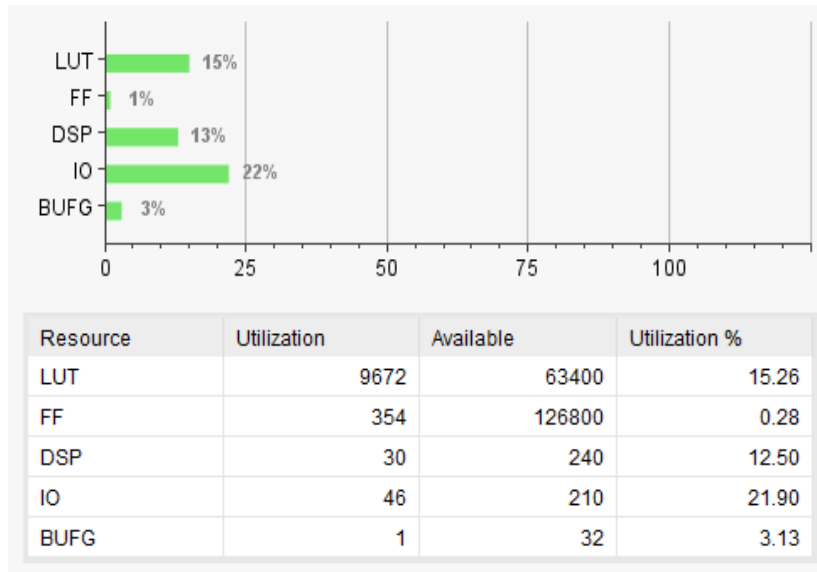


Figure 4-24 Vivado's report

The schematic the generated architecture, is depicted bellow, under the synthesis section, open synthesis design, we choose Schematic, there are for about 26 parts of the generated schema, for around 280 cells for each:

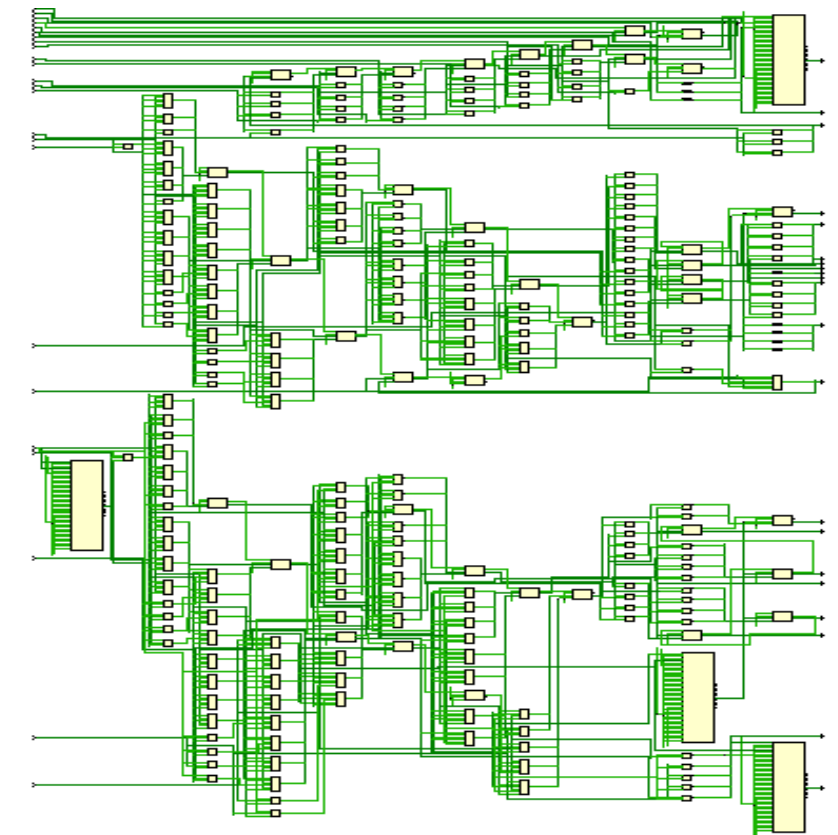


Figure 4-25 The synthesized FA-MPPT

The hardware implementation (mapping) of the FA MPPT controller is illustrated in figure 4-26 ... As we can see, the figure below represents the Mapping of the FA architecture in the target FPGA ARTIX7.

