

République Algérienne Démocratique et Populaire

Université Saad Dahlab, Blida-1

Faculté de Technologie

Département des Énergies Renouvelables



**Présenté par :**

AMEYOUUD Meriem Sirine

MEGNOUCHE Imene

**Pour l'obtention du diplôme de MASTER en énergie  
renouvelable**

Option : Conversion photovoltaïque

**Thème :**

**CONTRIBUTION A L'IMPLEMENTATION FPGA DU FILTRAGE  
NUMERIQUE FIR POUR LA COMMANDE MPPT**

**Soutenue devant le jury composé de :**

Monsieur T.DOUMAZ	MAA	USDB	Président
Monsieur O.AIT SAHED	MCB	USDB	Examineur
Monsieur R.BOUKENOUI	MCB	USDB	Examineur
Monsieur M.MAAMOUN	MCA	USDB	Encadreur

2021-2022

## الملخص

في هذا العمل، تم تطوير نهج جديد لوحدة تحكم الحد الأقصى لتتبع نقطة الطاقة MPPT ، أقل تعقيدا وأكثر كفاءة. وسوف يستند هذا النهج على MPPT الكلاسيكية من قبل طريقة الاضطراب والمراقبة، يتم ترجمة البرنامج إلى لغة البرمجة عالية السرعة الدوائر المتكاملة وصف الأجهزة. وتستخدم أدوات المحاكاة مثل Xilinx و SIMULINK و MATLAB. وتثبت النتائج أن خوارزمية تحكم الحد الأقصى لتتبع نقطة الطاقة يمكن تنفيذها بسهولة باستخدام (المجال للبرمجة بوابة صيف) FPGA في الوقت الحقيقي.

**الكلمات المفتاحية:** مولد الكهروضوئي، خوارزمية الاضطراب والمراقبة، MPPT, VHDL, FPGA.

## Résumé

Dans ce travail, une nouvelle approche est développée pour le contrôleur MPPT, moins complexe et plus performant. Cette approche sera basée sur un MPPT classique par la méthode perturbe and observe, ce programme est traduit en langage de description de matériel de circuit intégré à haute vitesse VHDL. Les outils de simulation comme Xilinx et SIMULINK-MATLAB sont utilisés. Les résultats prouvent que l'algorithme MPPT accordée à un filtre FIR pourrait être facilement mis en œuvre en utilisant FPGA (Field Programmable Gate Array) en temps réel.

**Mots-clés:** Générateur PV, algorithme P&O ,MPPT, VHDL,FPGA.

## Abstract

In this work, a new approach is developed for the MPPT controller, less complex and more efficient. This approach will be based on a classic MPPT by the perturb and observe method, the program is translated into a very high-speed integrated circuit hardware description language VHDL. Simulation tools like Xilinx and SIMULINK-MATLAB are used. The results prove that the MPPT algorithm granted to an FIR filter could be easily implemented using FPGA (Field Programmable Gate Array) in real time.

**Keywords:** PV generator, P&O algorithm, MPPT, VHDL, FPGA.

## **Remerciement**

*A Monsieur **D.LAFRI** notre chef de département pour son soutien, son attention.*

*J'adresse mes remerciements à Monsieur **T.DOUMAZ**, notre chef d'option, pour les efforts prodigués, avec générosité et assistance tout le long de ce travail.*

*Je tiens à remercier sincèrement **les membres du jury** qui me font le grand honneur d'évaluer ce travail.*

*Mes plus profonds remerciements à Monsieur **M.MAAMOUN** non seulement d'avoir accepté l'encadrement de ce travail, mais surtout pour son aide, ses orientations, et ses encouragements, sa gentillesse tout au long de la réalisation de ce présent travail.*

*Et je remercie mon binôme **I.MEGNOUCHE** pour m'avoir accompagné tout au long du chemin.*

*L'expression de ma plus profonde gratitude et mes vifs remerciements à ma famille pour leur soutien tout au long du chemin.*

**AMEYOUUD MERIEM**

**SIRINE**

## **Remerciement**

*Je remercie en premier lieu Dieu qui m'a donné la force et la patience pour terminer ce travail.*

*Je tenais à exprimer mes remerciements à notre encadreur **M.MAAMOUN** qui a proposé et a dirigé ce travail et à mon binôme **M.AMEYLOUD**.*

*Je remercie monsieur le président de jury, ainsi que les membres de jury d'avoir accepté de juger ce travail.*

*Ce travail de mémoire doit beaucoup à certaines personnes que je tenais à les remercier sincèrement.*

*Enfin, je remercie toutes les personnes, qui ont contribué de près ou de loin, directement ou indirectement à l'aboutissement de ce travail.*

**MEGNOUCHE IMENE**

## **Dédicace**

*Je dédie ce travail :*

À

*Ma Mère*

***Mm: ZERARI Samia***

À

*Mon père*

***Mr : AMEYOUND Mourad***

À

*Mes deux frères*

***Soheib et Moncef***

À

*Mes amies*

***Imene , Aya , Hasna , Khadidja et Rania***

À

*Mes grands-mères*

***LEZOUM Baya et MADENE Kheira***

AMEYOUND MERIEM

SIRINE

## *Dédicace*

*Je dédie ce mémoire :*

*À mes chers parents, pour tous leurs sacrifices et leur prière tout au long de mes études et à mon frère **Zerrouk** et à son épouse **Maroua** et à ma nièce **Yousra** .*

*Je dédie ce travail à mon chère grand-père **KAMEL** qui nous a quittés cette année.*

*A **NAILI Mohamed Abdennacer** pour son soutien tout au long de mon parcours et d'être toujours là pour donner le meilleur de moi-même , et à **SHAMS DJAZAIR**.*

*Et à tous mes amis.*

MEGNOUCHE IMENE

## Sommaire

Résumé .....	I
Remerciement.....	II
Dédicace .....	IV
<b>Sommaire</b> .....	<b>VI</b>
Liste des figures .....	IX
Liste des tableaux .....	XI
Liste des abréviations .....	XII
Introduction générale.....	1
Chapitre I : Les MPPT.....	3
I.1 Introduction .....	4
I.2 Le générateur PV .....	5
I.3 Effet photovoltaïque .....	5
I.4 La cellule .....	5
I.5 Maximum Power Point Tracking (MPPT) .....	7
I.5.1 Les algorithmes MPPT .....	7
I.5.2 Commande de base : la P&O « classique » .....	7
I.5.3 L'intelligence artificielle et générateur photovoltaïque .....	12
I.6 Conclusion .....	12
Chapitre II : Les filtres numériques .....	13
II.1 Introduction .....	14
II.2 Définition.....	14
II.3 Représentation d'un filtre numérique .....	15
II.3.1 Fonction de transfert en z .....	15
II.3.2 Réponse impulsionnelle .....	16
II.3.3 Equation aux différences .....	16
II.4 Classification des filtres.....	16
II.4.1 Types de filtres et Gabarits.....	17
II.4.2 Les filtres à réponse impulsionnelle infinie (RII) .....	18
II.4.3 Les filtres à réponse impulsionnelle finie (RIF).....	19
II.5 Conclusion .....	22
Chapitre III : Amélioration d'un MPPT classique (P&O) par un sous-échantillonnage.....	23
III.1 Introduction : .....	24
III.2 MATLAB .....	24
III.2.1 Interface principale.....	25

III.2.2	Fichiers SCRIPT .....	26
III.2.3	Simulink .....	26
III.3	Implémentation d'une commande MPPT de base sur Matlab Simulink .....	26
III.3.1	Le panneau photovoltaïque .....	26
III.3.2	Le convertisseur élévateur.....	28
III.3.3	Contrôleur PWM .....	29
III.3.4	La commande P&O de base sur Matlab .....	30
III.3.5	Le schéma avec la commande P&O sans sous-échantillonnage sur Matlab Simulink.....	31
III.3.6	Résultats obtenus sans sous- échantillonnage .....	31
III.4	Amélioration de la commande P&O avec sous-échantillonnage .....	33
III.4.1	Down Sample (sous-échantillonnage).....	33
III.4.2	Le schéma avec la commande P&O avec un sous-échantillonnage sur Matlab Simulink.....	35
III.5	Numérisation de la commande P&O et implémentation avec un sous-échantillonnage 35	
III.5.1	La commande P&O numériser sur Matlab.....	35
III.5.2	Résultats obtenus avec sous-échantillonnage.....	36
III.6	Conclusion .....	39
Chapitre IV : Synthèse d'une commande P&O avec introduction du filtrage FIR sur FPGA		
	40	
IV.1	Introduction .....	41
IV.2	Définitions et Méthodes.....	41
IV.2.1	Les circuits programmables FPGAs.....	41
IV.2.2	Langage VHDL .....	42
IV.2.3	Xilinx vivado.....	43
IV.3	Implémentation d'une commande P&O avec un sous-échantillonnage sur FPGA... 44	
IV.3.1	Le schéma du système PV avec la commande P&O avec un sous- échantillonnage sur Simulink.....	44
IV.3.2	Le schéma de la commande P&O avec sous-échantillonnage sur Simulink.....	44
IV.3.3	La commande P&O sur xilinx vivado.....	45
IV.3.4	La fonction min_max .....	47
IV.4	Synthèse d'une commande P&O avec filtrage FIR sur FPGA.....	48
IV.4.1	Conception des coefficients et le filtre FIR.....	49
IV.5	Implémentation d'une commande P&O avec sous-échantillonnage et filtrage FIR sur FPGA 52	



IV.5.1	Le schéma du système PV avec la commande P&O avec sous-échantillonnage et filtrage FIR sur Simulink .....	52
IV.5.2	Le schéma de la commande P&O avec sous-échantillonnage et filtrage FIR sur Simulink.....	52
IV.6	Conclusion .....	53
Chapitre V :Validation et résultats.....		V-54
V.1	Introduction .....	55
V.2	Résultats de la commande P&O avec un sous-échantillonnage .....	55
V.2.1	Interprétation .....	57
V.2.2	Comparaison des résultats .....	57
V.3	Résultats de la commande P&O avec un sous-échantillonnage et filtrage FIR .....	58
V.3.1	Interprétation .....	60
V.3.2	Comparaison des résultats .....	60
V.4	Comparaison des résultats avec et sans filtre .....	61
V.5	Plateforme de validation sur circuit FPGA.....	62
V.5.1	Implémentation du MPPT sur FPGA .....	63
V.6	Conclusion .....	63
Conclusion générale .....		64
Références bibliographiques .....		65

# Liste des figures

## *Chapitre I: Le système photovoltaïque*

Figure I-1: Puissance des installations ajoutées par différentes sources d'énergie en 2020 ....	4
Figure I-2: Schéma synoptique d'un système PV connecté au réseau. ....	5
Figure I-3: Structure d'une cellule PV .....	6
Figure I-4: Caractéristique électrique d'une cellule photovoltaïque. R.....	6
Figure I-5: Schéma de principe d'une connexion à travers un convertisseur DC/DC contrôlé par une commande MPPT R. ....	8
Figure I-6: Principe de fonctionnement de la MPPT P&O R. ....	8
Figure I-7: Chaîne de conversion photovoltaïque avec acquisition et commande numérique R.	9
Figure I-8: Organigramme de l'algorithme P&O de base. ....	11

## *Chapitre II: Les filtres numériques*

Figure II-1 La structure générale d'un filtre numérique.....	15
Figure II-2: schéma générale d'un système par sa réponse impulsionnelle.....	16
Figure II-3: Réponse fréquentielles idéales des 4 filtres de base . ....	17
Figure II-4: Gabarit des filtres .....	18

## *Chapitre III :Amélioration d'un MPPT classique (P&O) par un sous-échantillonnage*

Figure III-1: Interface principale de Matlab.....	25
Figure III-2 :Modèle panneau photovoltaïque sur Simulink.....	26
Figure III-3:Caractéristiques du panneau. ....	27
Figure III-4: Variation de l'irradiation en fonction du temps (s). ....	27
Figure III-5: Modèle boost sur Simulink.....	28
Figure III-6: Paramètre du PWM. ....	29
Figure III-7: Le schéma avec la commande P&O sans sous-échantillonnage sur Matlab Simulink.....	31
Figure III-8: La variation de la tension (V) en fonction du temps (s) pour différent delta D. .	31
Figure III-9: La variation du courant (A) en fonction du temps (s) pour différent delta D. ....	32
Figure III-10: La variation de la puissance (Watt) en fonction du temps (s) pour différent delta D.....	32
Figure III-11:La variation du duty cycle en fonction du temps (s) pour différent delta D. ....	33
Figure III-12: Organigramme de l'algorithme P&O avec un sous échantillonnage.....	34
Figure III-13: Le schéma avec la commande P&O avec un sous-échantillonnage sur Matlab Simulink.....	35
Figure III-14: La variation du courant (A) en fonction du temps (s) pour différent delta D avec un sous échantillonnage. ....	37
Figure III-15:La variation de la tension (V) en fonction du temps (s) pour différent delta D avec un sous échantillonnage. ....	37
Figure III-16: La variation de la puissance (Watt) en fonction du temps (s) pour différent delta D avec un sous échantillonnage. ....	38

Figure III-17: La variation du duty cycle en fonction du temps (s) pour différent delta D avec un sous échantillonnage. ....	38
--	----

### ***Chapitre IV : Amélioration d'un MPPT classique (P&O) par un sous-échantillonnage***

Figure IV-1: Le schéma avec la commande P&O avec un sous-échantillonnage sur Matlab Simulink. ....	44
Figure IV-2: la commande P&O avec un sous-échantillonnage sur Simulink à l'aide de Xilinx. ....	44
Figure IV-3: Les paramètres de V_In .....	45
Figure IV-4: Les paramètres de I_In .....	45
Figure IV-5: Organigramme de l'algorithme P&O avec un sous échantillonnage et filtre. ....	48
Figure IV-6: Le schéma du système PV avec la commande P&O avec sous-échantillonnage et filtrage FIR sur Matlab Simulink .....	52
Figure IV-7: la commande P&O avec sous-échantillonnage et filtrage FIR sur Simulink à l'aide de Xilinx. ....	52

### ***Chapitre V : Validation et résultats***

Figure V-1: La variation du courant (A) en fonction du temps (s) pour différent delta D.....	55
Figure V-2: La variation de la tension (V) en fonction du temps (s) pour différent delta D. ..	56
Figure V-3: La variation de la puissance (Watt) en fonction du temps (s) pour différent delta D.....	56
Figure V-4: La variation du duty cycle en fonction du temps (s) pour différent delta D.....	57
Figure V-5: La variation du courant (A) en fonction du temps (s) pour différent fréquence de coupure.....	58
Figure V-6: La variation de la puissance (Watt) en fonction du temps (s) pour différent fréquence de coupure. ....	59
Figure V-7: La variation de la tension (V) en fonction du temps (s) pour différent fréquence de coupure.....	59
Figure V-8: La variation du duty cycle en fonction du temps (s) pour différent fréquence de coupure.....	60
Figure V-9: Variation de la puissance (Watt) en fonction du temps (s) avec et sans filtre.....	61
Figure V-10: Genesys 2 Kintex-7 FPGA .....	62
Figure V-11: Spartan-3E FPGA .....	63

## ***Liste des tableaux***

### ***Chapitre III :Amélioration d'un MPPT classique (P&O) par un sous-échantillonnage***

Tableau III-1: Les valeurs des composants du boost. ....	28
Tableau III-2: Les caractéristique du IGBT du boost. ....	29
Tableau III-3: Les caractéristique de la diode du boost. ....	29
Tableau III-4: Comparaison entre les rendements de puissance pour différent delta D.....	33
Tableau III-5: Comparaison entre les rendements de puissance avec un sous échantillonnage pour différent delta D.....	39

### ***Chapitre V :Validation et résultats***

Tableau V-1: Comparaison entre les rendements de puissance avec un sous échantillonnage pour différent delta D sur Xilinx.....	57
Tableau V-2: Comparaison entre les rendements de puissance avec un sous échantillonnage et filtre pour delta $D=20.10^{-4}$ .....	60
Tableau V-3: : Comparaison entre les rendements de puissance pour delta $D=20.10^{(-4)}$ .....	62
Tableau V-4: les valeurs choisies pour l'implémentation sur la carte .....	63

## *Liste des abréviations*

**PV** : Photo Voltaïque.

**MPPT** : Poursuite du **P**oint de **P**uissance **M**aximale (**M**aximum **P**ower **P**oint **T**racking).

**Dc** : **D**irect **C**urrent.

**Ac** : **A**lternatif **C**urrent.

**N** : **N**égative .

**P** : **P**ositive.

**PN** : La jonction .

**Cv** : **C**onstant **V**oltage .

**P&O** : **P**erturbe & **O**bserve.

**Gpv** : **G**énérateur **P**hoto**V**oltaïque.

**Vopt** : Tension **O**ptimale.

**Voc** : Tension de **C**ircuit **O**uvert.

**Ppm** : **P**oints de **P**uissance **M**aximale.

**Dc-Dc** : **C**ontinue-**C**ontinue.

**ADC** : **A**nalogue to **D**igital **C**onverter

**PWM** : **P**ulse **W**idth **M**odulation.

**VHDL** : **V**ery high-speed integrated circuit **H**ardware **D**escription **L**anguage.

**FPGA** : **F**ield **P**rogrammable **G**ates **A**rrays.

**ASIC** : **A**pplication **S**pecific **I**ntegrated **C**ircuits.

**V** : Tension.

**I**: Courant.

**P** : **P**uissance.

**V<sub>mp</sub>** : Tension du panneau à la puissance maximale.

**SE** : **S**ortie-**E**ntrées.

**ANN** : **A**rtificial **N**eural **N**etwork.

**EA** : Evolutionary Algorithms.

**FIR** : Finite Impulse Response.

**IIR** : Infinite Impulse Response.

**IGBT** : Insulated Gate Bipolar Transistor.

**C-IN** : Capacité –Entrée.

**R<sub>on</sub>** : Resistance

**L<sub>on</sub>**: Inductance

**V<sub>f</sub>** : Forward Voltage

**R<sub>s</sub>** : Snubber Resistance

**C<sub>s</sub>** : Snubber Capacitance

**IDE** :Integrated Design Environment

**ESL** :Electronic System Level

# Introduction générale

L'énergie étant le premier besoin de l'humanité, et face aux prévisions des ressources énergétiques fossiles mondiales, la science s'est tournée vers les énergies renouvelables et parmi ces sources d'énergie renouvelables la plus importante et avantageuse est l'énergie photovoltaïque, qui consiste à convertir le rayonnement électromagnétique en énergie électrique sous forme de courant électrique continu. Dans la plupart des systèmes de production d'énergie PV, il existe une technique ou un algorithme spécifique appelé "Maximum Power Point Tracking " (MPPT). Cette technologie a fait l'objet de nombreuses recherches pour améliorer les performances dynamiques des systèmes photovoltaïques, principalement en termes de capacité à accélérer le point de puissance global, améliorer et optimiser les performances des générateurs photovoltaïques.

L'objectif de notre travail est de développer une nouvelle approche pour le contrôleur MPPT ,moins complexe et plus performant et implémentable sur FPGA . Cette approche sera basée sur un MPPT classique accordé à un filtre FIR.

La difficulté actuelle est d'arriver à connaître les performances exactes de ces nouvelles commandes et d'en faire le choix. Un certain nombre de questions se rapportent à notre problématique, à savoir :

- Comment faire de la commande MPPT une commande performante et moins complexe ?

Ce mémoire intitulé «Contribution à l'implémentation FPGA du filtrage numérique FIR pour la commande MPPT» est planifié en cinq chapitres :

- Dans le premier chapitre nous présenterons les généralités sur le système Photovoltaïque ainsi que la commande MPPT par la méthode perturbe and observe.
- Le deuxième chapitre introduit les filtres numériques et leurs différents types, caractéristiques et structures, pour cela nous allons étudier les filtres à réponse impulsionnelle finie et infinie ainsi que les structures de leurs réalisations.
- Dans le troisième chapitre, un contrôleur de charge MPPT basé sur la technique P & O est conçu pour faire fonctionner un module PV fixe au MPP avec Matlab .
- Dans le quatrième chapitre nous traduisons la commande MPPT en langage VHDL puis on ajoute un filtre FIR ,le tout sera implémenté sur FPGA.

- Le dernier chapitre est consacré à la simulation et l'évaluation des résultats qui sont effectués pour la vérification du fonctionnement du panneau photovoltaïque à l'aide de Xilinx et Matlab Simulink.

Une conclusion générale a établi le bilan de cette étude .



# **Chapitre I : Les MPPT**

## I.1 Introduction

Le chapitre présent fournit un profil humble sur une approche nouvelle qui set à optimiser la performance d'un générateur photovoltaïque suivant la méthode de la poursuite du point de puissance maximum (MPPT).

L'énergie solaire constitue une alternative écologique aux énergies fossiles habituelles utilisées pratiquement dans tous les différents secteurs. Cette ressource est disponible en grande quantité à l'échelle internationale et cette énergie peut être récupérée et utilisée sous forme thermique ou électrique, quelle que soit la nature de son milieu : urbain, au contraire ou site isolé. En effet, son exploitation permet de récupérer jusqu'à  $1000\text{W}/\text{m}^2$  dans les zones tempérées [1].

L'augmentation de la capacité installée (de 10,3 %) dépasse les projections de long terme qui tablaient sur une croissance plus modeste d'une année sur l'autre. A la Fin de l'année 2020, la capacité mondiale de production d'énergie de sources renouvelables s'élevait à 2 799 GW, l'hydroélectricité se tenant toujours en tête (1 211 GW), bien que le solaire et l'éolien rattrapent rapidement leur retard. Les deux sources variables d'énergies renouvelables qui ont dominé l'expansion des capacités en 2020 sont les nouvelles installations solaires (127 GW) et éoliennes (111 GW) [2].

