

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique



Département d'Electronique

Mémoire de Magister

Option : Electricité solaire

Présenté par

M^{elle} FAIZA BELHACHAT

Ingénieur d'état en électronique, USTHB

Option : contrôle

Thème

Commande

neuro-floue d'un hacheur MPPT

Membres du jury

Mr Haddadi Mourad, Professeur ENP

Président

M^{elle} Barazane Linda, Maître de conférences USTHB

Rapporteur

Mr Chérif Larbes, Maître de conférences ENP

Rapporteur

Mr Berkouk El-Madjid, Professeur ENP

Examineur

Mr Malek Ali, Maître de Recherches CDER

Examineur

Mr Bousbia-Salah Hichem, Maître de conférences ENP

Examineur

Juin 2007

Ecole Nationale Polytechnique 10, Avenue Hassen Badi, El-Harrach, ALGER

ملخص:

يمكن للوح الشمسي أن يشتغل في مجال واسع من التوتر ولتيار المخرج لكنه لا يستطيع أن يوفر الاستطاعة العظمى إلا عند قيم خاصة من التوتر والتيار.

في الواقع إن خصائص اللوح الشمسي تتعلق كثيرا بالضوء الشمسي والحرارة هذه التغيرات تؤدي إلى تذبذب نقطة الاستطاعة العظمى.

بسبب تغير موضع هذه النقطة نضع غالبا محولا مستمرا محكم أو أكثر لأجل تتبع نقطة الاستطاعة العظمى في كل لحظة هذه المحولات تعمل على ضمان تحويل هذه الاستطاعة القصوى إلى المستقبل .

العمل المطروح في هذه المذكرة يتمثل في استعمال تقنيتين جديدتين تعتمدان على الذكاء الاصطناعي والمتمثلتان في شبكة العصبونات الاصطناعية وتقنية شبكة العصبونات بالمنطق الغامض لتتبع نقطة الاستطاعة العظمى بطريقة مستمرة.

النتائج المتحصل عليها عن طريق المحاكاة تبين أن التقنيتين المستعملتين ذات نتائج جيدة لكن المراقب بطريقة شبكة العصبونات بالمنطق الغامض تمثل التقنية الأكثر فعالية في الحالة القصوى .

كلمات مفتاحية: تتبع الاستطاعة القصوى كهر وضوئي شبكة العصبونات الاصطناعية شبكة العصبونات بالمنطق الغامض المراقب العصبوني و العصبوني الغامض.

Abstract:

A photovoltaic generator can operate in a broad range of voltage and of current outputs but it can deliver a maximum power only for a particular value of current and voltage. The characteristic I (V) of the generator depends on solar insolation and temperature. These variations induce the fluctuation of the maximum power point. Because of this fluctuation, one often inserts between the generator and the load a static converter allowing to track the maximum power point at any time. These converters, known under the name of MPPT (Maximum Power point tracking) ensure the coupling between the generator and the load by forcing the first to deliver its maximum power. The work presented in this thesis is based on the use of two approaches: neural network and neuro-fuzzy control to optimize the flow of power. Results from simulation show that the two techniques used present good results but the neuro-fuzzy controller is more performant and robust.

Key-words: MPPT, photovoltaic, artificial neural network, neuro-fuzzy, neural network and neuro-fuzzy controller.

Résumé:

Un générateur photovoltaïque peut fonctionner dans une large gamme de tension et de courant de sortie mais il ne peut délivrer une puissance maximale que pour des valeurs particulières du courant et de la tension. En effet la caractéristique I (V) du générateur dépend de l'éclairement solaire et de la température. Ces variations climatiques entraînent la fluctuation du point de puissance maximale. A cause de cette fluctuation, on intercale souvent entre le générateur et le récepteur un ou plusieurs convertisseurs statiques commandés permettant de poursuivre le point de puissance maximale. Ces commandes sont, connus sous le nom de MPPT (Maximum Power Point Tracking) associées au hacheur, lequel assure le couplage entre le générateur PV et le récepteur en forçant le premier à délivrer sa puissance maximale.

Le travail présenté dans ce mémoire s'appuie sur l'utilisation de deux approches : neuronale et neuro-floue pour la poursuite du PPM. Les résultats de simulation ont montré que ces deux techniques présentent de bons résultats mais le contrôleur neuro-flou est plus performant et plus robuste.

Mots clés : MPPT, photovoltaïque, les réseaux de neurone artificiels, les réseaux neuro-flous, contrôleur neuronal et neuro-flou.