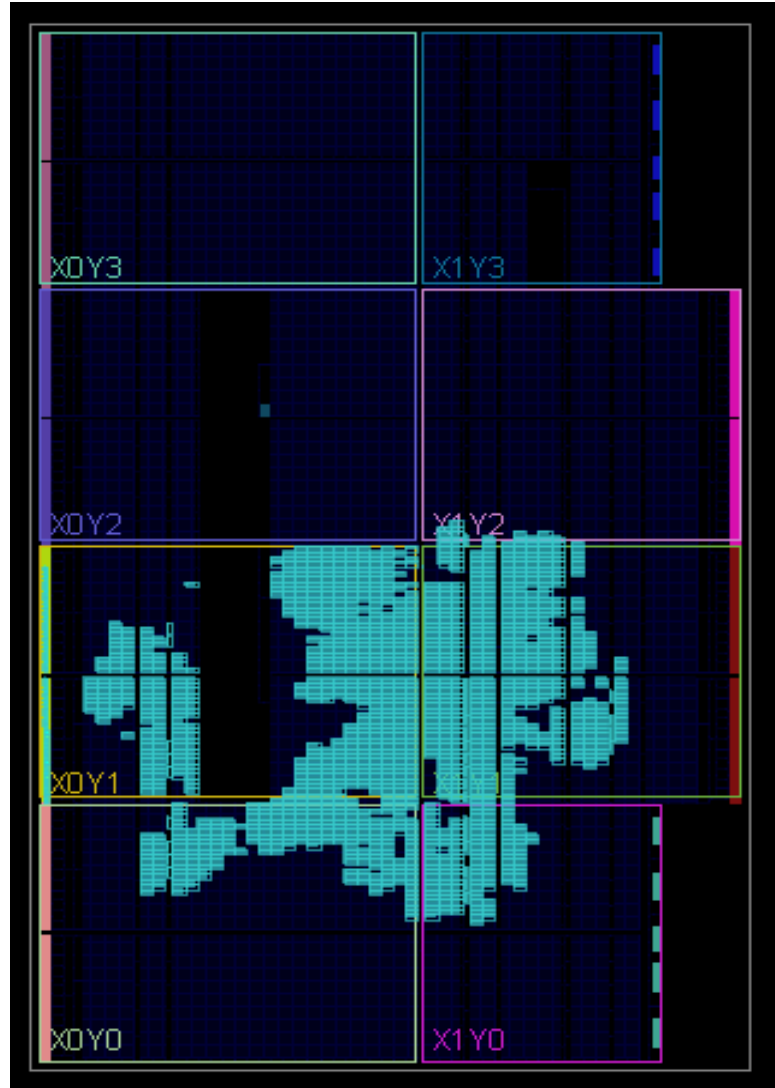


Figure 4-26 Synthesis design

4.4.5 Generating bitstream file using Xilinx Vivado

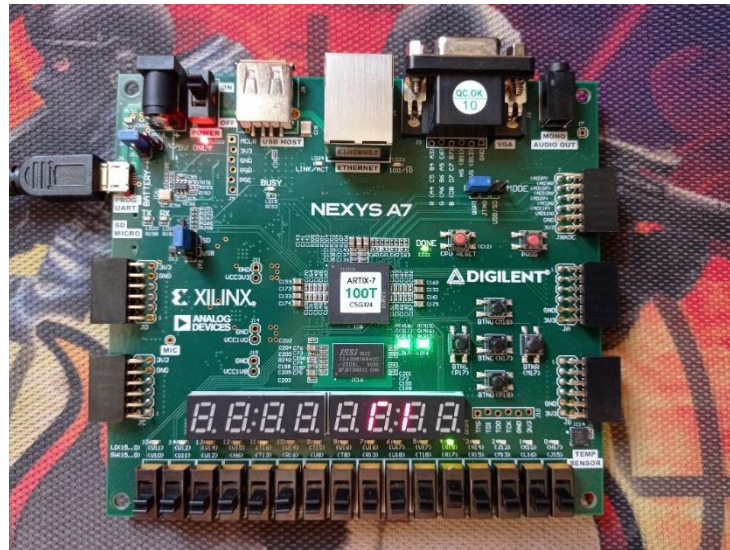


Figure 4-27 Artix7

To generate the bitstream file and link Vivado with the Artix7 board, we have to use the panel on the left, PROGRAM AND DEBUG section,

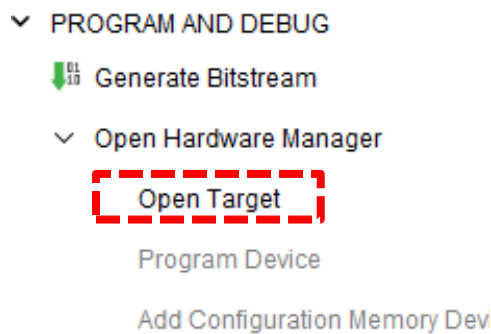


Figure 4-28 PROGRAM AND DEBUG

After generating the bitstream, we open the target device and select auto connect, and the FPGA device will be detected.