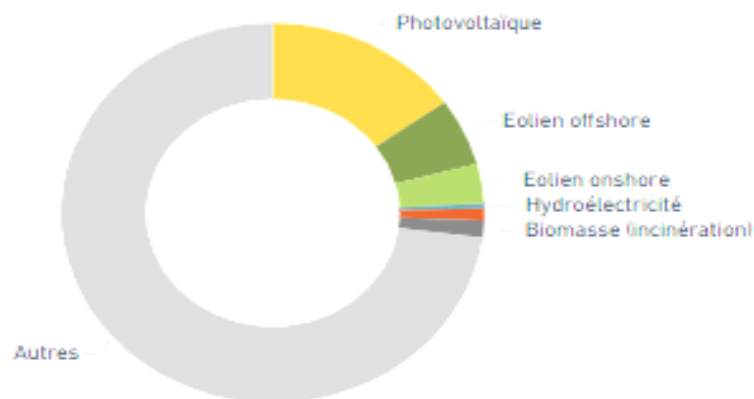


Figure I-1: Puissance des installations ajoutées par différentes sources d'énergie en 2020 [2].

Le contrôleur de poursuite de point de puissance maximale (Maximum Power Point Tracking (MPPT)) est conçu pour parer aux contraintes engendrées par différents changements climatiques. Dernièrement, la technique MPPT est devenue le centre d'intérêt d'un nombre important de recherches afin d'améliorer les performances dynamiques du

système PV, principalement en termes d'aptitude à poursuivre rapidement le point de puissance global.

## I.2 Le générateur PV

Le générateur PV est un ensemble d'équipements connectés pour exploiter l'énergie PV afin de satisfaire les besoins en charge. En fonction de la puissance désirée, les panneaux peuvent être assemblés pour constituer un "champ PV". Relié au récepteur sans autre élément, le panneau solaire fonctionne "au fil du soleil", c'est-à-dire que la puissance électrique fournie au récepteur est fonction de la puissance d'ensoleillement [2].

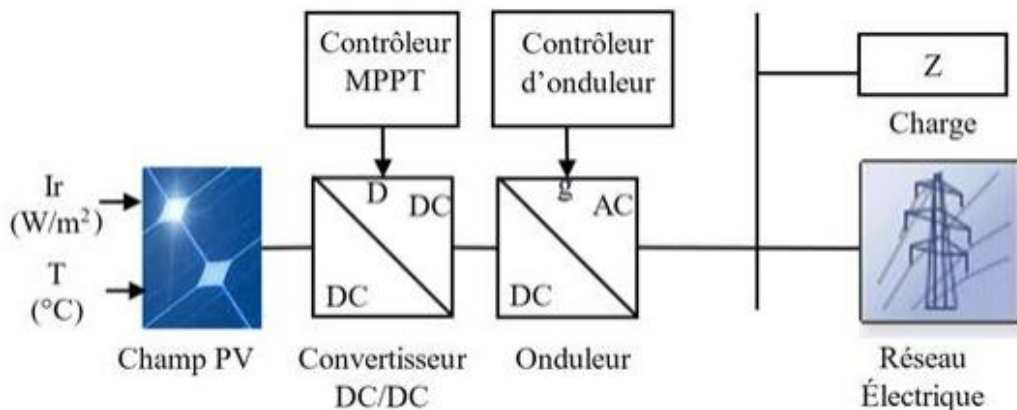


Figure I-2: Schéma synoptique d'un système PV connecté au réseau.

## I.3 Effet photovoltaïque

L'effet PV ou la conversion photovoltaïque est la transformation directe d'une énergie électromagnétique (rayonnement). Ce phénomène se produit lorsqu'un corps semi-conducteur est exposé à la lumière du soleil, ses atomes réagissent aux photons constituant la lumière.

A cet effet, les électrons des couches électroniques supérieures (appelés électrons des couches de valence) ont tendance à se décrocher. Dans les cellules PV, les électrons " décrochés " créent une tension électrique continue et faible [3].

## I.4 La cellule

Le fonctionnement de la cellule photovoltaïque est fondé sur les propriétés de semi-conducteurs qui, percutés par les photons, mettent en mouvement un flux d'électrons .

Les photons sont des particules élémentaires qui transportent l'énergie solaire et qu'Albert Einstein appelait dans les années 1920 les « grains de lumière ».

Lorsqu'ils frappent un élément semi-conducteur comme le silicium, ils arrachent des électrons à ses atomes. Ces électrons se mettent en mouvement, de façon désordonnée, à la recherche d'autres « trous » où se repositionner.

Mais pour qu'il y ait un courant électrique, il faut que ces mouvements d'électrons aillent tous dans le même sens. Pour les y aider, on va associer deux types de silicium. La face exposée au soleil est « dopée » avec des atomes de phosphore qui comportent plus d'électrons que le silicium, l'autre face est dopée avec des atomes de bore qui comportent moins d'électrons. Cette double face devient une sorte de pile : le côté très chargé en électrons devient la borne négative (N), le côté avec moins d'électrons devient la borne positive (P). Entre les deux il se crée un champ électrique.

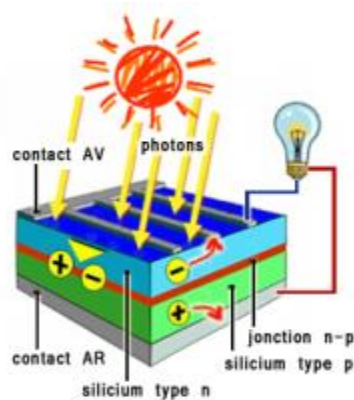


Figure I-3: Structure d'une cellule PV [5].

Le modèle équivalent d'une cellule est proche de celui d'une diode à jonction PN dont les caractéristiques dépendent du semi-conducteur, l'effet photovoltaïque pouvant être décrit par une simple source de courant proportionnelle au courant engendré par l'éclairement de la cellule [4].

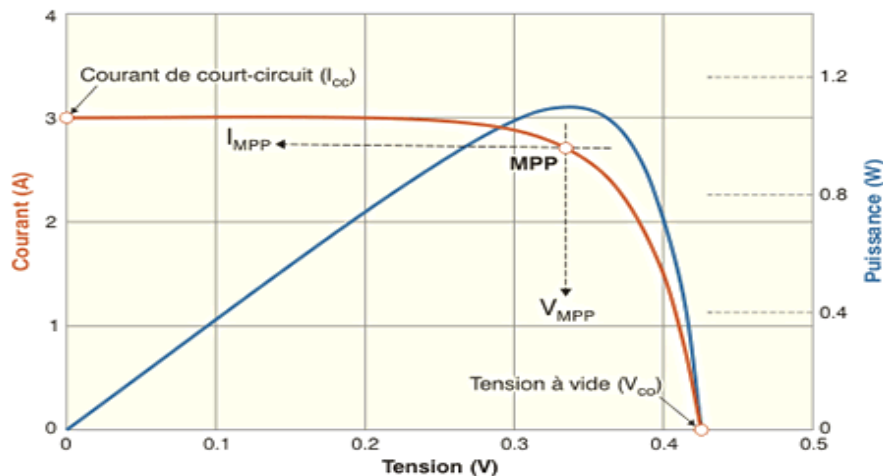


Figure I-4: Caractéristique électrique d'une cellule photovoltaïque [2].

## I.5 Maximum Power Point Tracking (MPPT)

C'est un montage électronique au niveau du régulateur qui permet de tirer le maximum d'énergie d'une installation solaire photovoltaïque.

Le régulateur **MPPT** permet d'optimiser le rendement des panneaux solaires jusqu'à **30%** : pour un panneau photovoltaïque donné, les batteries recevront 30% d'électricité en plus par rapport à un régulateur solaire classique.

### I.5.1 Les algorithmes MPPT

Il existe dans la littérature plusieurs algorithmes MPPT, les plus utilisés sont:

- Algorithme de la Tension Constante (CV, Constant Voltage).
- Algorithme du Courant constant (Constant Current).
- Perturbation et Observation (P&O, Perturb and Observe).
- Conductance Incrémentale (IncCond, Incremental Conductance).

Les algorithmes P&O et IncCond font partie de la technique appelée Hill Climbing (montée de pente).

### I.5.2 Commande de base : la P&O « classique »

La puissance maximale que peut fournir un GPV ainsi que sa tension optimale  $VOPT$  n'est pas connue à l'avance. Pour extraire le maximum de puissance du GPV, des lois des commandes permettant de trouver le point maximum existant. Appelées MPPT pour Maximum Power Point Tracking, elles permettent de contrôler le convertisseur de puissance afin qu'il assure la meilleure adaptation possible entre le GPV et sa charge [5].

Les commandes MPPT modélisées sont de type « Perturb and Observe ». Cinq algorithmes seront programmés : la « P&O » de Cédric Cabal, sa version améliorée « P&O adaptative », la « new P&O » de Youssef El-basri dont les informations se trouvent dans les thèses et publications [5].

La quatrième loi sera l'optimisation de la « New P&O », qui n'a pas encore été développée. La cinquième commande utilisera le principe de la fraction de VOC. Bien que cette commande soit simpliste, il sera intéressant de voir si elle peut apporter une solution pour certaines applications [5].

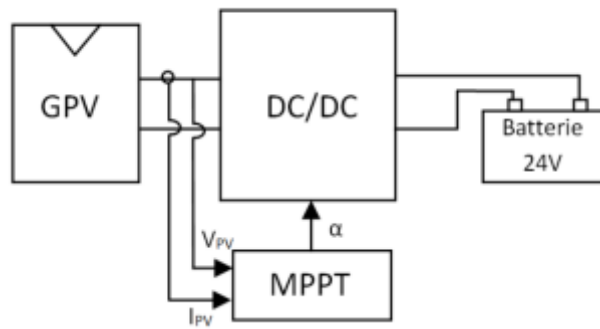


Figure I-5: Schéma de principe d'une connexion à travers un convertisseur DC/DC contrôlé par une commande MPPT [7].

Une des méthodes les plus couramment utilisées est la méthode nommée « Perturb and Observe » (P&O). Cette commande est de type extrémal, c'est-à-dire qu'il est possible de connaître l'évolution d'un paramètre ou d'une variable physique à l'aide d'une perturbation du paramètre d'entrée quand toutes les autres méthodes ne sont pas applicables en termes de prévisions [5].

La méthode P&O consiste à perturber la tension  $V_{PV}$  du  $G_{PV}$  avec l'ajout d'un signal de faible amplitude positif ou négatif, puis d'analyser la variation de puissance (noté  $dP$ ) correspondante à cette perturbation. Par conséquent, il est alors possible de savoir où se situe le MPP par rapport au point de fonctionnement actuel. Par exemple, si la perturbation est de nature à augmenter la tension  $V_{PV}$ , et que la puissance augmente également, c'est que le MPP se situe à droite du point de fonctionnement actuel. Cette information permet de déplacer le point de fonctionnement jusqu'au MPP, en contrôlant le rapport cyclique du convertisseur DC/DC. Une fois le MPP atteint, le point de fonctionnement oscille autour de celui-ci :

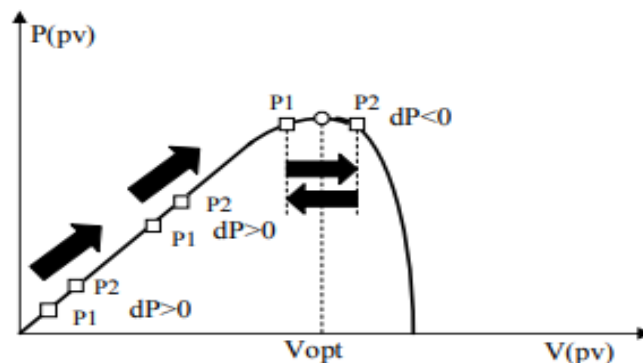


Figure I-6: Principe de fonctionnement de la MPPT P&O [5].

La commande doit également assurer un bon fonctionnement en dynamique. Lors d'un changement d'un/des paramètres météo (ensoleillement, température), le nouveau PPM doit être retrouvé le plus rapidement possible. Bien que la commande P&O puisse être réalisée de manière analogique, les commandes réalisées pendant ce mémoire sont de type numérique. La réalisation d'une commande numérique nécessite moins de composants, une dimension de carte réduite ainsi qu'une plus grande flexibilité par rapport à une commande analogique [5].

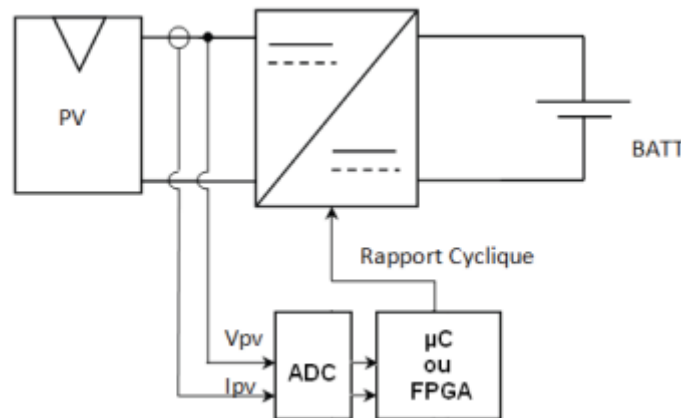


Figure I-7: Chaîne de conversion photovoltaïque avec acquisition et commande numérique [5].

Des convertisseurs analogiques numériques (ADC) sont intégrés dans la puce, ainsi que des sorties pouvant générer des signaux de type PWM à des fréquences suffisamment élevées pour les convertisseurs de puissance. Les progrès réalisés sur ces circuits permettent une faible consommation électrique pour réaliser la commande. La seconde possibilité est de se servir d'un FPGA. Programmé en VHDL, le FPGA permet de décrire le comportement du composant souhaité. Il est possible de réaliser plusieurs actions en parallèle, d'optimiser la largeur des bus pour une application spécifique, de contrôler chaque bit un à un. Bien que certains intègrent des ADC, la majorité des FPGA n'en sont pas dotées, ce qui oblige à rajouter un ou des composants externes supplémentaires pour réaliser l'acquisition des signaux analogiques [6].

### I.5.2.1 Le Fonctionnement de l'Algorithme P&O

D'abord la tension  $V$  et le courant  $I$  sont mesurés pour calculer la puissance  $P$ . Cette valeur  $P$  est comparée à la valeur de la puissance obtenue durant la dernière mesure .

- Si la puissance fournie par le panneau a augmenté depuis la dernière mesure, l'incrémentations ou décrémentations du rapport cyclique d continuera dans le même sens que lors du dernier cycle et ceci est fait par le test sur  $dV$ .

- Si  $dV > 0$  cela veut dire qu'on a incrémenté  $V$  durant le dernier cycle c'est-à-dire  $D_{i+1} = D_i + \Delta D$ .
- Si  $dV < 0$  cela veut dire qu'on a décrémenté  $V$  durant le dernier cycle c'est-à-dire  $D_{i+1} = D_i - \Delta D$ .

Donc on termine dans le chemin où  $P$  continue à augmenter. Si la puissance fournie par le panneau a diminué depuis la dernière mesure, l'incrémentation ou la décrémentation du rapport cyclique  $d$  sera en sens inverse par rapport au dernier cycle et ceci est fait aussi par le test sur  $dV$ . Avec cet algorithme la tension de fonctionnement  $V$  est perturbée à chaque cycle. Dès que le MPP sera atteint,  $V$  oscillera autour du point de fonctionnement idéal  $V_{mp}$ . Mais ceci cause des pertes de puissance qui dépendent de la largeur du pas d'une simple perturbation  $\Delta D$ . Si la largeur du pas est grande, l'algorithme de MPPT répondra rapidement aux changements soudains mais occasionnera des pertes dans des conditions changeantes lentement et dans les états stables. Si la largeur du pas est très petite les pertes dans les états stables où les conditions changeantes lentement seront réduites, mais le système aura une réponse lente aux changements rapides de la température ou d'insolation. La valeur pour la largeur idéale du pas du système ne peut être déterminée qu'expérimentalement ou par simulation, est ainsi satisfaire un compromis entre la réponse rapide et perte de puissance dans les états stables [6].



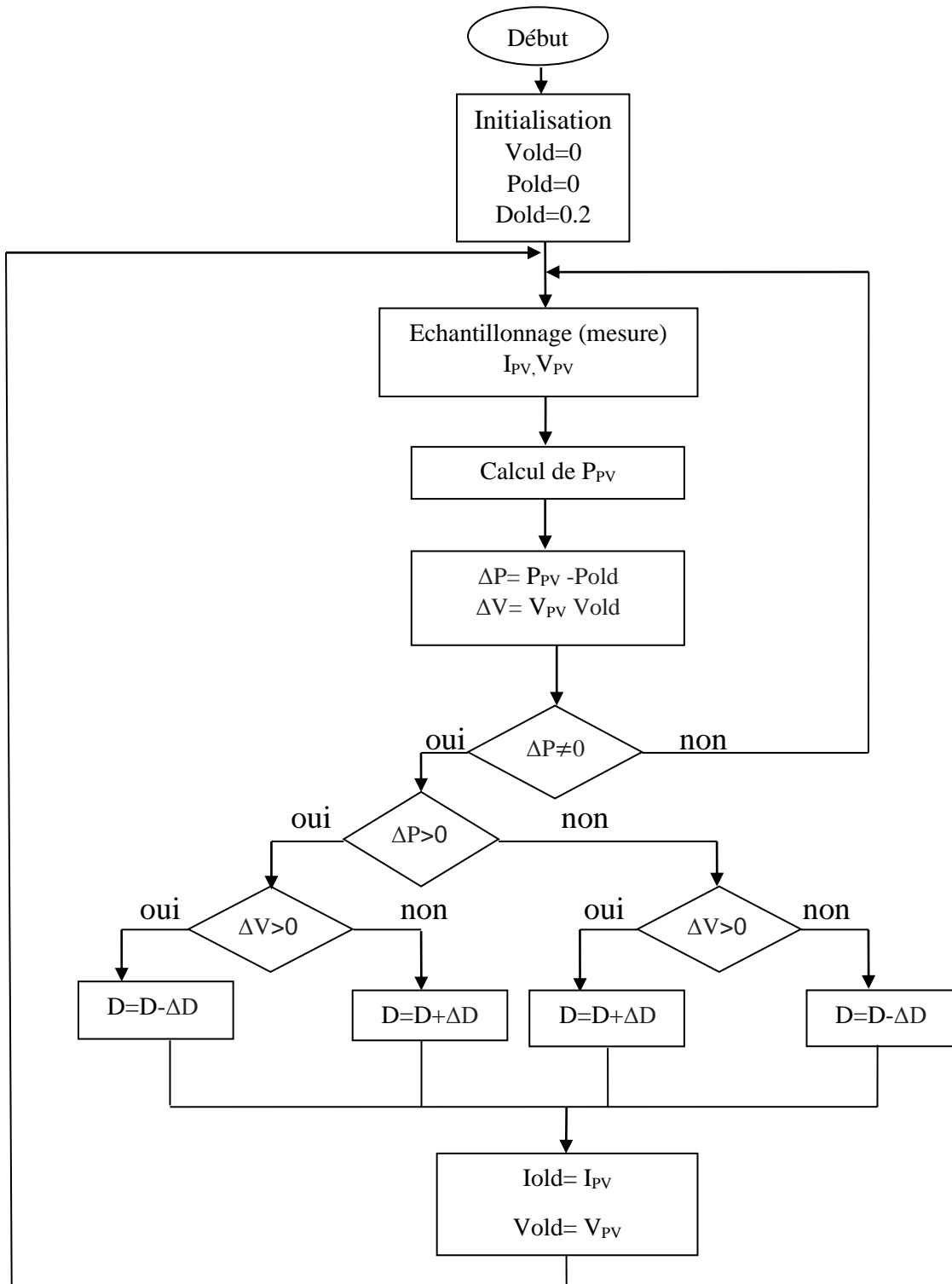


Figure I-8: Organigramme de l'algorithme P&O de base.

### **I.5.3 L'intelligence artificielle et générateur photovoltaïque**

Les techniques qui entrent dans cette catégorie comprennent, la logique floue, réseaux de neurones artificiels (ANN), et algorithmes évolutionnaires (EA).

#### **I.5.3.1 MPPT a la base des notions de logique flou**

La logique floue a fait son apparition en 1965 avec les travaux de L. Zadeh, ce dernier a formalisé la représentation et le traitement de connaissances imprécises ou approximatives afin de traiter des systèmes de grande complexité ou mal connu. La logique floue intervient dans la manipulation des connaissances imparfaites et elle est survenue comme une alternative efficace pour de tels systèmes populaires, les contrôleurs de logique floue ont les avantages de travailler avec entrées imprécises, n'ayant pas besoin d'un modèle mathématique [7].

#### **I.5.3.2 MPPT à base de réseaux de neurones artificiels**

Un neurone est une fonction non linéaire, paramétrée à valeurs bornées. Chaque neurone artificiel est un processeur élémentaire, il reçoit un nombre variable d'entrées en provenance de neurones amont ou à chacune de ces entrées est associée un poids  $w$ , Représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de Neurones aval [7].

Alors, pour estimer le MPPT, deux réseaux de neurones sont utilisés à la fois, le premier réseau dont le rôle est d'estimer le courant de sortie qui correspond à la puissance maximale, le deuxième est pour estimer la tension qui correspond à la puissance maximale aussi [7].

## **I.6 Conclusion**

La chaîne de conversion photovoltaïque se compose de plusieurs éléments mis bout à bout. Le module photovoltaïque, qui permet la conversion de la lumière en électricité, voit sa courbe de puissance changer suivant les conditions d'ensoleillement. Sa caractéristique non linéaire justifie l'utilisation d'un convertisseur DC/DC et de sa commande MPPT pour exploiter le maximum de la puissance disponible. Bien qu'ayant toutes le même objectif qui est d'optimiser le transfert de puissance, les différentes commandes MPPT utilisent des moyens différents pour y parvenir. La suite du mémoire sera consacrée à ces commandes proprement dites. Pour cela, tous les éléments de la chaîne seront modélisés et différentes commandes seront programmées dans les chapitres suivant.

## **Chapitre II : Les filtres numériques**

## II.1 Introduction

Le regain d'intérêt pour les filtres numériques a été causé par de nombreux facteurs, dont le plus important est l'utilisation accrue de matériel numérique, plutôt que de matériel analogique, pour le traitement des signaux. Ceci, à son tour, a été stimulé par les progrès de la technologie microélectronique, conduisant à la disponibilité de circuits numériques à faible coût et à grande vitesse. De plus, dans de nombreuses applications, les données à traiter sont fondamentalement discrètes, de sorte que le traitement numérique est la chose la plus naturelle à faire. Le filtrage numérique trouve des applications dans de nombreuses situations pratiques comme le traitement du signal radar et sonar, la communication vocale, le traitement des données géophysique .