Sommaire

	Page
Introduction générale	1
Chapitre I : Techniques intelligentes et classiques de la commande MPPT	
I.1.Introduction	3
I.2.Le générateur photovoltaïque	4
I.2.1. Principe	4
I.3.Les performances du générateur photovoltaïque	5
I.3.1.Caractéristique Courant-Tension	6
I.3.2.Influence de l'Éclairement	7
I.3.3.Influence de la Température	8
I.4.Connexion directe source charge	10
I.5.Définition de l'étage d'adaptation (Hacheur MPPT)	11
I.6.Principe de la recherche du point de puissance maximale	12
I.7.Classification des commandes MPPT	12
I.7.1.Les méthodes analogiques	14
I.7.2 Méthodes micro programmées	14
I.7.3. Recherche extrémale adaptative.....	15
I.7.3.1. Asservissement de la tension du générateur PV	14
I.7.3.2. Asservissement de la puissance du générateur PV	15
I.8.Critères de qualité d'une commande MPPT	15
I.9.Techniques intelligentes pour la commande MPPT	16
I.9.1.Commande par les algorithmes génétiques	16
I.9.2.Commande par la logique floue	17
I.9.3.Commande par réseau de neurones	17
I.9.4. Commande par réseaux neuro-flous	17
I.10.Conclusion	17
Chapitre II : Concepts des réseaux de neurones et de la logique floue	
II.1.Introduction	19
II.2.Les réseaux de neurones	19

II.2.1.Présentation générale	19
II.2.1.1.Le neurone biologique et le neurone formel	19
II.2.1.2.Modélisation	21
II.2.2. Classification et propriétés des RNA	22
II.2.2.1.Apprentissage Connexionniste	22
II.2.2.2.Les règles d'apprentissage	22
II.2.2.3.Type des Unités	26
II.2.2.4.Type d'Architecture du Réseau	27
II.2.2.5.Type d'Application du Réseau	28
II.2.3.Le perceptron multicouches (<i>PMC</i>)	29
II.2.4. Considérations pratiques pour le choix du réseau de neurones adéquat	30
II.2.5.Paramétrage et conception d'un réseau de neurone	32
II.2.6.Les limitations d'un réseau de neurone	33
II.2.7.Conclusion	34
II.3.La logique floue	34
II.3.1. Introduction	34
II.3.2.Structure générale d'un système flou	35
II.3.3. Conception d'un système flou	36
II.3.3.1.Univers de discours	36
II.3.3.2.Fuzzification	37
II.3.3.3. Inférences floues	41
II.3.3.3.1. Opérateurs de la logique floue	42
II.3.3.3.2. méthodes d'inférences floues	43
II.3.3.4. Defuzzification	46
II.3.3.Commande floue	47
II.3.4.Avantages et inconvénients de la commande par logique floue	48
II.3.5. Conclusion	49
II.4.L'approche neuro-floue	49
II.4.1. Introduction	49
II.4.2.Motivations pour une approche hybride	50
II.4.3.Systèmes hybrides neuro-flous	51
II.4.4.Système d'inférences flou basé sur les réseaux de neurones adaptatifs	52
II.4.4.1.Vue d'ensemble d'ANFIS	53
II.5.Conclusion	56

Chapitre III : Conception d'un contrôleur MPPT neuronal

III.1.Introduction	57
III.2. Description et architecture du contrôleur neuronal MPPT proposé	58
III.2.1.Choix de la structure du réseau	58
III.2.2. Présentation des deux algorithmes utilisés	61
III.2.2.1. Apprentissage par rétro propagation de gradient	61

III.2.2.2. Apprentissage par Levenberg Marquardt	62
III.2.3. Validation	63
III.3.Résultats de simulation de l'application de l'approche neuronale	64
III.3.1.Simulation du fonctionnement sous des conditions standard	64
III.3.2.Simulation sous des conditions variables d'ensoleillement	65
III.3.3. Simulation sous des conditions variables de température	70
III.3.4.Simulation sous des conditions aléatoires d'ensoleillement et de température	74
III.4.Conclusion	76

Chapitre IV : Conception d'un contrôleur MPPT neuro-flou

IV. 1.Introduction	77
IV.2.Conception d'un contrôleur MPPT à base de réseau neuro-flou	78
IV.2.1 : Description et structure du contrôleur MPPT neuro-flou proposé	78
IV. 2.2. Apprentissage du contrôleur	81
IV. 2. 3. Test de la capacité de généralisation après apprentissage	83
IV.3. Résultats de simulation de l'application de l'approche neuro-flou	86
IV.3.1.Simulation du fonctionnement sous des conditions standard	86
IV.3.2. Simulation des conditions variables d'ensoleillement	87
IV.3.3. Simulation sous des conditions variables de température	89
IV.3.4.Simulation sous des conditions aléatoires d'ensoleillement et de température.....	91
IV.4 .Synthèse et comparaison des résultats des trois techniques	92
III.4.1.Simulation du fonctionnement sous des conditions standard	92
II.4.2.Simulation sous des conditions variables d'ensoleillement	93
IV.4.3. Simulation sous des conditions variables de température	95
IV.4. Comparaison des trois approches	96
IV.5 .Conclusion	96
Conclusion générale	98
Annexe	100
Bibliographie	110