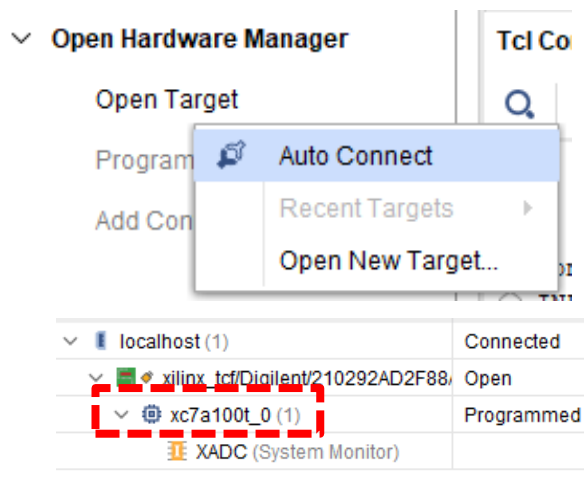


Figure 4-29 auto connect option in Vivado

Finally, when Vivado is fully connected with the target, we choose Program Device option to deploy the HDL code of the FA algorithm into the FPGA, when the operation is successfully completed, the device will be programmed a LED will turn on in the FPGA, labelled with DONE.

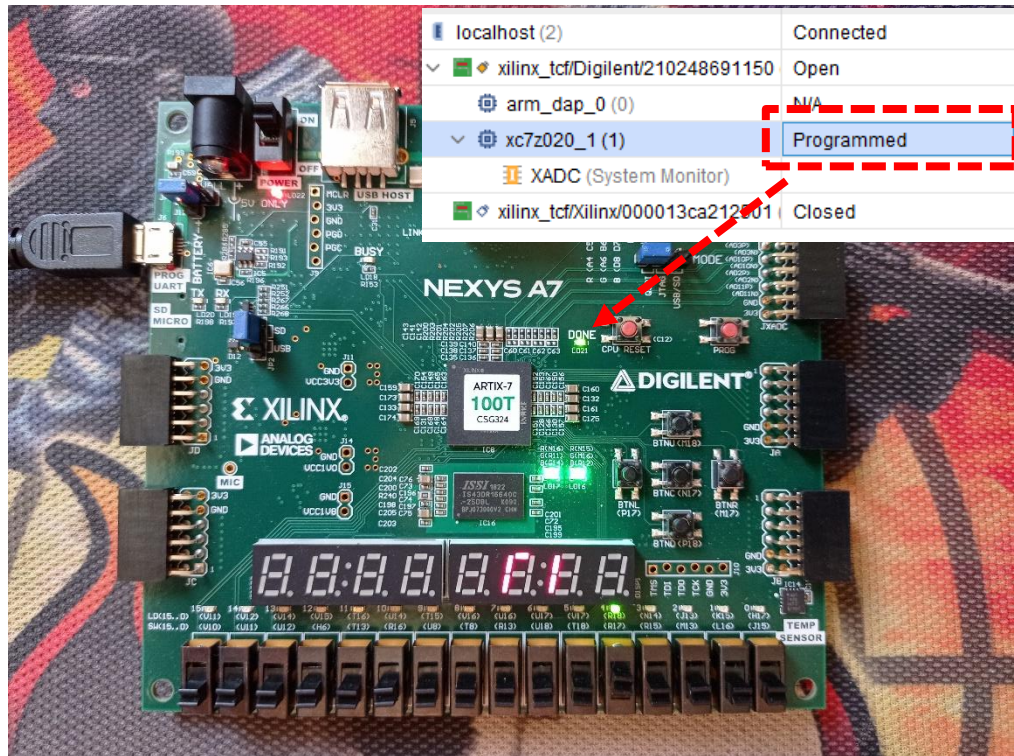


Figure 4-30 Programmed Device

4.4.6 FPGA in the loop

FIL does the test of the generated design in real hardware for the generated HDL code. In Simulink, it generates a Simulink FIL block that represents the HDL code, in the other hand, in MATLAB it deploys the generated programming file onto the FPGA then shows the results. The programming file is loaded onto an FPGA with JTAG connection. TX/RX of data from Simulink to FPGA is via gigabit Ethernet crossover cable that co-simulate and verifies the design in real time environment [5].