Ce chapitre permet de présenter les filtres numériques et leurs différents types, caractéristiques et structures. Pour cela nous allons étudier les filtres à réponse impulsionnelle finie et infinie ainsi que les structures de leurs réalisations.

## II.2 Définition

On appelle « filtre numérique » un système utilisé pour modifier la distribution fréquentielle d'un signal numérique selon des spécifications données. Un filtre numérique peut être vu comme un procédé de calcul permettant de transformer un signal numérique d'entrée (séquence de nombres) en un signal numérique de sortie (seconde séquence de nombres) pour obtenir la modification voulue du signal. Le problème du filtrage numérique consiste donc à déterminer l'équation régissant cette transformation des signaux numériques qui d'une part doit représenter la réponse fréquentielle spécifiée et d'autre part peut être effectivement réalisé. La transformation peut être implantée sous forme de logiciel (algorithme) ou matérielle (circuits électroniques). Les filtres numériques sont, pour les signaux échantillonnés, les équivalents des filtres analogiques pour les signaux continus. En raison du développement des circuits intégrés rapide, les filtres numériques deviennent plus intéressants que les filtres analogiques en apportant de nombreux avantages : précision, fiabilité, stabilité, adaptabilité et facilité de commande. [8]

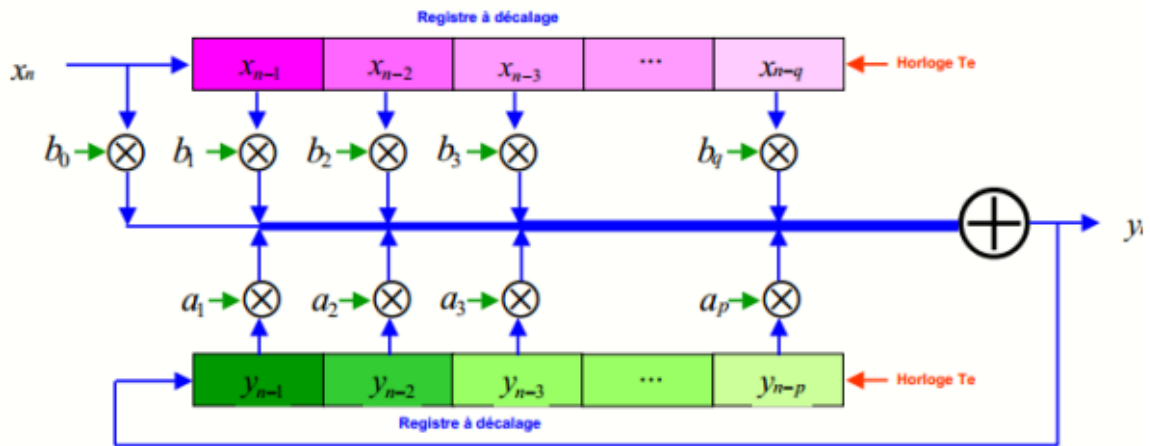


Figure II-1 La structure générale d'un filtre numérique

### II.3 Représentation d'un filtre numérique

Un filtre numérique accepte une série temporelle quantifiée  $\{x(kT)\}$  et produit une autre série temporelle quantifiée  $\{y(kT)\}$  par la solution d'une équation de différence linéaire[9].

Un filtrage numérique peut être représentée en utilisant plusieurs types de spécifications :

#### II.3.1 Fonction de transfert en z

Qui est un mode de représentation plus souvent usuel. Il permet de lier l'entrée et la sortie dans le plan z par :

$$H(z) = \frac{N(z)}{D(z)} = \frac{\sum_{i=0}^N b_i \cdot z^{-i}}{1 + \sum_{i=1}^N a_i \cdot z^{-i}} \quad \text{II-1}$$

**N(z):** polynôme numérateur de la fonction de transfert

**D(z) :** le dénominateur

**N :** l'ordre du filtre

Dans le cas où  $H(z)$  possède des pôles, on parlera de filtres RII (réponse impulsionnelle infinie). Si  $N(z)=1$ , on parlera de filtre tous-pôles. Dans le cas où  $D(z)=1$ , le filtre possède que des zéros. Cette famille de filtre correspond au cas des filtres RIF (réponse impulsionnelle finie). Celle-ci n'a pas d'équivalent en filtrage analogique et ses propriétés sont très utilisées en traitement du signal [10].

### II.3.2 Réponse impulsionnelle

La réponse impulsionnelle permet la représentation d'un système en fonction de son entrée et de sa sortie uniquement, par opposition à une représentation d'état.

Si les hypothèses de linéarité et d'invariance temporelle sont vérifiées, on peut caractériser le système par sa réponse impulsionnelle  $h(n)$ .

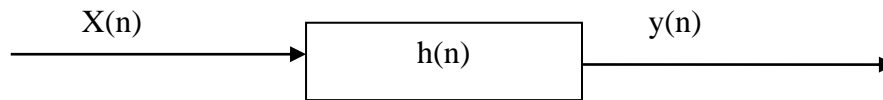


Figure II-2: schéma générale d'un système par sa réponse impulsionnelle

On peut en déduire l'effet d'une entrée quelconque sous la forme d'une convolution. Cette dernière est l'opération de traitement de signal la plus fondamentale. Elle indique que la valeur du signal de sortie à l'instant  $n$  est obtenue par la sommation (intégrale) pondérée des valeurs passées du signal d'excitation  $x(n)$ . La fonction de pondération est précisément la réponse impulsionnelle  $h(n)$  [10]:

$$y(n) = h(n) * x(n) \quad \text{II-2}$$

### II.3.3 Equation aux différences

Un filtre numérique peut être défini par une équation aux différences, c'est-à-dire l'opération mathématique du filtre dans le domaine temporel (discret) [9] :

$$y(n) = \sum_{i=0}^N b_i \cdot x(n-i) - \sum_{i=0}^N a_i \cdot y(n-i) \quad \text{II-3}$$

## II.4 Classification des filtres

Un filtre doit laisser passer certaines fréquences, alors qu'il doit en atténuer (voir éliminer) d'autres. Nous devons donc pouvoir représenter ces contraintes. Il y a quatre filtres de bases :

- Filtre passe bas : Le filtre passe bas laisse passer les fréquences qui sont au-dessous de la fréquence de coupure, et atténue les autres fréquences (voir figure II-2) [11].
- Filtre passe haut : Le filtre passe haut laisse passer les fréquences au-dessus de  $f_c$  (voir figure II-2) [11].

- Filtre passe bande : Le filtre passe bande contient un intervalle fréquentiel entre  $f_{c1}$  et  $f_{c2}$  qui s'appelle bande passante, il laisse passer les fréquences comprises entre  $f_{c1}$  et  $f_{c2}$  et atténue les autres fréquences (voir figure II-2) [11].
- Filtre coupe-bande : Le filtre coupe-bande est une combinaison entre le filtre passe bas et le filtre passe haut, il contient deux fréquences de coupure  $f_{c1}$  et  $f_{c2}$ , il atténue les fréquences comprises entre  $f_{c1}$  et  $f_{c2}$  et laisse passer les autres fréquences (voir figure II-2) [11].

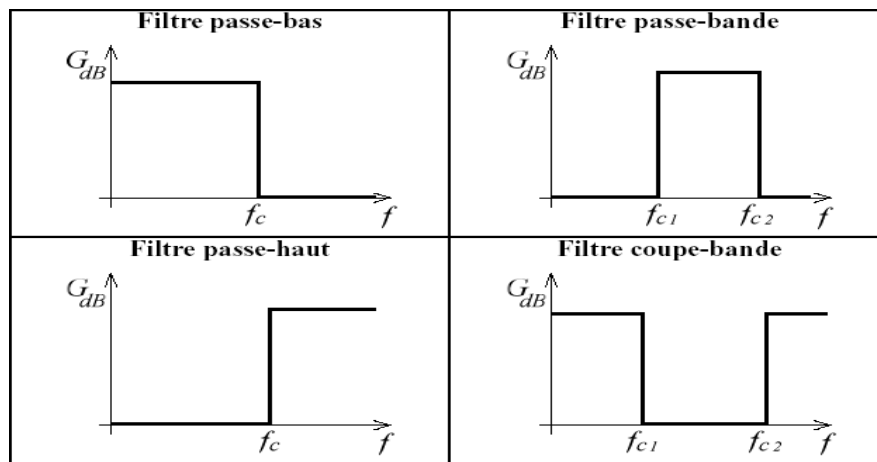


Figure II-3: Réponse fréquentielles idéales des 4 filtres de base [12].

### II.4.1 Types de filtres et Gabarits

Comme le filtre idéal n'est pas réalisable on spécifie un filtre à l'aide d'un gabarit qui donne les tolérances dans les différentes bandes de fréquences. On distingue trois bandes de fréquences différentes [10]:

- Bande passante :

C'est une zone de fréquence dans laquelle on souhaite un gain unitaire. Le gabarit en bande passante fixe une tolérance du gain qui doit être comprise dans l'intervalle.

- Bande atténuée :

C'est une zone de fréquence dans laquelle on souhaite un gain nul le gabarit en bande atténuée fixe une tolérance du gain qui doit être inférieur à une valeur maximum

- Bande de transition :

C'est une zone de fréquence entre une bande passante et une bande atténuée.

[10]

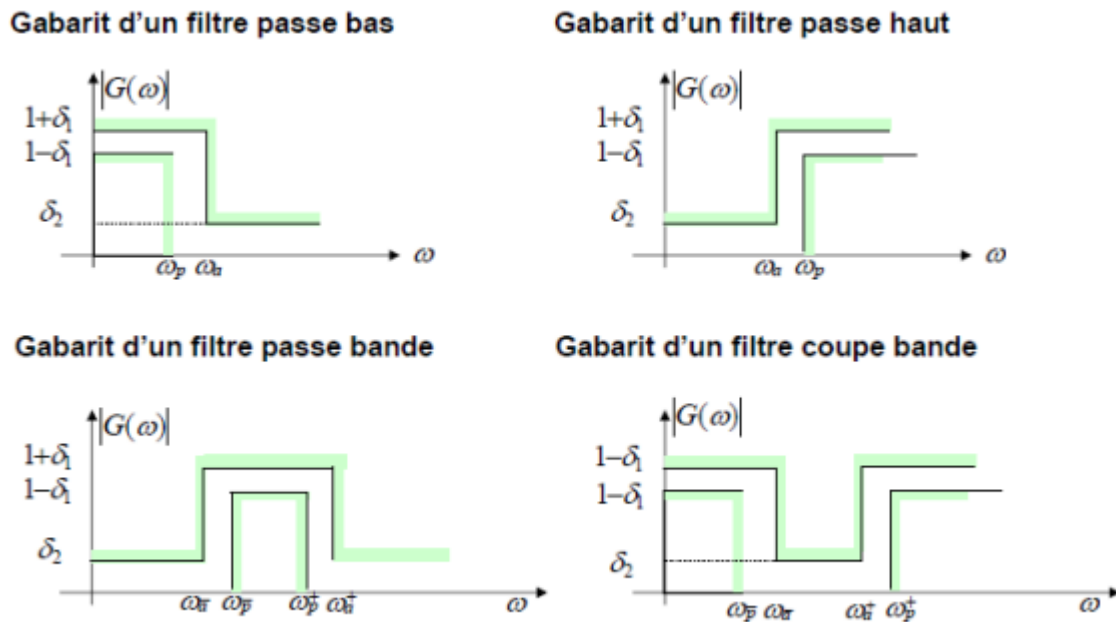


Figure II-4: Gabarit des filtres [10].

Outre la fonction du filtre (passe-haut, etc...) on peut aussi classer les filtres numériques en deux catégories selon la durée de la réponse impulsionnelle :

- Réponse impulsionnelle à durée finie (FIR).
- Réponse impulsionnelle à durée infinie (IIR).

## II.4.2 Les filtres à réponse impulsionnelle infinie (RII)

RII ou filtres récurrents la méthode la plus courante de conception de ces filtres est basée sur la conception du filtre analogique correspondant, suivie d'une transformation appropriée du plan  $s$  au plan  $z$ . La synthèse de filtres IIR en phase linéaire nécessite une approximation directe. [9] Cette catégorie de filtre est également caractérisée par et une équation aux différences récurrentes, c'est-à-dire lorsque la sortie  $y(n)$  dépend à la fois des entrées et des sorties précédentes. [10]

### II.4.2.1 Les principales caractéristiques des filtres RII sont

- Une bande de transition qui peut être étroite.
- Des méthodes de synthèse par transposition des méthodes pour les filtres analogiques.



- Une instabilité potentielle due à des pôles situés en dehors du cercle unité.
- Une instabilité numérique potentielle due au rebouclage.

### II.4.3 Les filtres à réponse impulsionnelle finie (RIF)

Les filtres numériques à réponse impulsionnelle finie (RIF) sont des systèmes linéaires discrets invariants dans le temps définis par une équation selon laquelle un nombre de sortie, représentant un échantillon du signal filtré, est obtenu par sommation pondérée d'un ensemble fini de nombre d'entrée, représentant les échantillons du signal à filtrer. Les coefficients de la sommation pondérée constituent la réponse impulsionnelle du filtre et un ensemble fini d'entre eux seulement prennent des valeurs non nulles. Ce filtre est du type « à mémoire finie », c'est-à-dire qu'il détermine sa sortie en fonction d'informations d'entrée d'ancienneté limitée. Il est fréquemment désigné par filtre non récursif, en raison de sa structure, car il ne nécessite pas de boucle de réaction dans sa réalisation, comme c'est le cas pour une autre catégorie de filtres, celle des filtres à réponse impulsionnelle infinie [13].

Les équations suivantes montrent la fonction de transfert en  $z$  et l'équation aux différences correspondantes de la forme générale d'un filtre RIF :

$$H(z) = \sum_{i=0}^{N-1} b_i \cdot z^{-i} \quad \text{II-4}$$

$$y(n) = \sum_{i=0}^{N-1} b_i \cdot x(n-i) = \sum_{i=0}^{N-1} h(i) \cdot x(n-i) \quad \text{II-5}$$

#### II.4.3.1 Les principales caractéristiques d'un filtre RIF

Les caractéristiques d'un filtre RIF sont [10] :

- Une bande de transition qui sera toujours plus large qu'un filtre RII ayant le même nombre de coefficients.
- Des méthodes de synthèse permettant de dériver n'importe quelle réponse fréquentielle.

- Une plus grande stabilité numérique que les filtres RII.
- Une plus grande facilité d'implantation dans un système numérique de traitement.
- Les méthodes de conception de filtre RIF simple met en œuvre un système numérique de traitement.
- Une phase qui peut être exactement linéaire.
- Pas d'équivalent analogique.

### II.4.3.2 Réalisation des filtres RIF

Les filtres numériques peuvent être réalisés à l'aide de trois éléments ou opérations de base. Soit l'élément gain, l'élément de sommation et le retard unitaire. Ces éléments sont suffisants pour réaliser tous les filtres numériques linéaires possibles [14].

### II.4.3.3 Synthèse des filtres RIF

A la différence des filtres RII, les filtres RIF ne sont réalisables que dans le domaine discret. Par conséquent, leurs méthodes de synthèse ne sont pas dérivées des filtres analogiques. On distingue trois principales méthodes, dont deux seront détaillées dans les sections suivantes [14]:

- La méthode des fenêtres.
- La méthode de l'échantillonnage en fréquence.
- Méthodes d'optimisation.

#### ► Méthode de la fenêtre

Cette technique consiste, connaissant l'expression analytique  $H(f)$  de la réponse fréquentielle continue (dont la formulation mathématique connue) à approcher, à déterminer par utilisation de la transformée de Fourier à temps discret inverse, la réponse impulsionnelle. Cette réponse temporelle non causale obtenue sera retardée pour la rendre causale [11].

A partir du gabarit idéal du filtre, on détermine les coefficients du filtre en limitant

le calcul à  $N$  valeurs réparties symétriquement autour de  $n = 0$ . Puis, on calcule de la TFD inverse du filtre idéal qui nous permettra de retrouver les échantillons de la réponse impulsionnelle soient les coefficients du filtre [11]:

$$h(n) = \begin{cases} \frac{1}{f_e} \int_{-f_e/2}^{f_e/2} h(f) e^{2\pi j f n T_e} df & N \text{ impair} \\ \frac{1}{f_e} \int_{-f_e/2}^{f_e/2} h(f) e^{2\pi j f (n-\frac{1}{2}) T_e} df & N \text{ pair} \end{cases} \quad \text{II-5}$$

Cette méthode produit une série infinie de coefficients, on limite, alors la réponse impulsionnelle à  $N$  échantillons (troncature). Sachant que la troncature induit des ondulations, on peut faire appel aux fenêtres de pondération pour les atténuer. Ainsi, la réponse impulsionnelle idéale  $h(n)$  sera multipliée par la fenêtre discrète  $\omega_N(n)$  de longueur  $N$  [11].

#### ► Méthode de l'échantillonnage fréquentiel

La méthode de synthèse par échantillonnage en fréquence est appliquée depuis la réponse fréquentielle d'un filtre continu idéal  $H(f)$  dont on ne connaît pas la formule mathématique (on ne peut alors calculer  $h(n)$  par TF inverse de  $H(f)$ ). On utilise alors la transformation de Fourier Discrète inverse. C'est-à-dire que l'on "échantillonne" la réponse désirée dans le domaine fréquentiel, on obtient  $N$  points de cette réponse fréquentielle auxquels on fait correspondre  $N$  points de la réponse temporelle équivalente obtenus par TFD inverse comme suit [11]:

On commence par échantillonner :

$$H(n) = \frac{1}{N} \sum_k^{(N-1)} H(k) e^{2\pi j k n / N} \quad \text{II-6}$$

$$k = -\frac{(N-1)}{2} \quad \text{II-7}$$

Puis on applique la TFD inverse :

Cette méthode de synthèse est très simple et permet de réaliser toute forme de filtre (chose qu'on ne peut réaliser avec la méthode précédente). Cependant, cette méthode de

synthèse ne garantit que les points fréquentiels  $H(k)$ . Entre ces points, la valeur de  $H(f)$  n'est pas maîtrisée, il peut y avoir des oscillations qui ne sont pas également réparties avec un maximum d'erreur entre la réponse idéale et la réponse obtenue se situant autour de la bande de transition. Pour obtenir la réponse en fréquence du filtre finalement obtenu, on peut par exemple appliquer une TFD à la réponse impulsionnelle  $h(n)$  de taille  $N$  obtenue, après avoir ajouté un grand nombre de zéros. Par ailleurs, du fait de l'emploi d'une TFD inverse sur  $N$  points, la réponse impulsionnelle  $h(n)$  obtenue est périodique de période  $N$  bien que la réponse impulsionnelle idéale souhaitée ne soit pas de durée limitée [11].

## **II.5 Conclusion**

Dans ce chapitre, nous avons donné un aperçu sur quelques types des filtres les plus répandus FIR et IIR, ainsi que leurs classifications qui est basé sur le passage des fréquences, nous avons sélectionné le filtre FIR pour continuer notre étude qu'on va l'implémenter dans un FPGA.

## **Chapitre III: Amélioration d'un MPPT classique (P&O) par un sous-échantillonnage**

### **III.1 Introduction :**

La puissance maximale que peut fournir un générateur PV ainsi que sa tension optimale  $V_{OPT}$  n'est pas connue à l'avance. Pour extraire le maximum de puissance du générateur PV, des lois de commandes, permettant de trouver le point maximum existant appelées MPPT pour Maximum Power Point Tracking, elles permettent de contrôler le convertisseur de puissance afin qu'il assure la meilleure adaptation possible entre le générateur PV et sa charge.

Dans ce chapitre, Nous présentons la procédure de conception d'un contrôleur de charge MPPT basé sur la technique P & O de base pour faire fonctionner un module PV avec Matlab, après nous allons améliorer cette méthode par un sous échantillonnage.

### **III.2 MATLAB**

A la fin des années 1970, les travaux du professeur de mathématiques Cleve Moler à l'Université du Nouveau-Mexique puis à Stanford ont abouti au développement d'un logiciel de calcul numérique nommé MATLAB qui signifie Matrix Labs, d'où MathWorks était chargé par sa commercialisation. [15] Ce logiciel est considéré à la fois comme un langage de programmation de quatrième génération et un environnement informatique numérique multi paradigme visant à aider les développeurs d'interagir avec des programmes développés dans différents langages et d'exploiter les atouts uniques de chacun de ces derniers à différentes fins. Parmi ses fonctionnalités : les calculs matriciels, le développement et l'exécution des algorithmes, la création des interfaces utilisateur (IU) et la visualisation des données.

L'interface de MATLAB inclut une fenêtre Type de console interactive pour l'exécution de commandes, ainsi que, le développement intégré (IDE) pour la programmation d'applications.

### III.2.1 Interface principale

Au lancement de Matlab, l'interface suivante apparaît :

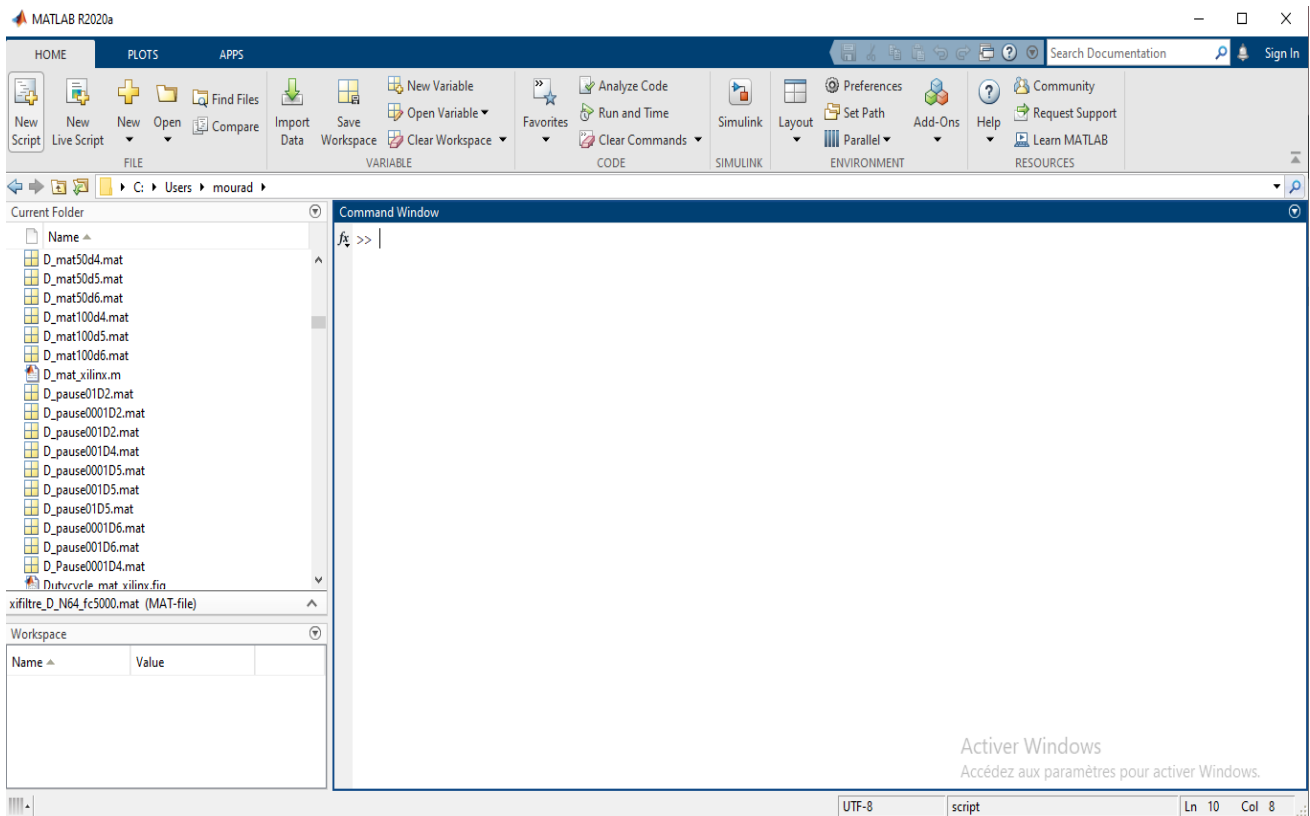


Figure III-1: Interface principale de Matlab.

### III.2.2 Fichiers SCRIPT

Un fichier script permet regrouper des séries de commandes Matlab. Cela évite d'avoir à saisir plusieurs fois de longues suites d'instructions. A son lancement, les instructions qu'il contient s'exécutent séquentiellement comme si elles étaient lancées depuis l'invite de commande. Un script stocke ses variables dans le workspace, lequel est partagé par tous les scripts. Ainsi, toutes les variables créées dans les scripts sont visibles depuis le command window et vice versa. Lorsque Matlab détecte une erreur, le programme s'arrête et un message d'erreur s'affiche à l'écran (avec le numéro de la ligne où l'erreur est détectée) [15].

### III.2.3 Simulink

Simulink est un environnement de diagramme de blocs utilisé pour concevoir des systèmes avec des modèles multi-domaines, simuler avant de passer au matériel et déployer sans écrire de code [15].

## III.3 Implémentation d'une commande MPPT de base sur Matlab Simulink

Le système se compose d'un panneau photovoltaïque et d'un convertisseur élévateur (boost) continu-continu connecté à une charge résistive et une commande MPPT.

### III.3.1 Le panneau photovoltaïque

Modèle panneau photovoltaïque a été choisi de Matlab Simulink ,module ( Sun Power SRP-315<sup>E</sup>-WHT-D) sous une température fix de 25 °C et irradiation variable :

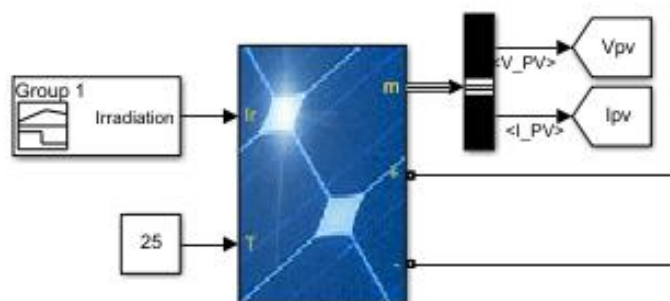


Figure III-2 :Modèle panneau photovoltaïque sur Simulink.



Parameters **Advanced**

Array data

Parallel strings

Series-connected modules per string

Module data

Module:

Maximum Power (W) <input type="text" value="315.072"/>	Cells per module (Ncell) <input type="text" value="96"/>
Open circuit voltage Voc (V) <input type="text" value="64.6"/>	Short-circuit current Isc (A) <input type="text" value="6.14"/>
Voltage at maximum power point Vmp (V) <input type="text" value="54.7"/>	Current at maximum power point Imp (A) <input type="text" value="5.76"/>
Temperature coefficient of Voc (%/deg.C) <input type="text" value="-0.27269"/>	Temperature coefficient of Isc (%/deg.C) <input type="text" value="0.061694"/>

Figure III-3:Caractéristiques du panneau.

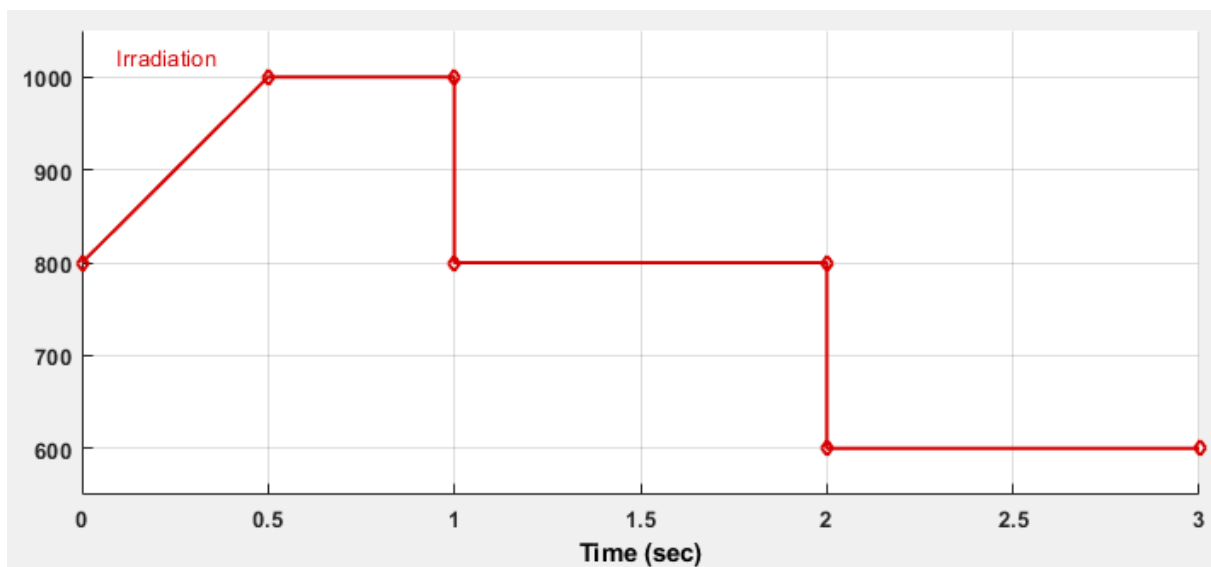


Figure III-4: Variation de l'irradiation en fonction du temps (s).

### III.3.2 Le convertisseur élévateur

Qui se compose des composants électriques suivants : condensateur d'entrée , inductance , interrupteur , diode , condensateur de sortie et une charge.

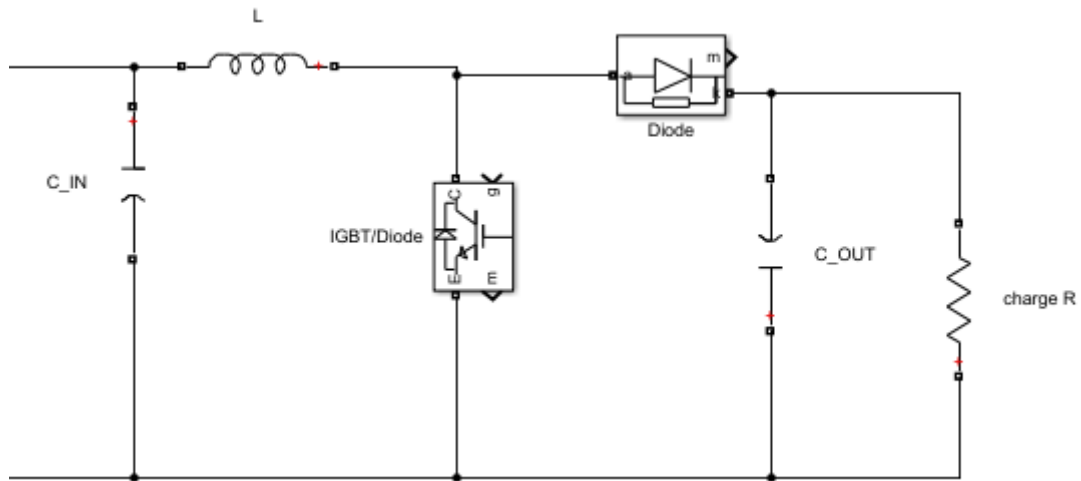


Figure III-5: Modèle boost sur Simulink.

Ce schéma réalisé sur Matlab Simulink. Ainsi que les valeurs des composants électroniques utilisées (Diode,IGBT ,Condensateur ,charge).

Tableau III-1: Les valeurs des composants du boost.

Composants	Valeurs
C_IN	$100 \cdot 10^{-6}(\text{F})$
L	$5 \cdot 10^{-3}(\text{H})$
C_OUT	$6 \cdot 10^{-4}(\text{F})$
Charge R	20(Ohms):

Tableau III-2: Les caractéristique du IGBT du boost.

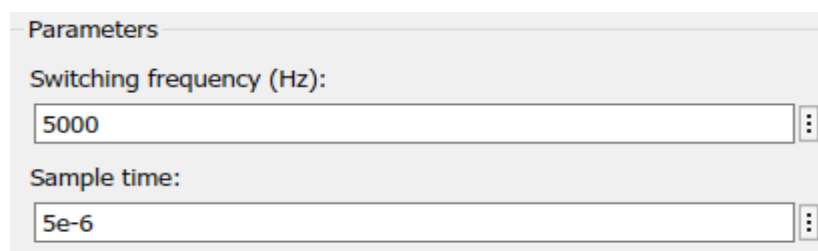
<b>R<sub>on</sub></b>	<b>1. 10<sup>-3</sup> (Ohms)</b>
<b>L<sub>on</sub></b>	<b>0 (H)</b>
<b>V<sub>f</sub></b>	<b>0.8 (V)</b>
<b>R<sub>s</sub></b>	<b>500(Ohms)</b>
<b>C<sub>s</sub></b>	<b>250. 10<sup>-9</sup> (F)</b>

Tableau III-3: Les caractéristique de la diode du boost.

<b>R<sub>on</sub></b>	<b>1. 10<sup>-3</sup> (Ohms)</b>
<b>R<sub>s</sub></b>	<b>1. 10<sup>-5</sup> (Ohms)</b>
<b>C<sub>s</sub></b>	<b>inf</b>

### III.3.3 Contrôleur PWM

Nous avons utilisé un régulateur de charge PWM selon les paramètres suivants :



The image shows a software interface for configuring PWM parameters. It has a title bar 'Parameters' and two input fields. The first field is labeled 'Switching frequency (Hz):' and contains the value '5000'. The second field is labeled 'Sample time:' and contains the value '5e-6'. Both fields have a small icon on the right side, likely for opening a unit or format menu.

Figure III-6: Paramètre du PWM.

### III.3.4 La commande P&O de base sur Matlab

```
function D = fcn(Ipv,Vpv)
Dinit=0.2;
Dmax=0.9;
Dmin=0.01;
DeltaD=20e-4;

Persistent Vold Pold Dold;
% dataType="double";
if isempty(Vold)
Vold=0;
Pold=0;
Dold=Dinit;
end
P=Ipv*Vpv;
DV=Vpv-Vold;
DP=P-Pold;

if DP~=0
if DP<0
    if DV<0
        D=Dold-DeltaD;
    else
        D=Dold+DeltaD;
    end
else
    if DV<0
        D=Dold+DeltaD;
    else
        D=Dold-DeltaD;
    end
end
else D=Dold;
end
if D>=Dmax | D<=Dmin
    D=Dold;
end
Dold=D;
Vold=Vpv;
Pold=P;
```

### III.3.5 Le schéma avec la commande P&O sans sous-échantillonnage sur Matlab Simulink

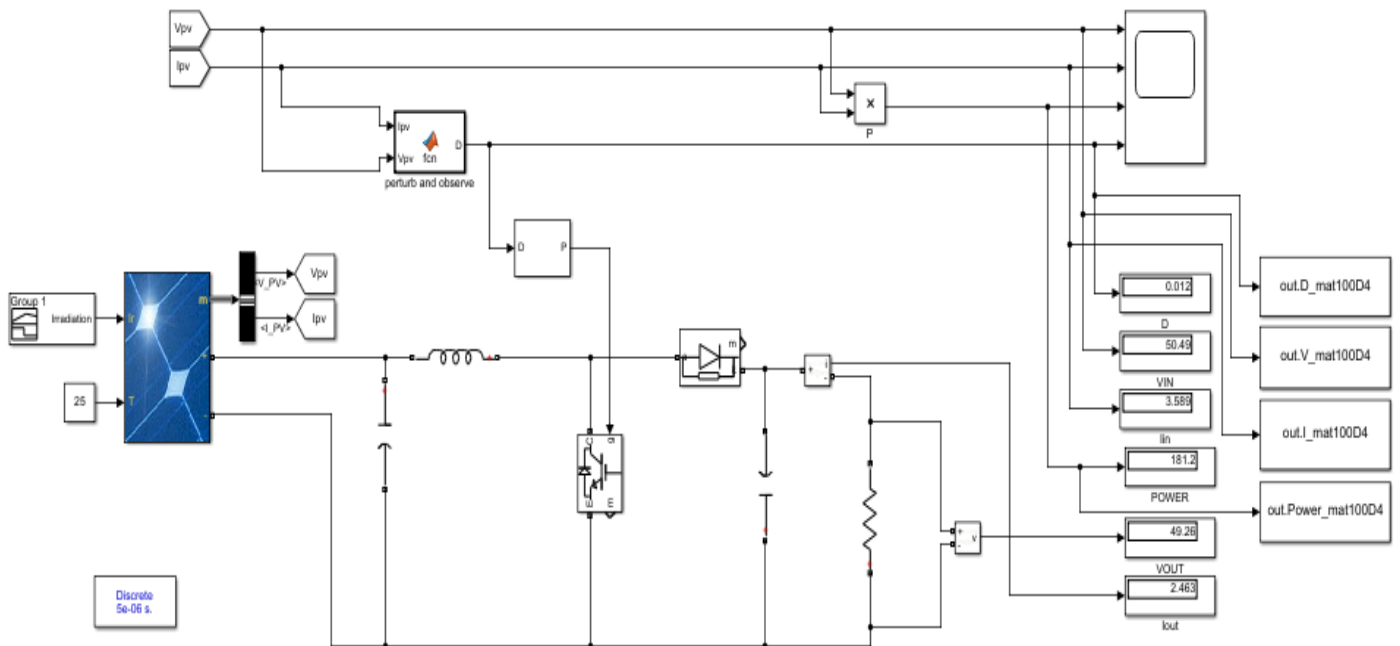


Figure III-7: Le schéma avec la commande P&O sans sous-échantillonnage sur Matlab Simulink.

### III.3.6 Résultats obtenus sans sous-échantillonnage

-Tension :

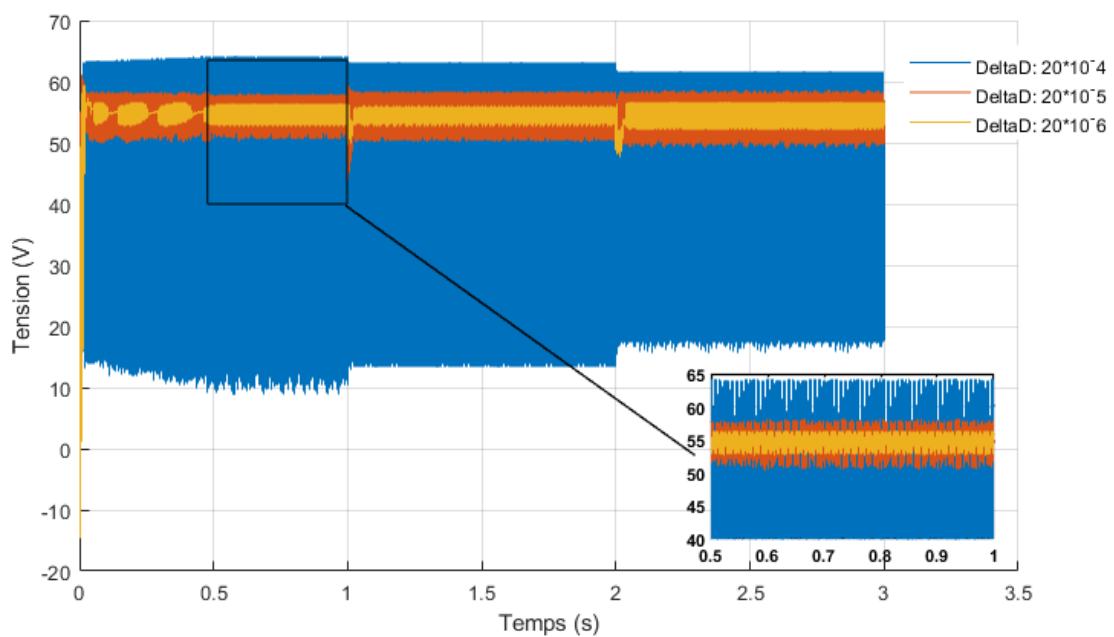


Figure III-8: La variation de la tension (V) en fonction du temps (s) pour différent delta D.

-Courant :

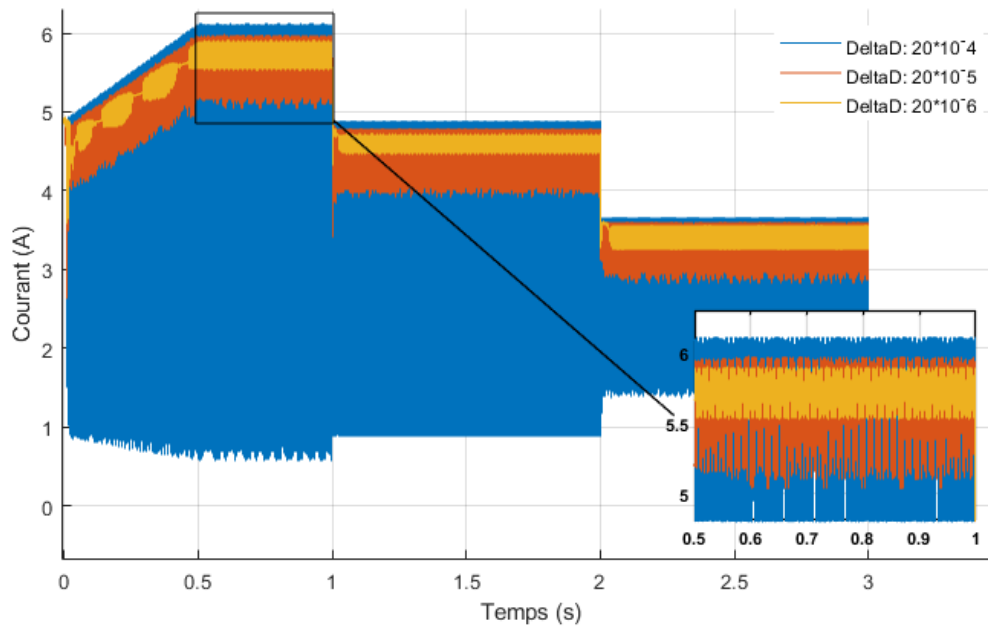


Figure III-9: La variation du courant (A) en fonction du temps (s) pour différent delta D.

-Puissance :

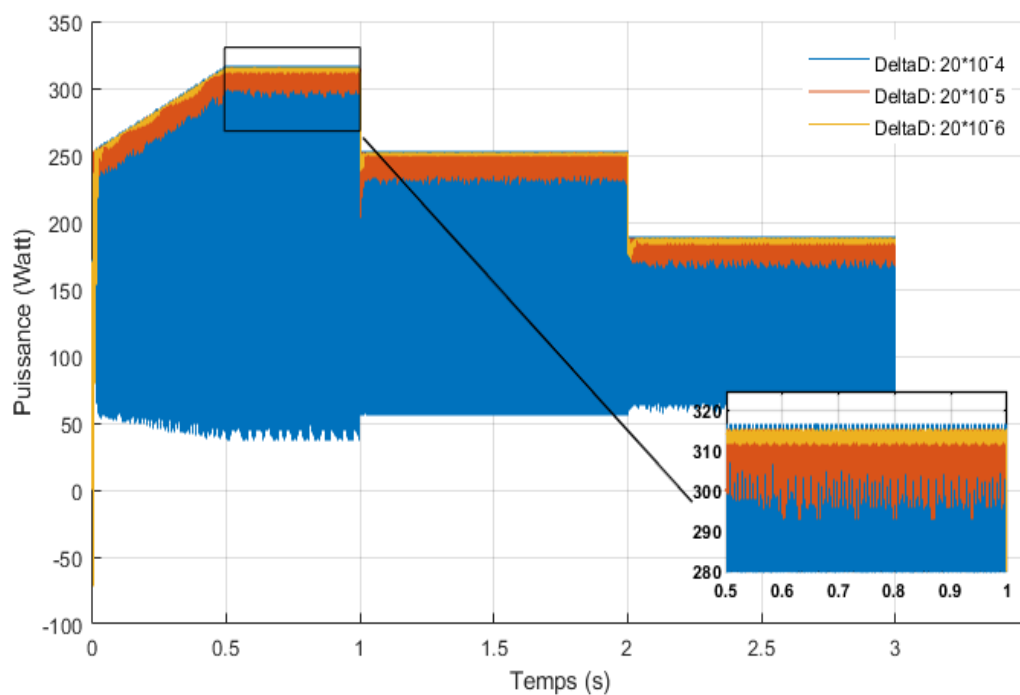


Figure III-10: La variation de la puissance (Watt) en fonction du temps (s) pour différent delta D.