Creating an FPGA-in-the-loop link between the simulator and the target enables allows to: [38]

- Verify HDL implementations directly against algorithms in Simulink or MATLAB.
- Apply data and test scenarios from Simulink or MATLAB to the HDL design on the FPGA.
- Integrate existing HDL code with models under development in Simulink or MATLAB.

To launch FIL, we should write the following line “[FPGA-in-the-Loop Wizard](#)”, and after that inserting the requested options as follows:

Verify Code with HDL Testbench simulates the generated HDL design under test (DUT) with test vectors from the test bench using the specified simulation tool [39].

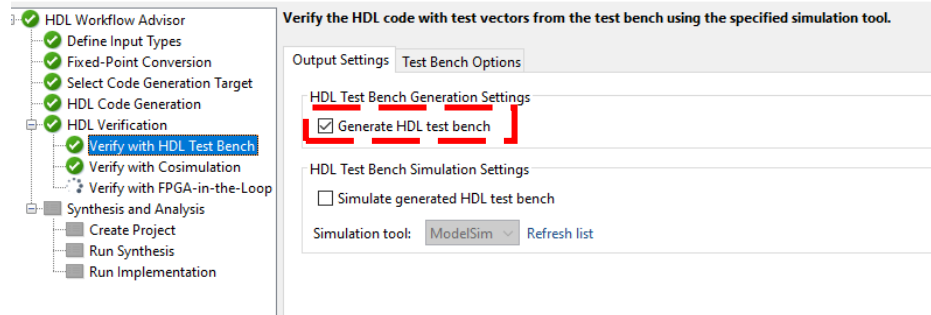
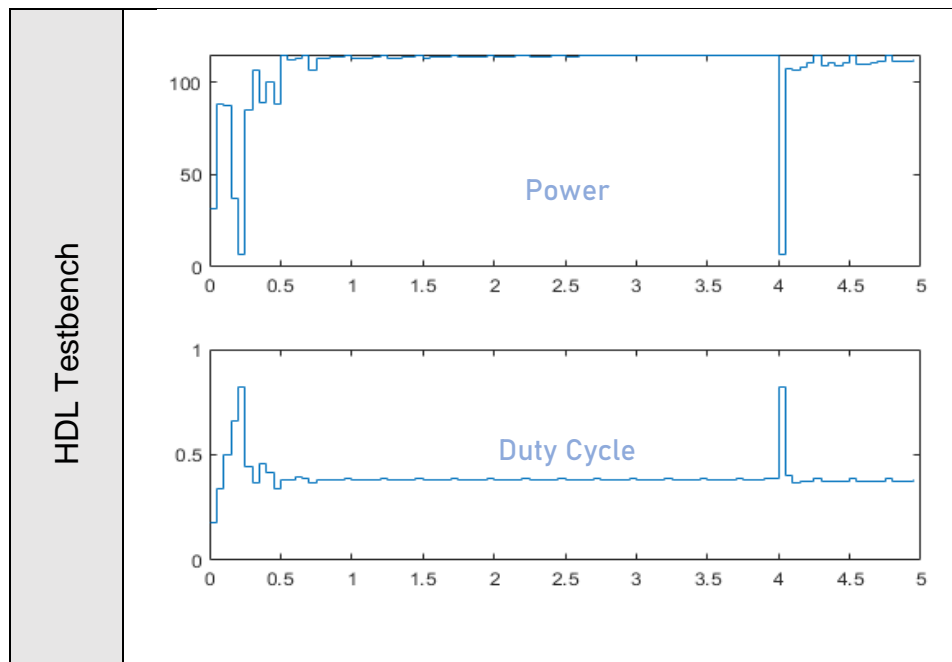


Figure 4-31 Verify Code with HDL Test Bench

This will give us the following plot, with a temperature of 25°C and irradiation for each panel is respectively 1000 ,800 ,600 300.