-Duty cycle :

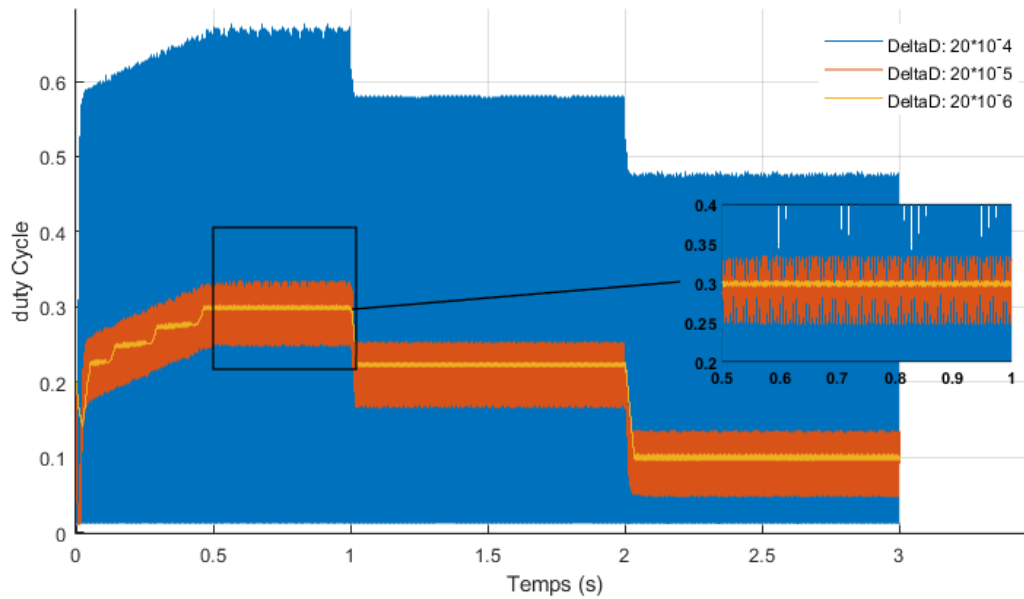


Figure III-11: La variation du duty cycle en fonction du temps (s) pour différent delta D.

### III.3.6.1 Interprétation

Les courbes de la tension, courant, puissance et le rapport cyclique obtenus sans sous-échantillonnage varient en fonction du temps (s), et nous remarquons que le changement de l'irradiation a un grand effet sur le résultat, tel que quand l'irradiation diminue les courbes de la tension, courant, puissance et le rapport cyclique diminuent et vis vers ça. Nous remarquons que dès le delta D augmente les ondulations augmentent aussi.

### III.3.6.2 Comparaison des résultats

Nous avons calculé les rendements de puissance pour différents delta D :

Tableau III-4: Comparaison entre les rendements de puissance pour différent delta D.

Delta D	$20 \cdot 10^{-4}$	$20 \cdot 10^{-5}$	$20 \cdot 10^{-6}$
$\eta$ (%)	55.2863	97.7334	99.4770

Nous remarquons que le rendement diminue avec l'augmentation de delta D, le rendement désiré pour un delta D =  $20 \cdot 10^{-6}$ .

## III.4 Amélioration de la commande P&O avec sous-échantillonnage

### III.4.1 Down Sample (sous-échantillonnage)

Dans le traitement du signal numérique, le sous-échantillonnage, la compression et la décimation sont des termes associés au processus de ré échantillonnage dans un système de

traitement du signal numérique à débits multiples. Le sous-échantillonnage et la décimation peuvent être synonymes de compression, ou ils peuvent décrire tout un processus de réduction de la bande passante (filtrage) et de réduction de la fréquence d'échantillonnage. Lorsque le procédé est effectué sur une séquence d'échantillons d'un signal ou d'une fonction continue, il produit une approximation de la séquence qui aurait été obtenue en échantillonnant le signal à un débit inférieur.

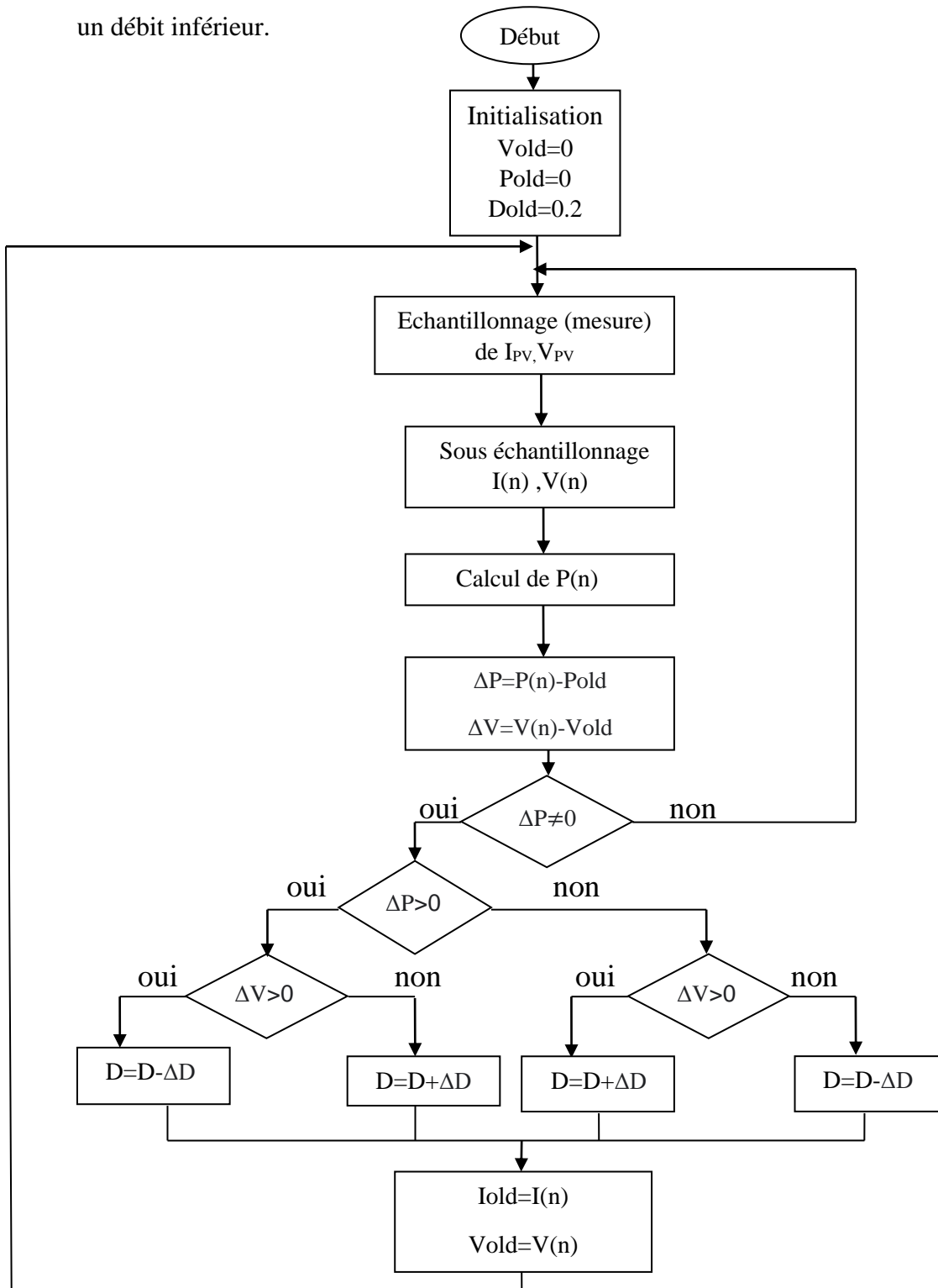


Figure III-12: Organigramme de l'algorithme P&O avec un sous échantillonnage.



### III.4.2 Le schéma avec la commande P&O avec un sous-échantillonnage sur Matlab Simulink

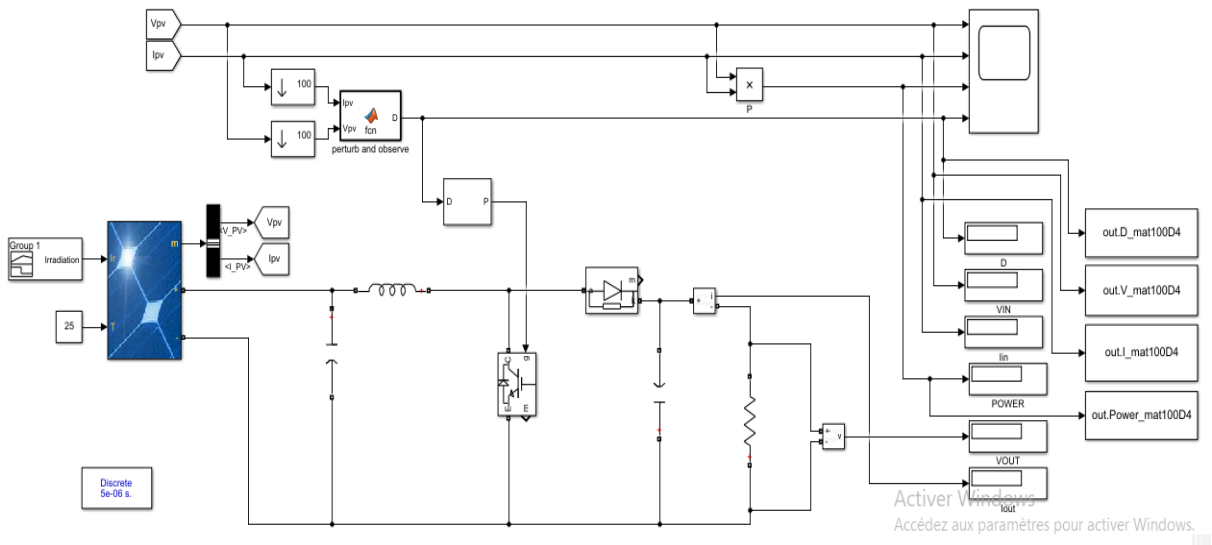


Figure III-13: Le schéma avec la commande P&O avec un sous-échantillonnage sur Matlab Simulink.

### III.5 Numérisation de la commande P&O et implémentation avec un sous-échantillonnage

Le but de la numérisation est de convertir un signal analogique qui contient une quantité infinie d'amplitudes en un signal numérique contenant lui une quantité finie de valeurs. La conversion de l'analogique au numérique consiste en deux étapes: l'échantillonnage et la conversion analogique-numérique (ADC).

Pour l'échantillonnage que nous avons fait précédemment, nous avons servi la fonction Floor dans Matlab pour limiter les valeurs.

#### III.5.1 La commande P&O numériser sur Matlab

```
function D = fcn(Ipv,Vpv)
Dinit=0.2;
Dmax=0.9;
Dmin=0.01;
DeltaD=20e-4;

persistent VoldPoldDold;
% dataType="double";
if isempty(Vold)
Vold=0;
Pold=0;
Dold=Dinit;
End
```

```
% numérisation

svp=sign(Vpv);
sip=sign(Ipv);
avp=abs(Vpv);
aip=abs(Ipv);
V1=avp*(2^6);
I1=aip*(2^9);
V2=svp*(2^-6)*floor(V1);
I2=sip*(2^-9)*floor(I1);
Vpv=V2;
Ipv=I2;

% Fin de numérisation

P=Ipv*Vpv;
DV=Vpv-Vold;
DP=P-Pold;

if DP~=0
if DP<0
if DV<0

                D=Dold-DeltaD;
else
                D=Dold+DeltaD;
end
else
if DV<0
                D=Dold+DeltaD;
else
                D=Dold-DeltaD;
end
end
else D=Dold;
end
if D>=Dmax | D<=Dmin
    D=Dold;
end
Dold=D;
Vold=Vpv;
Pold=P;
```

### III.5.2 Résultats obtenus avec sous-échantillonnage

Le sous-échantillonnage consiste à prélever des échantillons du signal pour obtenir un signal discret, dans le but de mémoriser, transmettre, et le traiter.

Avec des valeurs de down sample qui varient entre 5 , 50 et 100 .

-Courant

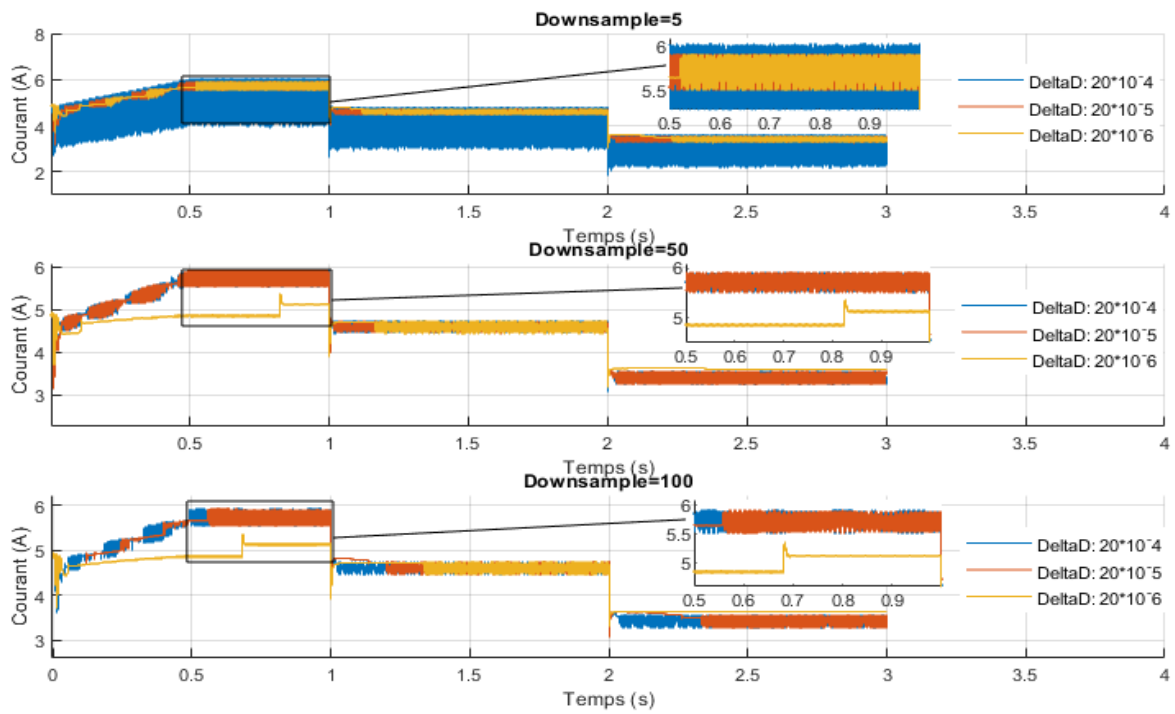


Figure III-14: La variation du courant (A) en fonction du temps (s) pour diffèrent delta D avec un sous échantillonnage.

-Tension

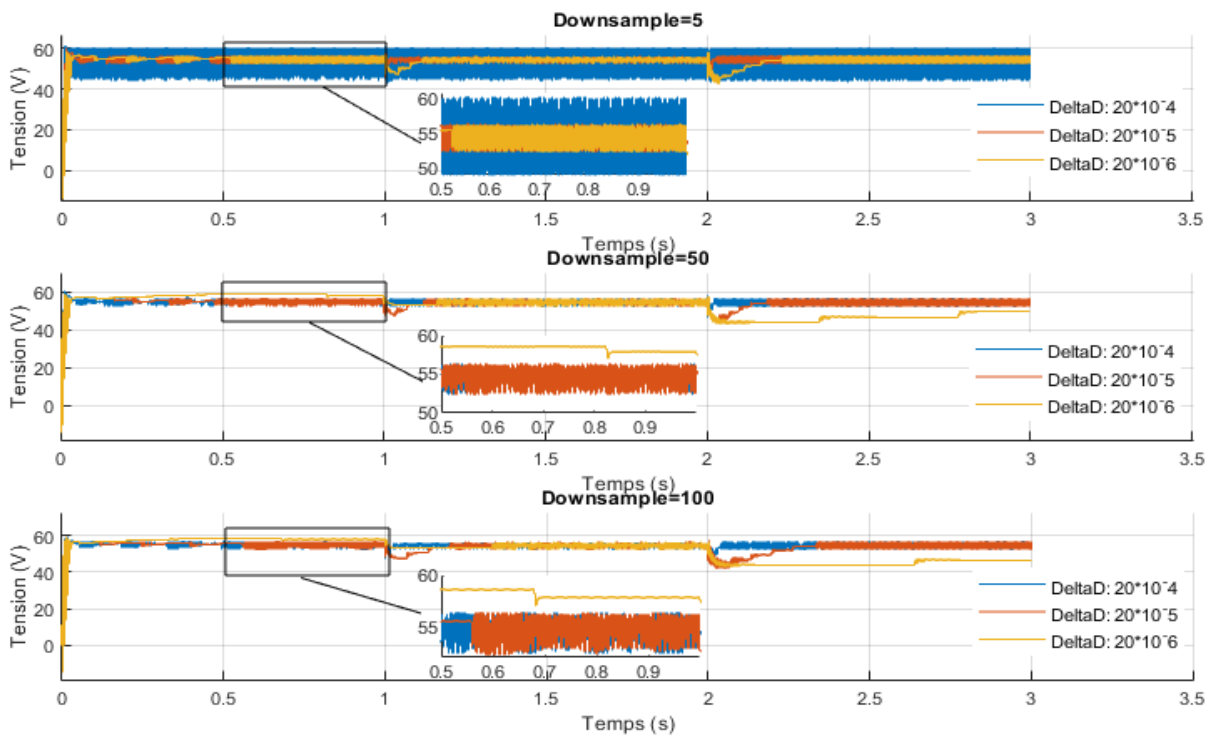


Figure III-15: La variation de la tension (V) en fonction du temps (s) pour diffèrent delta D avec un sous échantillonnage.

-Puissance

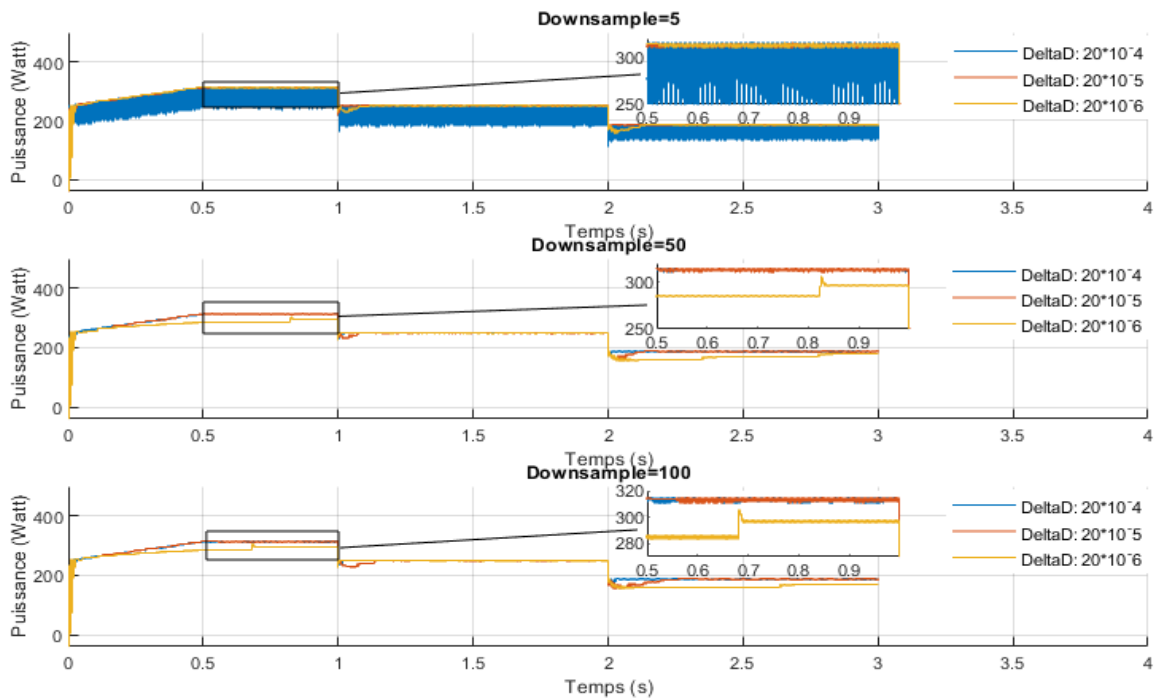


Figure III-16: La variation de la puissance (Watt) en fonction du temps (s) pour différent delta D avec un sous échantillonnage.

-Duty Cycle

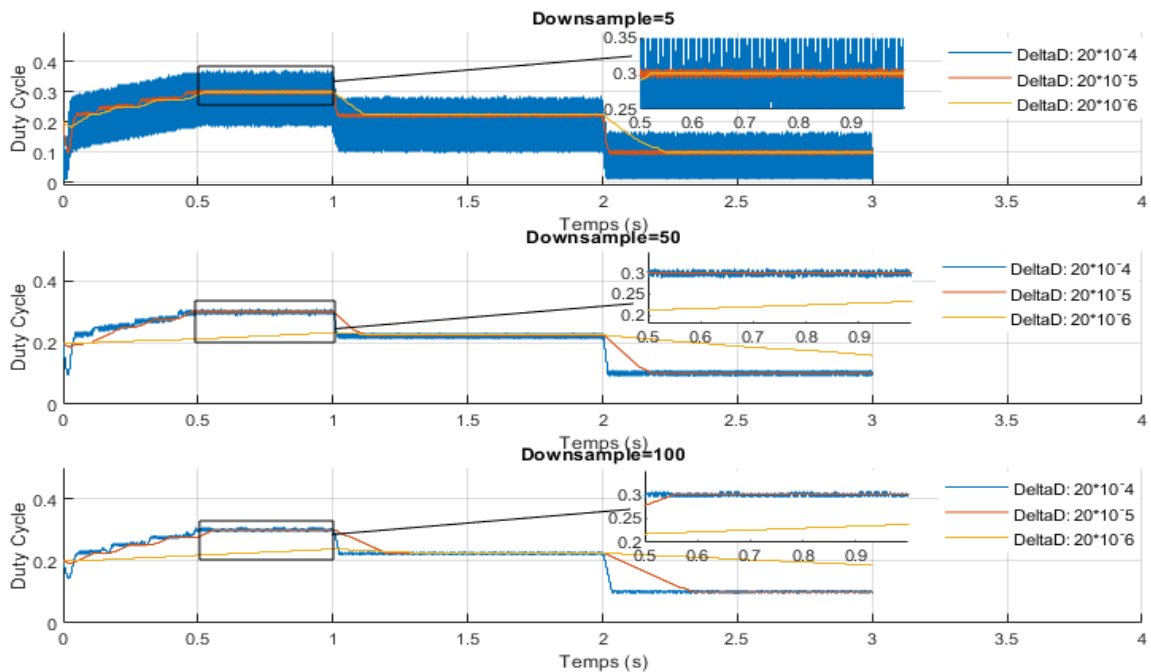


Figure III-17: La variation du duty cycle en fonction du temps (s) pour différent delta D avec un sous échantillonnage.

### III.5.2.1 Interprétation

Nous apercevons que les résultats de la tension, courant, puissance et le rapport cyclique obtenus avec sous-échantillonnage sont bien améliorés. Ces signaux sont compressés et traités, ainsi que les valeurs favorables ont été obtenue dans le cas où  $\text{down sample} = 100$ .

### III.5.2.2 Comparaison des résultats

Nous avons calculé les rendements de puissance pour différents delta D avec différentes valeurs de sous-échantillonnage :

Tableau III-5: Comparaison entre les rendements de puissance avec un sous-échantillonnage pour différents delta D.

Sous échantillonnage	100			50			5		
Delta D	$20 \cdot 10^{-4}$	$20 \cdot 10^{-5}$	$20 \cdot 10^{-6}$	$20 \cdot 10^{-4}$	$20 \cdot 10^{-5}$	$20 \cdot 10^{-6}$	$20 \cdot 10^{-4}$	$20 \cdot 10^{-5}$	$20 \cdot 10^{-6}$
$\eta$ (%)	99.4128	99.4465	92.8883	99.3903	99.3943	91.8069	90.6375	99.3340	99.3549

Nous remarquons que le sous-échantillonnage a un effet sur le rendement, quand le sous-échantillonnage égale à 100 et 50 le rendement désiré est obtenu à  $\text{delta D} = 20 \cdot 10^{-5}$  et lorsque le sous-échantillonnage égale à 5 le rendement désiré est obtenu à  $\text{delta D} = 20 \cdot 10^{-6}$ .

## III.6 Conclusion

Dans ce chapitre, nous avons présenté un système photovoltaïque composé d'un panneau PV, un convertisseur boost DC/DC (boost), et la commande MPPT par la méthode P&O. Dans un premier lieu nous avons simulé la commande de base sur Matlab Simulink puis nous avons ajouté un sous-échantillonnage et nous avons fait une numérisation de cette commande pour l'améliorer.

Les résultats de simulation obtenus montrent que lorsque le sous-échantillonnage égale à 100 la puissance à améliorer.

## **Chapitre IV : Synthèse d'une commande P&O avec introduction du filtrage FIR sur FPGA**

## **IV.1 Introduction**

Avec la révolution technologique du 20ème siècle et l'implantation de l'électronique dans tous les domaines de la vie, le développement des circuits électroniques a été remarquable, et pour des applications très spécifiques, le besoin de circuits puissants et spécialisés est apparu. Un nombre croissant de ces applications nécessitent que ces circuits soient programmables, ce qui a abouti à une famille de circuits programmables tels que les FPGA.

Dans ce chapitre nous traduisons la commande MPPT améliorée en langage VHDL puis on ajoute un filtre FIR.

## **IV.2 Définitions et Méthodes**

### **IV.2.1 Les circuits programmables FPGAs**

Les FPGA (Field Programmable Gate Arrays ou "réseaux logiques programmables") sont des circuits à architecture programmable et configurables. C'est un circuit intégré qui est (re)programmé par l'utilisateur dans un langage informatique spécifique VHDL ou Verilog après fabrication, il n'y a donc pas besoin de modifier le matériel. Ainsi, les FPGA se distinguent des ASIC (circuits intégrés spécifiques à l'application), qui sont des circuits intégrés spécifiques à l'application qui ne peuvent pas être reprogrammés une fois construits. Les FPGA sont puissants, rapides et reprogrammables, ce qui permet d'économiser du temps et de l'argent et ils offrent la possibilité de réaliser des fonctions numériques plus ou moins complexes. Contrairement aux microprocesseurs traditionnels, les FPGA peuvent effectuer plusieurs opérations en parallèle. Le premier appareil reprogrammable a été fabriqué par Altera en 1984. L'année suivante, les co-fondateurs de Xilinx inventent le premier FPGA. Depuis les années 1990, les FPGA sont de plus en plus utilisés, notamment par de grandes entreprises comme Microsoft. Ils offrent la possibilité de réaliser des fonctions numériques plus ou moins complexes [16].

#### **IV.2.1.1 Avantages et inconvénients des FPGA**

Les avantages et les inconvénients des FPGA sont multiples dont on cite :

- Les Avantages
  - Technologie « facile » à maîtriser.
  - Reprogrammable.
  - Temps de développement réduit.

- Idéal pour le prototypage (rapide).
  - Coût peu élevé.
  - Parallélisme de traitement.
  - Flexibilité et la possibilité de réduire
  - Fortement les délais de développement et commercialisation.
  - La reconfiguration, parfois en temps réel.
- Les Inconvénients
    - Performances non optimisées.
    - Temps de réponse long par rapport aux ASIC

## IV.2.2 Langage VHDL

VHDL signifie Very high-speed integrated circuit Hardware Description Language. Le langage a été développé à partir de 1983 ; il s'agissait au départ d'un projet du ministère de la défense des États-Unis, puis, il a été normalisé par l'IEEE en 1987. Le langage a été mis à jour en 1993, 2000, 2002 et 2008. Les modifications apportées en 2000 et 2002 sont mineures, en revanche, celles de 1993 et 2008 sont importantes. Comme celle de 2008 qui n'est pas encore entièrement supportée par les outils, nous présentons ici le VHDL-1993 [17].

### IV.2.2.1 Simulation et synthèse

Le but d'un langage de description matériel tel que le VHDL est de faciliter le développement d'un circuit numérique en fournissant une méthode rigoureuse de description du fonctionnement et de l'architecture du circuit désiré. L'idée est de ne pas avoir à réaliser (fondre) un composant réel, en utilisant à la place des outils de développement permettant de vérifier le fonctionnement attendu. Ce langage permet en effet d'utiliser des simulateurs, dont le rôle est de tester le fonctionnement décrit par le concepteur [18].

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
use IEEE.STD_LOGIC_arith.ALL;  
use IEEE.std_logic_unsigned.all;
```



Déclaration des bibliothèques

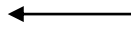
```
entity pando2 is  
  Port (  
  
  );  
end pando2;
```



Déclaration de l'entité :  
entrées/sorties du composant



```
architecture Behavioral of pando2 is  
  
begin  
  
end Behavioral;
```



Déclaration de l'architecture :  
Fonctionnement du composant

La syntaxe du VHDL est tirée du langage Ada, dont les mots clefs ont été adaptés à la conception matérielle. L'une des particularités du VHDL provient du fait qu'il est possible d'exprimer facilement le parallélisme présent à l'intérieur d'un circuit.

En VHDL, tout composant (dans le sens logiciel) est décrit sous deux aspects :

- L'interface avec le monde extérieur, décrite dans une section dénommée entité.
- L'implémentation elle-même, décrite dans une section dénommée architecture.

C'est donc l'ARCHITECTURE qui contient la description de la fonction matérielle désirée :

- Soit sous forme de description structurelle précise de l'architecture matérielle (les portes logiques à utiliser et leurs interconnexions),
- Soit sous forme de comportement attendu, c'est-à-dire orienté fonctionnel.

[18]

### IV.2.3 Xilinx vivado

Xilinx, était une société américaine de technologie et de semi-conducteurs qui fournissait principalement des dispositifs logiques programmables. La société était connue pour avoir inventé le premier réseau de portes programmables sur le terrain (FPGA) commercialement viable et créé le premier modèle de fabrication sans usine.

La principale boîte à outils de conception que Xilinx fournit aux ingénieurs est la suite de conception Vivado, un environnement de conception intégré (IDE ) avec des outils de niveau système à circuit intégré basés sur un modèle de données évolutif partagé et un environnement de débogage commun. Vivado comprend des outils de conception au niveau du système électronique (ESL) pour la synthèse et la vérification de la propriété intellectuelle algorithmique [18].

## IV.3 Implémentation d'une commande P&O avec un sous-échantillonnage sur FPGA

### IV.3.1 Le schéma du système PV avec la commande P&O avec un sous-échantillonnage sur Simulink

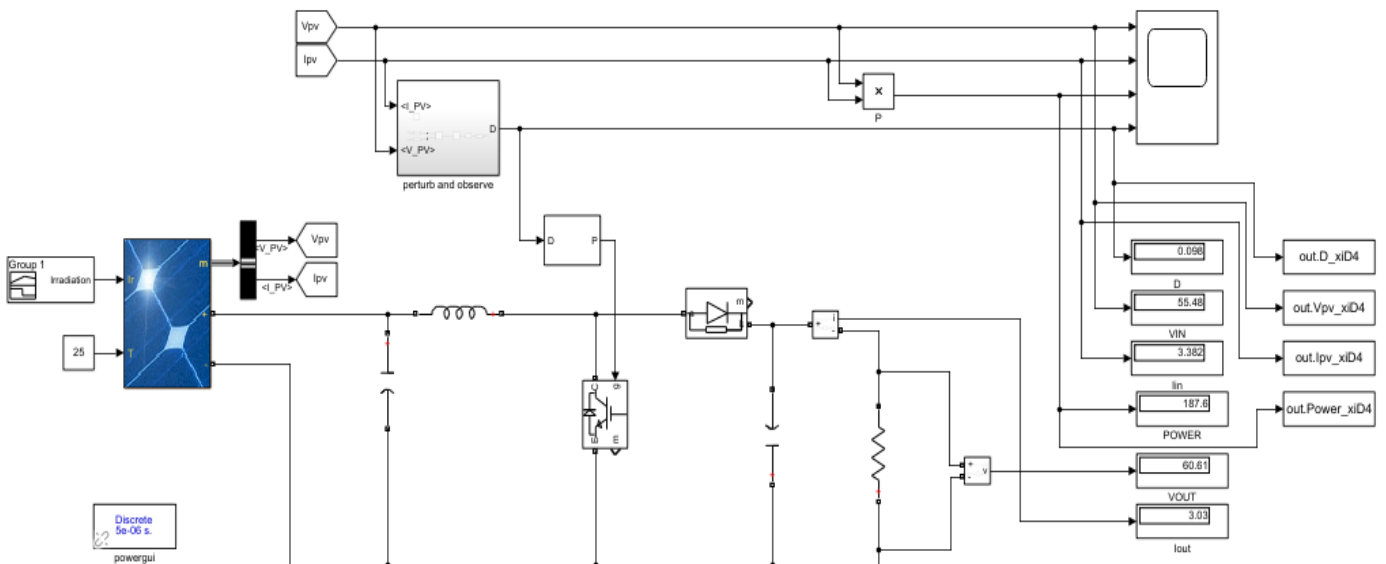


Figure IV-1: Le schéma avec la commande P&O avec un sous-échantillonnage sur Matlab Simulink.

### IV.3.2 Le schéma de la commande P&O avec sous-échantillonnage sur Simulink

Nous avons ajouté une fonction min\_max qui limite le rapport cyclique pour avoir des résultats plus précis :

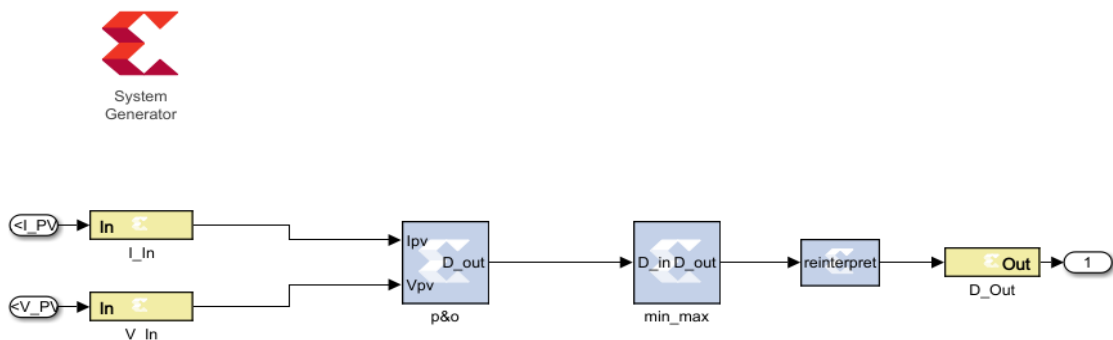


Figure IV-2: la commande P&O avec un sous-échantillonnage sur Simulink à l'aide de Xilinx.

Les paramètres de V\_In et I\_In sont mentionnés dans les figures IV-3 et IV-4 :

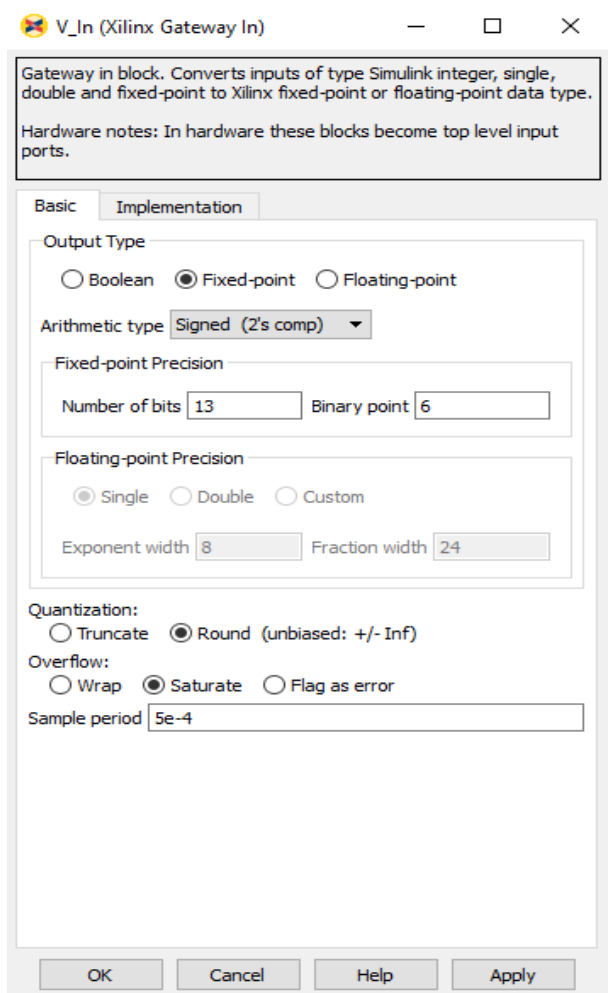


Figure IV-3: Les paramètres de V\_In

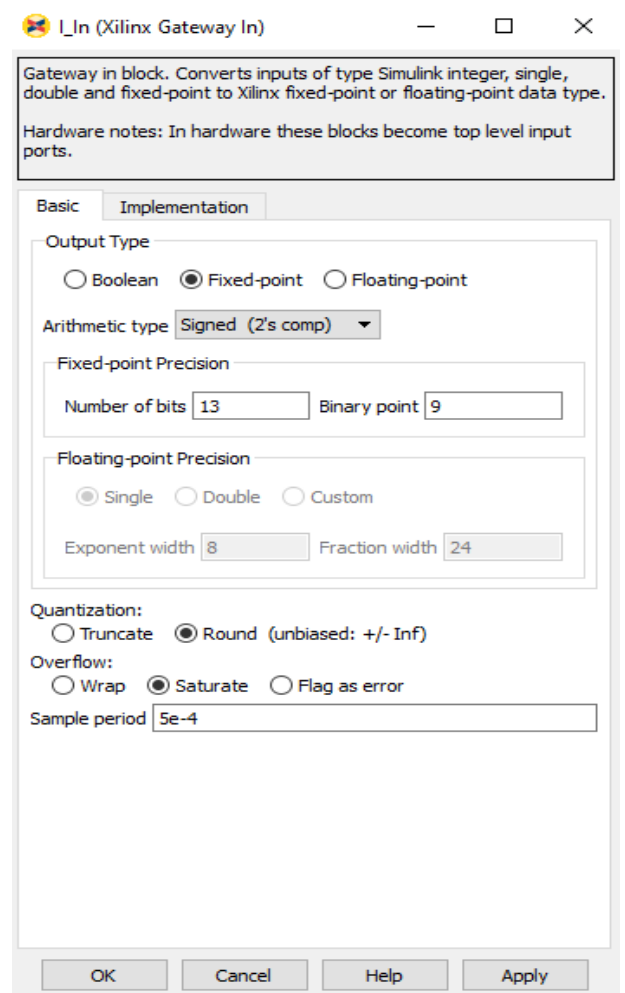


Figure IV-4: Les paramètres de I\_In

### IV.3.3 La commande P&O sur xilinx vivado

Modèle de la commande P&O sur Xilinx vivado en langage VHDL :

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_arith.ALL;
use IEEE.std_logic_unsigned.all;

entity pando2 is
    Port ( Ipv : in STD_LOGIC_VECTOR (12 downto 0);
          Vpv : in STD_LOGIC_VECTOR (12 downto 0);
          D_out : out STD_LOGIC_VECTOR (23 downto 0);
          C : in STD_LOGIC;
          CE : in STD_LOGIC);
end pando2;
```

```
architecture Behavioral of pando2 is

signal delta :STD_LOGIC_VECTOR (23 downto 0) :=
"000000001000001100010010";
signal mp : STD_LOGIC_VECTOR (25 downto 0) ;
signal mp2 : STD_LOGIC_VECTOR (25 downto
0):="000000000000000000000000";
signal v2 : STD_LOGIC_VECTOR (12 downto 0):= "000000000000";
signal pcp : STD_LOGIC ;
signal vcp : STD_LOGIC ;
signal x_out : STD_LOGIC ;
signal adout: STD_LOGIC_VECTOR (23 downto 0) ;
signal adout2: STD_LOGIC_VECTOR (23 downto 0)
:="001100110011001100110011";

begin

process (adout2,delta, x_out)

begin
    if x_out = '0' then
        adout <= adout2 + delta;
    else
        adout <= adout2 - delta ;
    end if;
end process;

mp <= Ipv *Vpv;

process (mp,mp2)
begin

    if ( mp > mp2 ) then
        pcp <= '1';
    else
        pcp <= '0';
    end if;

end process;
process (Vpv,v2)
begin

    if ( Vpv > v2 ) then
        vcp <= '1';
    else
        vcp <= '0';
    end if;

end process;
```

```
x_out <= not (pcp xor vcp) ;

process (c)
begin
  if c'event and c='1' then
    if CE='1' then
      mp2 <= mp;
      v2 <= Vpv;
      adout2 <= adout;
    end if;
  end if;
end process;

D_out <= adout;

end Behavioral;
```

### IV.3.4 La fonction min\_max

Modèle de la fonction sur Xilinx vivado en langage VHDL :

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity abs1 is
  Port ( D_in : in STD_LOGIC_VECTOR (23 downto 0);
        D_out : out STD_LOGIC_VECTOR (23 downto 0));
end abs1;

architecture Behavioral of abs1 is

  signal min :STD_LOGIC_VECTOR (23 downto 0) :=
"000000101000111101011100";
  signal max :STD_LOGIC_VECTOR (23 downto 0) :=
"111001100110011001100110";
begin
  process (D_in)
  begin
    if (D_in > min and D_in < max ) then
      D_out <= D_in;
    end if;
  end process;
end Behavioral;
```



## IV.4.1 Conception des coefficients et le filtre FIR

Dans cette partie nous avons élaboré un filtre FIR , D'abord nous avons créé les coefficients h à l'aide de Matlab :

### IV.4.1.1 Le programme script

```
clear
close all
Fc=5000; % fréquence de coupure
Fs=200000;% fréquence d'échantillonnage
R1=5e-3;
R2=5e-3;
N=15;% Nombre de coeffi
F0=(Fc)/(Fs/2);
Hi = firceqrip(N,F0,[R1 R2],'passedge');
n=12;
m=n+1;