The results of HDL testbench is as MATLAB simulation results, as shown in the figure 4.33, under the latter partial shading conditions, except the showed parasite above after 4 seconds, we can remark that the power is converging to the max after certain time, which means that the generated FIL is working.



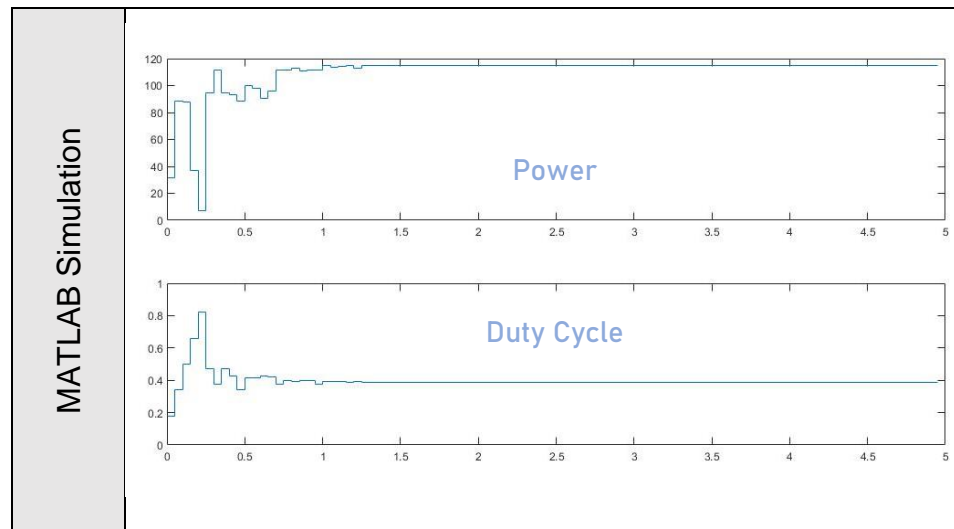


Figure 4-32 HDL testbench result

4.5 Conclusion

The generated HDL code from a MATLAB “.m” file gives an accurate result in time, this latter can be simulated and synthesized with variety of software, beside implementing it onto an FPGA circuit. MATLAB can control the HDL architecture and implementation, generates a detailed report showing the amount of consumed resources. This design method from modelling to setting a programmable device implementation becomes recursive. By this method, the generated result system, with its behaviour and synthesis can be simulated and tested. The modern programmable devices in combination with appropriate software packages of synthesis and simulation gives a significant accelerated design process for the electronic systems.

This automatic generation approach of a hardware descriptions block-built models of functions and systems saves significant amount of time comparing to writing the code from scratch, besides debugging and verifying the generated hardware.

We can conclude that this approach is suitable for the hardware generation using MATLAB. It also provides an appropriate environment for simultaneous design, simulation, analysis and visualise with MATLAB and Simulink. All of these give a contribution to the development of the hardware-based system design in general, which made a significant enhancement beside overcoming the difficulty of writing the HDL code from scratch.

General Conclusion

General conclusion

The ultimate goal of this research is to contribute to the optimisation of photovoltaic systems, through the reduction of losses, particularly those arising from the presence of partial shading. The objective of the Maximum Power Point Tracking (MPPT) algorithm is to optimise the operating point of the PV system at the MPP point of the I-V curve where the module produces its maximum power. However, achieving this goal is a demanding task because the MPP on the P-V curve is unstable due to the continuous variation of solar radiation and temperature.

Considering the ability of metaheuristics to handle multimodal functions, an MPPT based on the FA algorithm was proposed. The proposed MPPT control was able to track the maximum power point of the PV generator with high efficiency. Moreover, this performance was evaluated under different partial shading conditions (temperature and irradiation).

From the simulation results, it is concluded that the proposed FA method tracks the GMPP with high accuracy and gives a satisfactory efficiency. Moreover, it outperforms the PSO method in terms of efficiency, convergence time and GMPP tracking.

In this thesis a BUCK-BOOST converter has been used in the optimisation of a photovoltaic system. Beside the FA-MPPT controller that we have targeted choosing the MBD approach which is time saving comparing to writing from scratch, by converting it to HDL Code, synthesizing and implementing using MATLAB HDL Coder and Vivado software as third-party, the process passed by a lot of optimizations in order to consume less usage resources in the target FPGA ARTIX7, ending it with the HDL test, which gave satisfactory results.

Regarding the perspectives of this work, we can mention:

- Using the proposed algorithm to optimize the power for different configurations of the photovoltaic field.
- Extend the comparison with other methods, it would be interesting to compare the proposed algorithm with other meta-heuristics such as Ant Colony Optimization (ACO), Cuckoo Search (CS) and Seagull optimization Algorithm (SOA).
- Hardware implementation on FPGA device of this algorithm for experimental validation using FIL method

References

References

- [1] D. L. M. C. P. L. L. Jiang, "A novel ant colony optimization-based maximum power point tracking for photovoltaic systems under partially shaded conditions," in *Energy Build*, 2013.
- [2] M. V. F. T. N. F. H. K. M. Miyatake, "Maximum power point tracking of multiple photovoltaic arrays: a pso approach," in *Journal IEEE Trans: Aerospace and, Electronic Systems*, Vol.47, N° 1, pp. 367–380,, 2011.
- [3] A. C. K. K. S. S. O. S. A.S. Benyoucef, ""Artificial bee colony-based algorithm for maximum power point tracking (MPPT) for PV systems operating under partial shaded conditions," *Appl Soft Comput.*, 2015.
- [4] G. Li, Y.Jin, M.W. Akram, X. Chen, J. Ji, Guiqiang. Lia, "Application of bio-inspired algorithms in maximum power point tracking for PV systems under partial shading conditions," in *Renewable and Sustainable Energy Reviews*, Vol.81, pp. 840–873, 2018.
- [5] G. E. T. Yam P. Siwakoti, "Design of FPGA-Controlled Power Electronics and Drives Using MATLAB Simulink," Australia..
- [6] M. Rabiaa, Contribution à l'étude des onduleurs dans les systèmes, thèse de doctorat en sciences,université de Batna, 2015.
- [7] Wikipedia, "Cellule photovoltaïque de quatre pouces en silicium polycristallin," 2007. [Online]. Available: https://fr.wikipedia.org/wiki/Cellule_photovolta%C3%AFque#/media/Fichier:4inch_poly_solar_cell.jpg.
- [8] W.Xiao, Modified Adaptative Hill Climbing Maximum Power Point Tracking (MPPT) Control Method For Photovoltaic Power Systems, The University of British Columbia., Thesis for degree of Master,, 2003.
- [9] "Solar Cell I-V Characteristic," [Online]. Available: <https://www.alternative-energy-tutorials.com/photovoltaics/solar-cell-i-v-characteristic.html>.
- [10] alternative-energy-tutorials, "Bypass Diode for Solar Panel Protection," [Online]. Available: <https://www.alternative-energy-tutorials.com/photovoltaics/bypass-diode.html>.
- [11] S. N. R. K. N. D. V. Nikhil Kumar, "A state-of-the-art review on conventional, soft computing, and hybrid techniques for shading mitigation in photovoltaic applications,Volume30, Issue9," 13 April 2020. [Online]. Available: <https://onlinelibrary.wiley.com/doi/full/10.1002/2050-7038.12420>. [Accessed September 2020].
- [12] W. K. M. Y. C. M. a. C. D. D. G. Bunea, "Effect of Light Intensity," [Online]. Available: <https://www.pveducation.org/pvcdrom/solar-cell-operation/effect-of-light-intensity>.