H=Hi*(2^(m));
fid = fopen('Hi_bin1.txt','wt');

for i=1:(N+1)
    ai=i-1;
    scbin=dec2bin(H(1,i),m);
    fprintf(fid,'constanth%d:std_logic_vector(%d downto 0) :="%s"
;\n',ai, n, scbin);
end
```

### IV.4.1.2 Le filtre en VHDL

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.NUMERIC_STD.ALL;

entity fir is
port(x:in std_logic_vector (12 downto 0);
    c:in std_logic;
    ce:in std_logic;
    y:out std_logic_vector (12 downto 0));
end fir ;

architecture Behavioral of fir is

constant h0:signed(12 downto 0):="1111110111100";
```

```
constant h1:signed(12 downto 0):="1111110001000";
constant h2:signed(12 downto 0):="1111110001111";
constant h3:signed(12 downto 0):="0000000101011";
constant h4:signed(12 downto 0):="0000110001011";
constant h5:signed(12 downto 0):="0001101111100";
constant h6:signed(12 downto 0):="0010101101100";
constant h7:signed(12 downto 0):="0011010100011";
constant h8:signed(12 downto 0):="0011010100011";
constant h9:signed(12 downto 0):="0010101101100";
constant h10:signed(12 downto 0):="0001101111100";
constant h11:signed(12 downto 0):="0000110001011";
constant h12:signed(12 downto 0):="0000000101011";
constant h13:signed(12 downto 0):="1111110001111";
constant h14:signed(12 downto 0):="1111110001000";
constant h15:signed(12 downto 0):="1111110111100";

signal m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, m12, m13,
m14, m15, m16:signed(25 downto 0);

signal d1, d2, d3, d4, d5, d6, d7, d8, d9, d10, d11, d12, d13,
d14, d15:signed(12 downto 0):="0000000000000";

signal yin:signed(25 downto 0);
signal yout:signed(12 downto 0);

begin

yin <= m1+ m2+ m3+ m4+ m5+ m6+ m7+ m8+ m9+ m10+ m11+ m12+ m13+
m14+ m15+ m16;

m1 <= signed(x)*h0;
m2 <= d1*h1;
m3 <= d2*h2;
m4 <= d3*h3;
m5 <= d4*h4;
m6 <= d5*h5;
m7 <= d6*h6;
m8 <= d7*h7;
m9 <= d8*h8;
m10 <= d9*h9;
m11 <= d10*h10;
m12 <= d11*h11;
m13 <= d12*h12;
m14 <= d13*h13;
m15 <= d14*h14;
m16 <= d15*h15;

process (c)
begin
if c='1' and c'event then
```



```
if ce='1' then
  d1<=signed(x);
d2 <= d1;
d3 <= d2;
d4 <= d3;
d5 <= d4;
d6 <= d5;
d7 <= d6;
d8 <= d7;
d9 <= d8;
d10 <= d9;
d11 <= d10;
d12 <= d11;
d13 <= d12;
d14 <= d13;
d15 <= d14;
end if;
end if;
end process;

yout <= yin(25 downto 13);

y <= STD_LOGIC_VECTOR(yout);

end Behavioral;
```

## IV.5 Implémentation d'une commande P&O avec sous-échantillonnage et filtrage FIR sur FPGA

### IV.5.1 Le schéma du système PV avec la commande P&O avec sous-échantillonnage et filtrage FIR sur Simulink

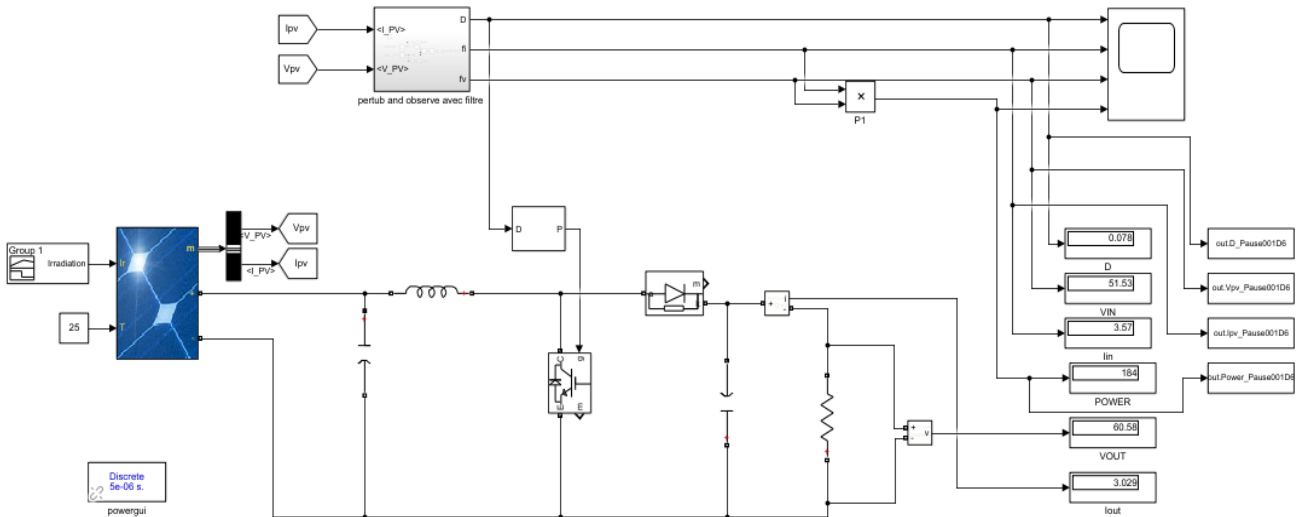


Figure IV-6: Le schéma du système PV avec la commande P&O avec sous-échantillonnage et filtrage FIR sur Matlab Simulink

### IV.5.2 Le schéma de la commande P&O avec sous-échantillonnage et filtrage FIR sur Simulink

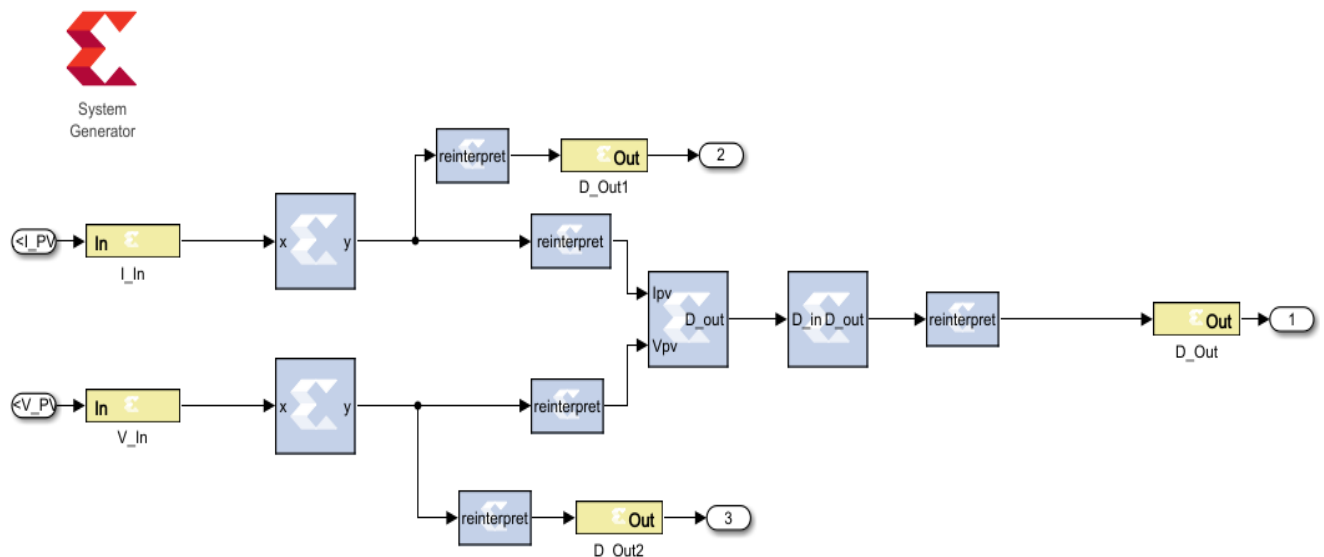


Figure IV-7: la commande P&O avec sous-échantillonnage et filtrage FIR sur Simulink à l'aide de Xilinx.

## **IV.6 Conclusion**

Dans ce chapitre, nous avons entamé la partie la plus importante de cette étude. Il s'agit de l'implémentation d'une commande P&O avec un sous-échantillonnage sur FPGA et avec filtrage FIR., ainsi que la conception du programme en VHDL. Dans le chapitre suivant nous allons visionner et analysé les résultats obtenus par cette étude.

## **Chapitre V : Validation et résultats**

## V.1 Introduction

Ce chapitre est consacré à la simulation des commandes mentionnées dans le chapitre précédent à l'aide des logiciels Xilinx et Matlab Simulink, et l'interprétation des résultats obtenus , après nous allons implémenter ces codes dans le circuit FPGA.

## V.2 Résultats de la commande P&O avec un sous-échantillonnage

-courant

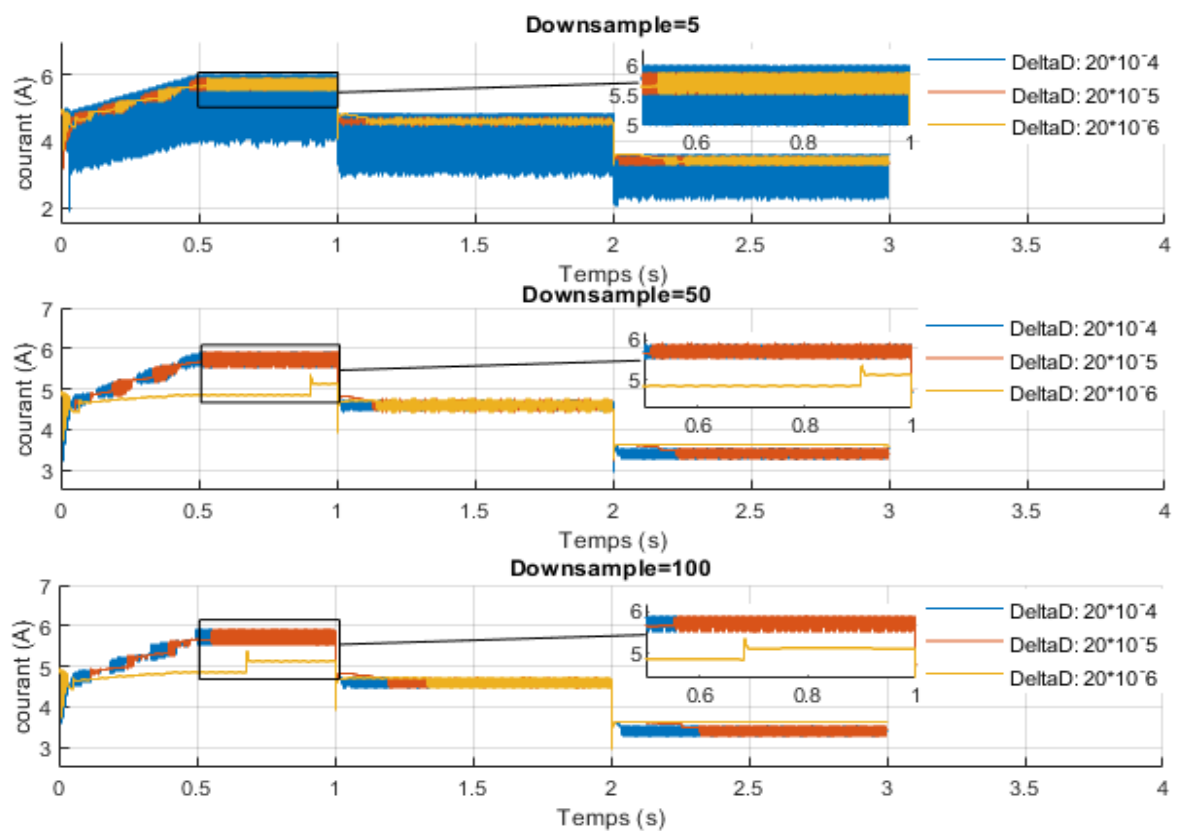


Figure V-1: La variation du courant (A) en fonction du temps (s) pour différent delta D.

-Tension

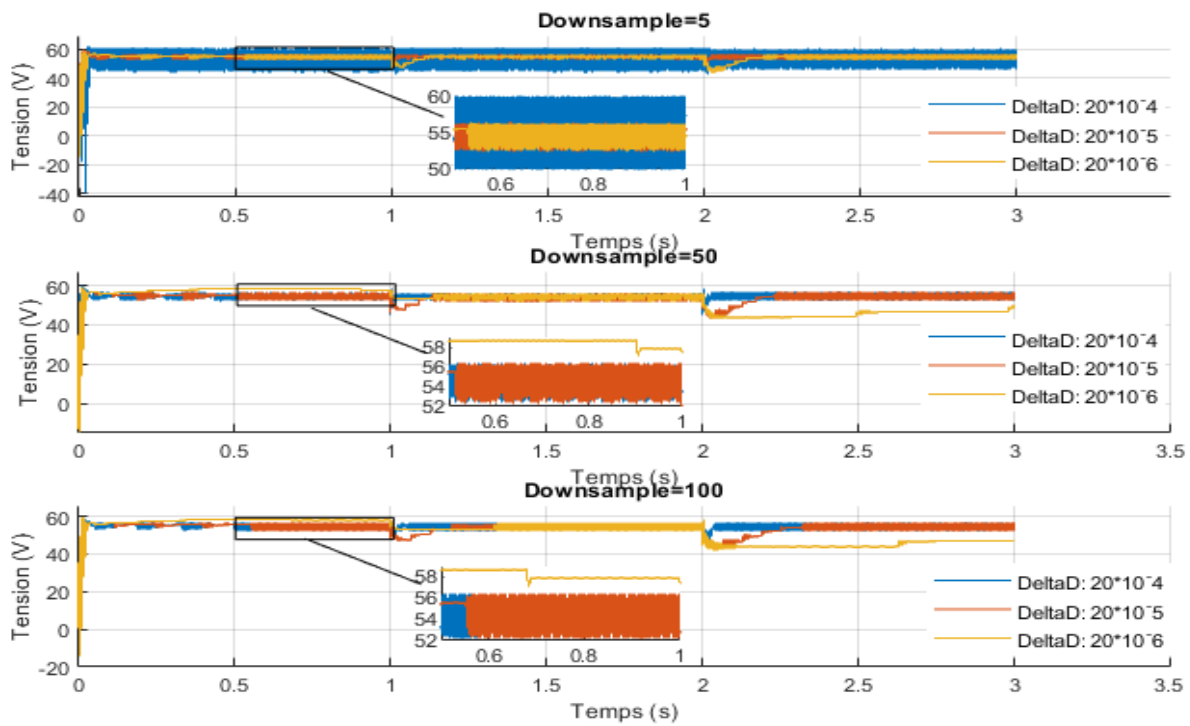


Figure V-2: La variation de la tension (V) en fonction du temps (s) pour diffèrent delta D.

-Puissance

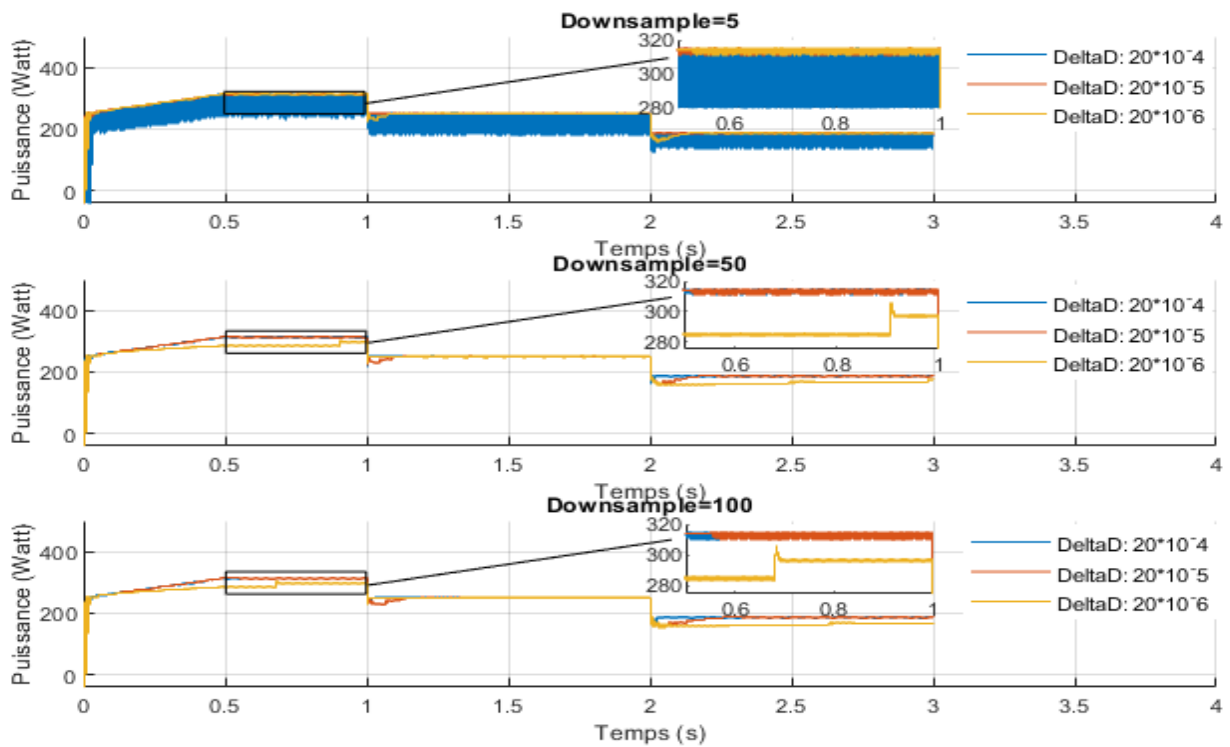


Figure V-3: La variation de la puissance (Watt) en fonction du temps (s) pour diffèrent delta D.

-Duty Cycle

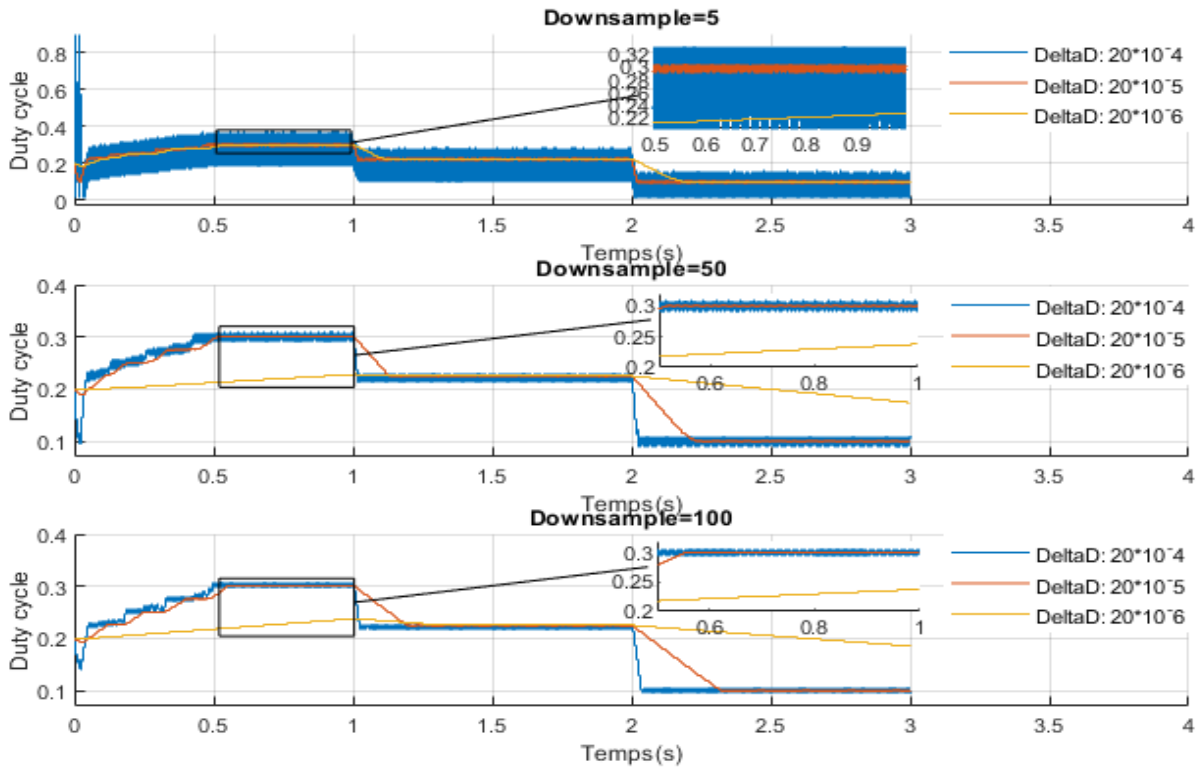


Figure V-4: La variation du duty cycle en fonction du temps (s) pour différent delta D.

**V.2.1 Interprétation**

Les courbes de la tension, courant, puissance et le rapport cyclique obtenus avec sous-échantillonnage varie en fonction du temps (s), et nous remarquons que le changement de l’irradiation à un grand effet sur le résultat, tel que quand l’irradiation diminue les courbes de la tension, courant, puissance et le rapport cyclique diminuent et vis vers ça. Nous remarquons que dès le delta D augmente les ondulations augmentent aussi .

**V.2.2 Comparaison des résultats**

Nous avons calculé les rendements de puissance pour différents delta D avec différente valeurs de sous échantillonnage sur Xilinx :

Tableau V-1: Comparaison entre les rendements de puissance avec sous échantillonnage pour différent deltaD sur Xilinx.

Sous échantillonnage	100			50			5				
	Delta D	20.10 <sup>-4</sup>	20.10 <sup>-5</sup>	20.10 <sup>-6</sup>	Delta D	20.10 <sup>-4</sup>	20.10 <sup>-5</sup>	20.10 <sup>-6</sup>	Delta D	20.10 <sup>-4</sup>	20.10 <sup>-5</sup>
$\eta$ (%)	99.2559	99.3055	92.8883	99.4382	99.3816	91.2004	90.1704	99.4036	99.4411		

Nous remarquons que le sous échantillonnage à un effet sur le rendement, quand le sous échantillonnage égale à 100 le rendement désiré est obtenus à  $\Delta D = 20 \cdot 10^{-5}$  et lorsque le sous échantillonnage égale à 50 le rendement désiré est obtenus à  $\Delta D = 20 \cdot 10^{-4}$ , et quand le sous échantillonnage égale à 5 le rendement désiré est obtenus à  $\Delta D = 20 \cdot 10^{-6}$ .