- [13] S. K. S. P. G. R. K. U. Shinde, Solar PV emulator for realizing PV characteristics under rapidly varying environmental conditions, IEEE, 2016.
- [14] "Effect of Temperature," [Online]. Available: <https://www.pveducation.org/pvcdrom/solar-cell-operation/effect-of-temperature>.
- [15] A. A. a. H. S. Lahcen El Mentaly, "Comparison between Seven MPPT Techniques Implemented in a Buck Converter [Vol. 12 , Issue. 6] Pages 476 - 486 (11)," [Online]. Available: <http://www.recentadvancesinelectricalandelectronicengineering.com/articles/163535/comparison-between-seven-mppt-techniques-implemented-in-a-buck-converter>.
- [16] W. G. D. X. Weidong, "A modified adaptive hill climbing MPPT method for photovoltaic power systems," in *Power Electronics Specialists Conference*, vol. 3, pp. 1957-1963, 2004.
- [17] C.-E. Y. I. ., C. Yu-Pei Huang, "A Modified Firefly Algorithm with Rapid Response Maximum Power Point Tracking for Photovoltaic Systems under Partial Shading Conditions," Department of Electronic Engineering, National Quemoy University, Kinmen County 892, Taiwan;.
- [18] Y.-P. Huang and S.-Y. Hsu, "A performance evaluation model of a high concentration photovoltaic module with," 331-342, 2016.
- [19] G. G. M. N. Femia, "Optimization of perturb and observe maximum power point tracking method," IEEE Trans: Power Electron, Vol. 20, N°4, pp. 963-973, 2005.
- [20] M. R. D.P. Hohm, "Comparative Study of Maximum Power Point Tracking Algorithms," Journal Progress in Photovoltaic: Research and Applications, vol.11, N°1, pp.47-62,, 2003.
- [21] P. C. ., T ESRAM, "Comparison of photovoltaic array maximum power point tracking techniques," IEEE Transactions on energy conversion 22, 2007.
- [22] K. Sobaihi, "Etude et réalisation d'un hacheur de tracking (MPPT) à contre réaction de," in *Mémoire de Magister, Ecole Nationale Polytechnique*, 2003.
- [23] L. Z. A. Al-Amoudi, "Optimal control of a grid-connected pv system for maximum power point tracking and unity power factor," seventh international conference on power electronics and variable speed drives, N°456, pp. 80-85, 1998.
- [24] F.Belhachat, "Commande neuro-floue d'un hacheur MPPT," in *Mémoire de Magister, Ecole Nationale Polytechnique d'Alger*, 2007.
- [25] H. Knopf, "Analysis, Simulation, and Evaluation of Maximum Power Point Tracking (MPPT) Methods for a Solar Powered Vehicle1," in *Master Thesis, Portland State University*, 1999.
- [26] V. G. K. K. R. D. B. M. R. Ramaprabha, "Maximum power point tracking using GA-optimized artificial neural network for solar PV system," in *Conference on Electrical Energy Systems (ICEES)*, pp.264 - 268, 2011.

- [27] S. S.Binitha, "A survey of bio inspired optimization algorithms," *Int J Soft Comput Eng*, Vol.2, N°.02, pp.137-151, 2012.
- [28] Z. S. J. Ahmed, "A maximum power point tracking (MPPT) for PV system using cuckoo search with partial shading capability," in *Appl Energy*, , Vol.119, pp.118-130,, 2014.
- [29] S. Titri, "Circuit reconfigurables pour la gestion de l'énergie dans les systèmes photovoltaïques," in *Doctorat, ECOLE NATIONALE POLYTECHNIQUE d'Alger*,, 2018.
- [30] Xin-SheYang, "Nature-Inspired Metaheuristic Algorithms," in *in Engineering Optimization An Introduction with Metaheuristic Applications*,, UK, University of Cambridge,, 2010.
- [31] Xin-SheYang, "Engineering Optimization An Introduction with Metaheuristic Applications," in *Contemporary Physics*, 2012.
- [32] A. Blanchardon., "ynthèse d'architectures de circuits FPGA tolérants aux défauts. Technologies," Université Pierre et Marie Curie - Paris VI, 2015, 2015PA066274 . tel-01243976v2 .
- [33] Y. P. S. Graham E.Town, "Design of FPGA-Controlled Power Electronics and Drives Using MATLAB Simulink," Australia.
- [34] RAPID PROTOTYPING FROM ALGORITHM TO FPGA, "Joonas Järviluoma," 2015.
- [35] Loren, "MATLAB to FPGA using HDL Coder," 2013. [Online]. Available: <https://blogs.mathworks.com/loren/2013/04/11/matlab-to-fpga-using-hdl-codertm/>.
- [36] MATLAB, "Find Blocks That Support HDL Code Generation," [Online]. Available: <https://fr.mathworks.com/help/comm/ug/find-blocks-and-system-objects-supporting-hdl-code-generation.html>.
- [37] MATHWORKS, "FPGA-in-the-Loop," [Online]. Available: <https://fr.mathworks.com/help/hdlverifier/fpga-in-the-loop.html>.
- [38] MATHWORKS, "Verify Code with HDL Test Bench," [Online]. Available: <https://ww2.mathworks.cn/help/hdlcoder/ug/verify-with-hdl-test-bench.html>.
- [39] D. B. A. B. H. M. Patricia de Rango, "sciencedirect," Volume 18, Issues 7-8, September-October 2017, Pages 381-390, [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1631070517300555>.