### V.3 Résultats de la commande P&O avec un sous-échantillonnage et filtrage FIR

Nous avons choisi le cas où le  $\Delta D$  égale  $20 \cdot 10^{-4}$  et sous échantillonnage égale à 100 pour poursuivre notre étude :

-Courant

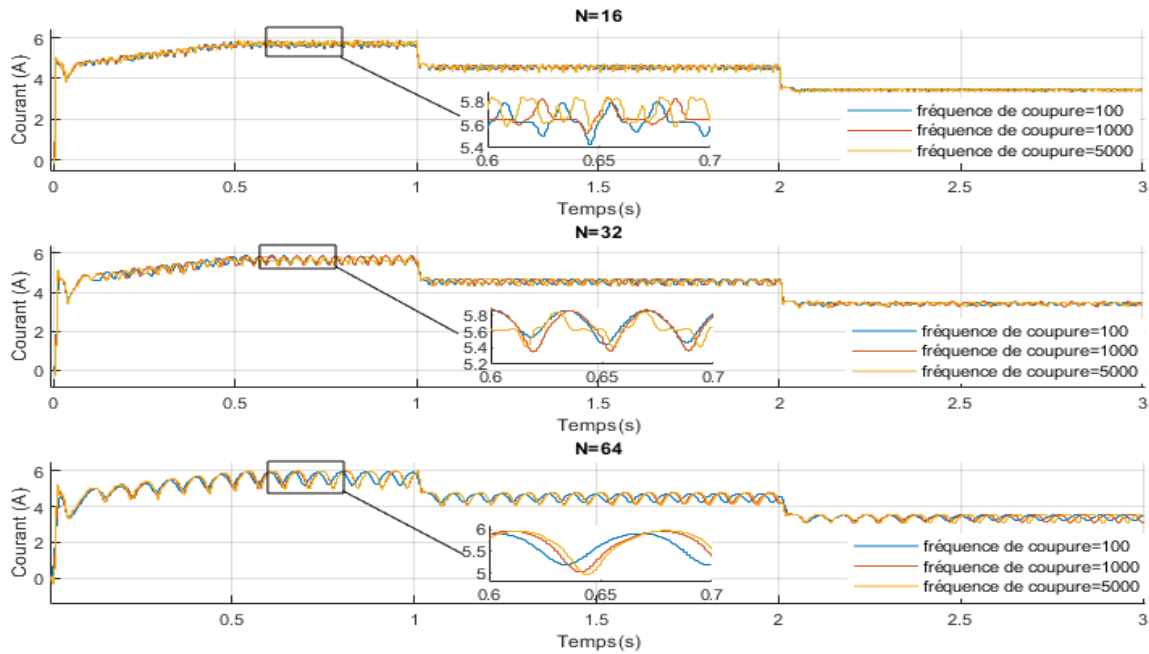


Figure V-5: La variation du courant (A) en fonction du temps (s) pour diffèrent fréquence de coupure.



-Tension

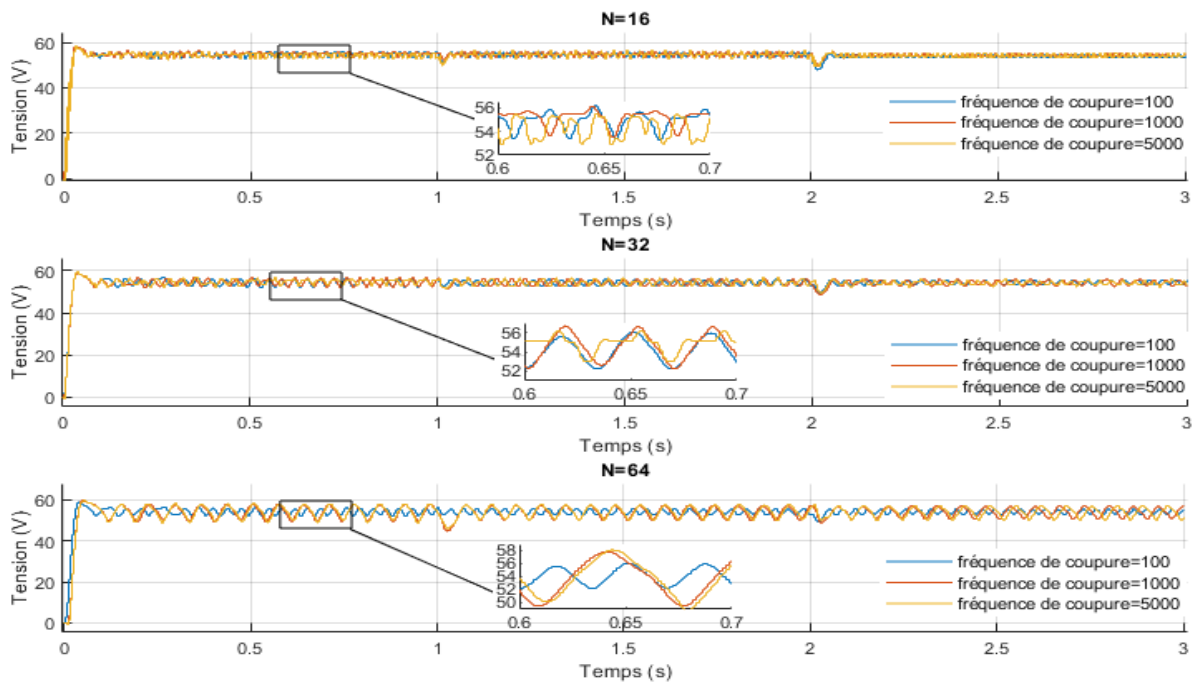


Figure V-7: La variation de la tension (V) en fonction du temps (s) pour diffèrent fréquence de coupure.

-Puissance

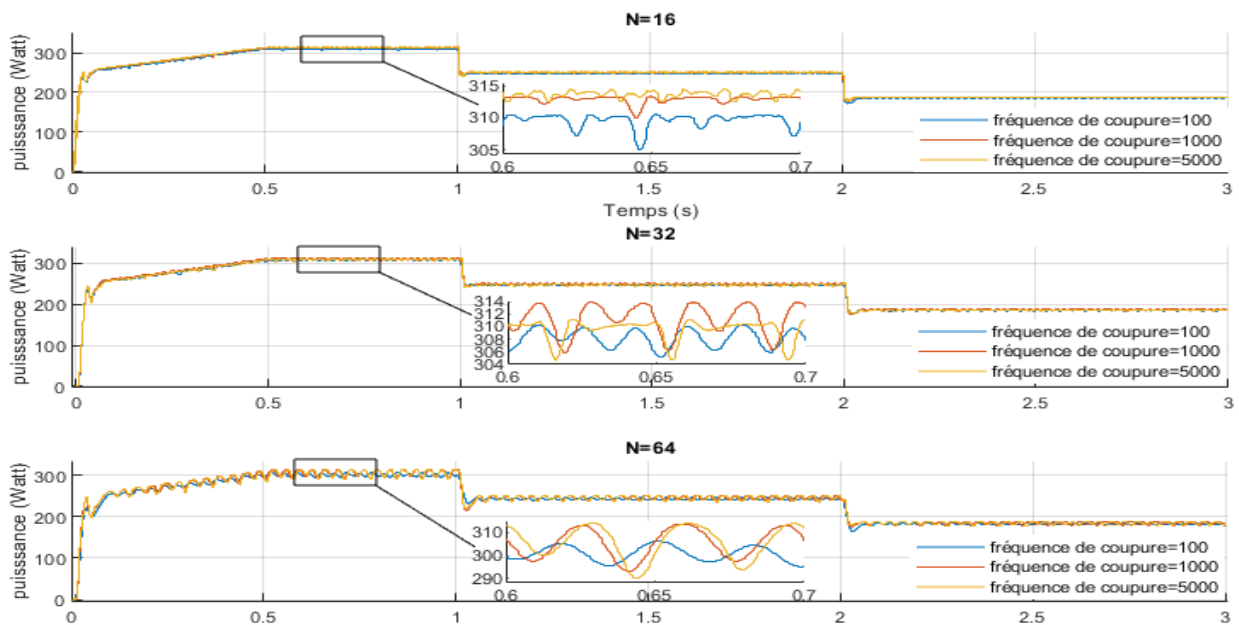


Figure V-6: La variation de la puissance (Watt) en fonction du temps (s) pour diffèrent fréquence de coupure.

-Duty Cycle

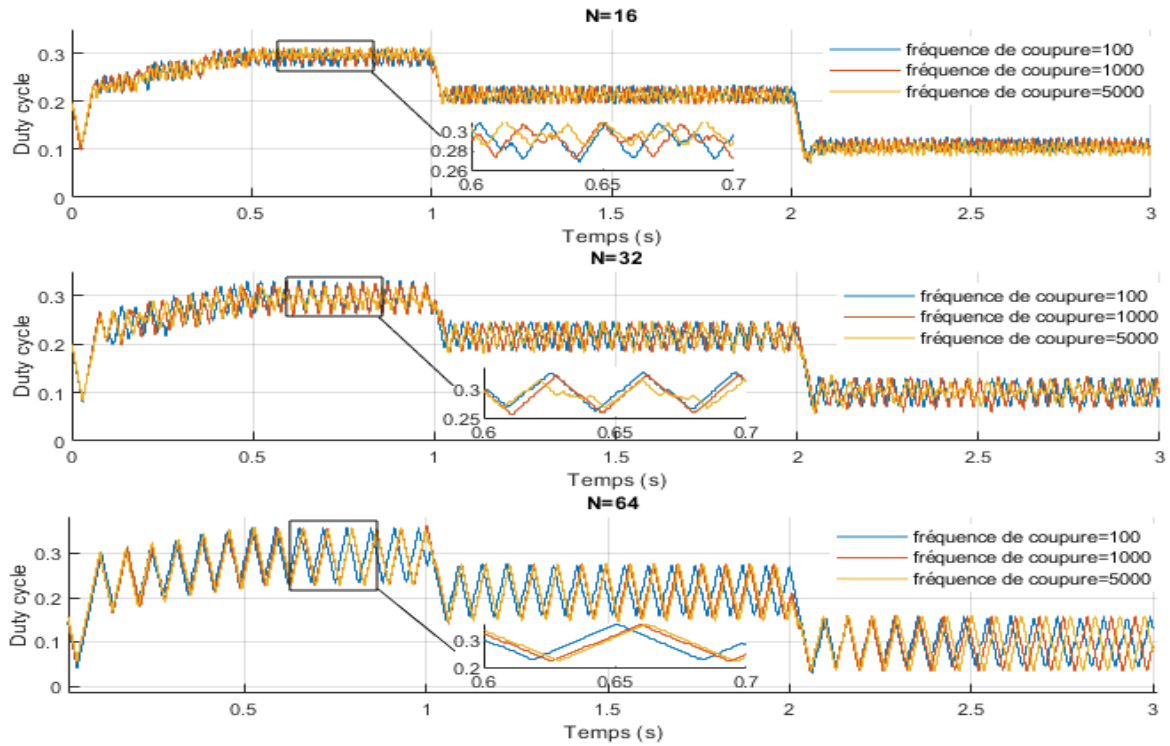


Figure V-8: La variation du duty cycle en fonction du temps (s) pour différent fréquence de coupure.

### V.3.1 Interprétation

Les courbes de la tension, courant, puissance et le rapport cyclique obtenus avec un sous- échantillonnage et filtre varie en fonction du temps (s),et nous remarquons que l’ajout du filtre FIR à améliorer tous les résultats et celle de la puissance en particulier.

### V.3.2 Comparaison des résultats

Nous avons calculé les rendements de puissance pour un delta  $D \cdot 20 \cdot 10^{-4}$  avec un sous échantillonnage 100 pour différents ordres et fréquence de coupure:

Tableau V-2: Comparaison entre les rendements de puissance avec un sous échantillonnage et filtre pour delta  $D=20 \cdot 10^{-4}$

Ordre	N=16			N=32			N=64		
	Fréquence de coupure	100	1000	5000	100	1000	5000	100	1000
$\eta$ (%)	98.2624	99.2822	99.5653	97.8102	98.8319	98.2463	95.6067	96.8256	97.8160

Nous observons que dès que l'ordre augmente le rendement diminue , à l'ordre 16 nous avons obtenus un meilleur rendement pour les trois fréquences de coupure .et nous remarquons quand la fréquence de coupure augmente le rendement s'améliore.

#### V.4 Comparaison des résultats avec et sans filtre

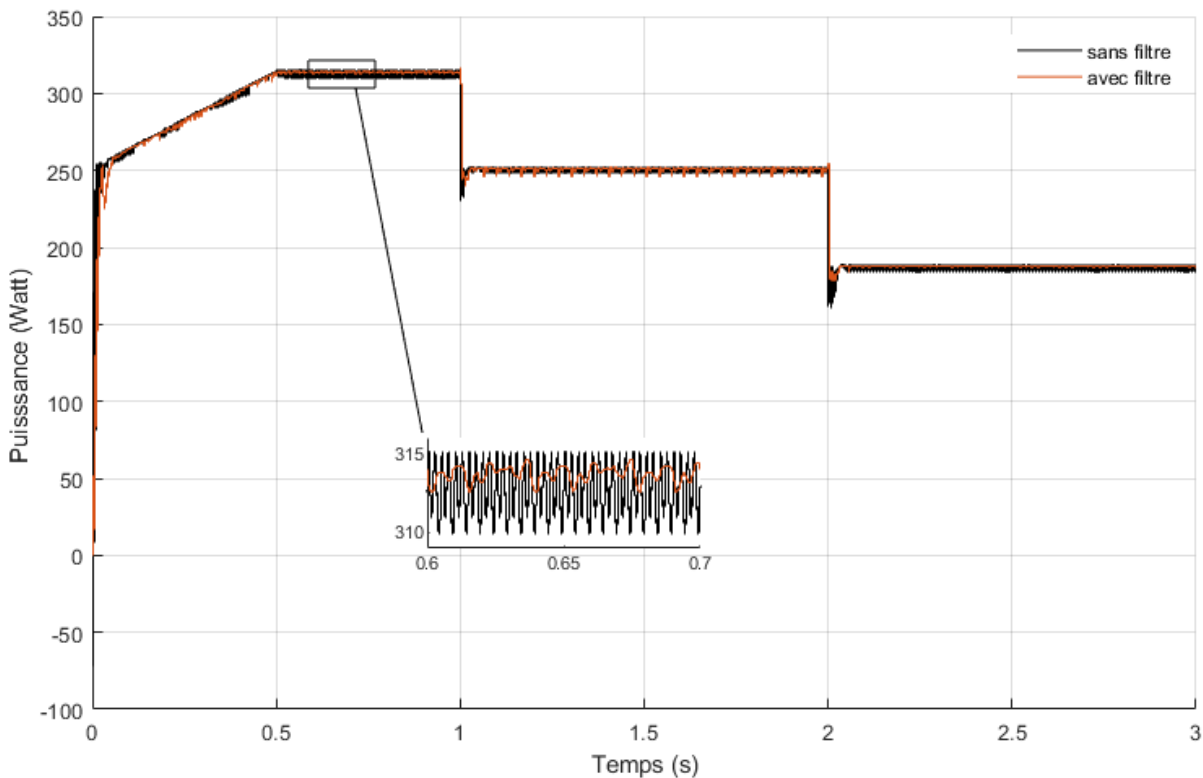


Figure V-9: Variation de la puissance (Watt) en fonction du temps (s) avec et sans filtre.

Dans le cas d'implémentation d'une commande P&O avec un sous-échantillonnage sur FPGA les résultats sont presque identiques avec ceux de Matlab. Tandis que l'implémentation de la commande P&O avec sous-échantillonnage et filtrage FIR sur FPGA sont améliorés au niveau de la puissance et non par le rapport cyclique.

Nous justifions l'augmentation de l'ondulation dans le rapport cyclique par augmentation des calculs, pour donner des valeurs de puissance plus précises.

Tableau V-3: : Comparaison entre les rendements de puissance pour delta  $D=20.10^{-4}$ .

	Sans Filtre		Avec filtre
	Sans sous-échantillonnage	Avec sous-échantillonnage sur xilinx	Avec sous-échantillonnage
$\eta$ (%)	55.2863	99.4382	99.5653

Nous remarquons l'amélioration du rendement avec un sous échantillonnage et plus avec un filtre.

## V.5 Plateforme de validation sur circuit FPGA

Les résultats de simulation obtenus nous ont permis d'implanter le MPPT par la méthode de perturbation et observation avec un sous échantillonnage et filtre sur les deux cartes Genesys 2 Kintex-7 FPGA et Spartan-3E FPGA :



Figure V-10: Genesys 2 Kintex-7 FPGA



Figure V-11: Spartan-3E FPGA .

### V.5.1 Implémentation du MPPT sur FPGA

A l'aide de Matlab Simulink et Xilinx nous avons implanter le schéma de la commande MPPT avec un sous échantillonnage et filtre référencé dans le chapitre précédent (la figure IV-7).

Tableau V-4: les valeurs choisies pour l'implémentation sur la carte

Delta D	Sous échantillonnage	Fréquence de coupure	L'ordre
$20 \cdot 10^{-4}$	100	5000	16

### V.6 Conclusion

Dans ce chapitre, nous avons simulé l'algorithme numérique MPPT écrit en VHDL référencé dans le chapitre précédent, à l'aide de logiciel Xilinx et Matlab Simulink pour vérifier le bon fonctionnement de notre système. Les résultats de simulation obtenus étaient satisfaisants qui montre une grande amélioration de la puissance. Cela nous a permis d'implémenter l'algorithme dans une carte Genesys 2 Kintex-7 FPGA et Spartan-3E FPGA pour contrôler un système réel.

# Conclusion générale

Ce projet de fin d'études porte sur l'étude de l'optimisation d'un contrôleur MPPT, qui garantit la poursuite du point de puissance maximale dans une chaîne de conversion PV, afin que l'énergie maximale fournie par le générateur PV puisse être obtenue dans des conditions climatiques spécifiques (irradiation).

Nous avons commencé le côté pratique et la simulation de notre méthode, d'abord nous avons amélioré la commande MPPT « la méthode de perturbation et d'observation (P&O) », pour essayer d'explorer, d'exploiter et tirer la puissance maximale, par un sous échantillonnage, après l'essai de plusieurs valeurs de sous échantillonnage (5, 50, 100) pour différents delta D ( $20 \cdot 10^{-4}$ ,  $20 \cdot 10^{-5}$ ,  $20 \cdot 10^{-6}$ ) nous remarquons que la puissance est bien améliorée avec un sous échantillonnage 100 pour un delta D  $20 \cdot 10^{-5}$ .

Ensuite, nous avons traduit cette commande en langage VHDL et l'accordée à un filtre FIR, les programmes sont cités dans le chapitre 4.

En dernier, nous avons simulé l'algorithme numérique MPPT écrit en langage VHDL accordée à un filtre FIR, à l'aide des logiciels Xilinx et Matlab Simulink pour vérifier le bon fonctionnement de notre système. Les résultats de simulation obtenus étaient satisfaisants qui montrent l'optimisation de la puissance maximale, cela nous a permis d'implémenter l'algorithme dans une carte Genesys 2 Kintex-7 FPGA et Spartan-3E FPGA pour contrôler un système réel.

Enfin, nous espérons que notre travail est une modeste contribution dans la recherche relative à l'étude de la maîtrise de l'énergie photovoltaïque.

## Références bibliographiques

- [1] Beauvais A, Chevillard N, Paredes MG, Heisz M, Rossi R, Schmela M., *Global market outlook for solar power / Connecting african and european developers and investors*, 2018-2022.
- [2] «L'ACTUALITÉ DE L'ÉNERGIE DURABLE,» <https://www.renouvelle.be>. [Accès le juillet 2022].
- [3] P. S, *Nouvelles architectures distribuées de gestion et de conversion de l'énergie pour les applications photovoltaïques*, Thèse de doctorat, Université de Toulouse III T-Paul Sabatier., 2010.
- [4] B. L, *Détection et localisation de défauts pour un système PV*, Thèse de doctorat. Université de Grenoble, 2011..
- [5] M. T, *Caractérisation de panneaux solaires photovoltaïques en conditions réelles d'implantation et en fonction des différentes technologies. Thèse de Doctorat*, Université ParisSud École doctorale : STITS, 2014.
- [6] A.Chermitti, O.Boukli-Hacene, S.Mouhadjer, *Design of a Library of Components for Autonomous Photovoltaic System under Matlab/Simulink*, International Journal of Computer Applications (0975 – 8887), September 2012.
- [7] «<https://www.ummtto.dz/dspace/handle/ummtto/6680>,».
- [8] Cottet, Francis, AIDE-MÉMOIRE TRAITEMENT DU SIGNAL, Paris: Dunod, 2000.
- [9] Roy, S. C. Dutta, «A State of the Art Survey on Linear Phase Digital,» *IETE Journal of Research*, 1976.
- [10] EZ-ZRARAE Amine, LAMRHARI Amina, ELYOUSSOUFI Mohammed, *Filtre numérique*.
- [11] Kourgi, Assia, *Traitement Numérique du Signal (Support de cours)*, Université des Sciences et de la Technologie Houari Boumediene, 20 June 2020.
- [12] O. Sentieys, *Analyse et synthèse des filtres numériques: une introduction*, Rennes, mai 2008.
- [13] Aigrain, Pierre, TRAITEMENT NUMÉRIQUE, France, août 2006.
- [14] CHEBOUROU Mohammed , REMAOUN Ahmed Ilyas, *Modélisation d'un filtre numérique multi-cadence à l'aide d'un DSP TMS320VC5416*, Université Aboubakr Belkaïd ,Tlemcen, 24/06/2018.
- [15] Yassine Ariba et Jérôme Cadieux, *Manuel MATLAB*, Toulouse.
- [16] «<https://www.futura-sciences.com/>,».
- [17] D. Guihal, *Modélisation en langage VHDL-AMS des systèmes*, Université Toulouse III : 25 Mai 2007.

[18] «<https://www.techno-science.net/>,».

[19] Beauvais A, Chevillard N, Paredes MG, Heisz M, Rossi R, Schmela M, *Global Market Outlook For Solar Power / Power FS. Mobilising Investments in Emerging Markets.*, 2019 - 2023.

[20] J.-C. MULLER, *Electricité photovoltaïque, Technique de l'ingénieur.*.

[21] Y. El Basri, S. Petibon, B. Estibals, C. Alonso, *New P&O MPPT algorithm for FPGA implementation*, IECON , 2010.