Introduction Générale

Introduction générale

La production d'énergie est un défi de grande importance pour les années à venir. En effet, les besoins énergétiques des sociétés industrialisées ne cessent d'augmenter. Par ailleurs, les pays en voie de développement auront besoin de plus en plus d'énergie pour mener à bien leur développement. De nos jours, une grande partie de la production mondiale d'énergie est assurée à partir de sources fossiles. La consommation de ces sources donne lieu à des émissions de gaz à effet de serre et par conséquent une augmentation du réchauffement et de la pollution. Le danger supplémentaire est qu'une consommation excessive du stock de ressources naturelles réduit les réserves de ce type d'énergie de façon dangereuse pour les générations futures.

En vertu du Protocole de Kyoto, l'Union Européenne s'est engagée à réduire ses émissions de gaz à effet de serre de 8% en dessous de leurs niveaux de 1990 sur la période 2008-2012. Les énergies renouvelables telles que l'énergie éolienne, l'énergie solaire, l'énergie hydroélectrique et la biomasse doivent jouer un rôle important pour atteindre cet objectif. Dans ce contexte, en septembre 2001, l'Union Européenne a adopté la directive relative à la promotion de l'électricité produite à partir de sources d'énergie renouvelable sur le marché intérieur de l'électricité. L'objectif de cette directive est la promotion et l'exploitation à l'avenir du potentiel des sources d'énergie renouvelable.

Par énergie renouvelable, on entend des énergies issues du soleil, du vent, de la chaleur de la terre, de l'eau ou encore de la biomasse. A la différence des énergies fossiles, les énergies renouvelables sont des énergies à ressource illimitée qui regroupent un certain nombre de filières technologiques selon la source d'énergie valorisée et l'énergie utile obtenue. La filière étudiée dans ce mémoire est l'énergie solaire photovoltaïque (PV).

L'énergie solaire photovoltaïque provient de la transformation directe d'une partie du rayonnement solaire en énergie électrique effectuée par le biais d'une cellule dite photovoltaïque basée sur un phénomène physique appelé effet photovoltaïque. Ce dernier, a pour rôle de produire une force électromotrice lorsque la surface de cette cellule est exposée à la lumière. La tension générée peut varier en fonction du matériau utilisé pour la fabrication de la cellule. L'association de plusieurs cellules PV en série/parallèle donnent lieu à un générateur photovoltaïque (GPV) qui a une caractéristique courant-tension (I-V) non linéaire présentant un point de puissance maximale (PPM) laquelle dépend du niveau

d'éclairage et de la température de la cellule ainsi que du vieillissement de l'ensemble. De plus, le point de fonctionnement du GPV dépend directement de la charge qu'il alimente.

Afin d'extraire à chaque instant le maximum de puissance disponible aux bornes du GPV l'insertion d'un étage d'adaptation entre le générateur et la charge pour coupler les deux éléments le plus parfaitement possible est indispensable.

Le problème du couplage parfait entre un générateur photovoltaïque et une charge de type continu n'est pas encore réellement résolu. En effet un des verrous technologiques qui existe dans ce type de couplage réside dans la manière la plus adéquate de transférer la puissance maximale du GPV à la charge à cause d'une mauvaise adaptation. Le point de fonctionnement qui en découle est alors parfois très éloigné du point de puissance maximale, ce qui a axé les recherches à développer plusieurs algorithmes de contrôle qui effectue une recherche de point de puissance maximale lorsque le GPV est couplé à une charge à travers un convertisseur statique.

Dans ce mémoire, nous nous intéressons au problème de poursuite du maximum de puissance d'un générateur suivant deux concepts : l'approche neuronale et l'approche neuro-floue.

A cet effet, Il est donc nécessaire d'utiliser des méthodes de poursuite permettant de faire un contrôle fiable du point de puissance maximale par utilisation d'un réseau de neurone. De plus, et dans le même contexte, une méthode neuro-floue est proposée pour permettre à la fin de comparer les performances des deux approches par des simulations à l'aide du logiciel Matlab Simulink. Le présent mémoire est donc subdivisé en quatre chapitres à savoir :

Dans le premier chapitre, nous présentons les principales caractéristiques d'un GPV ainsi que la problématique de l'énergie solaire photovoltaïque. En suite, nous décrivons les principaux types d'algorithmes de recherche du point de puissance maximale (MPPT) d'un GPV développés et les plus répandus, en particulier ceux émergents (les techniques intelligentes tel que la logique floue, les réseaux de neurones, le réseau neuro-flou).

Dans le deuxième chapitre nous présenterons une étude théorique sur les notions de la logique floue, les réseaux de neurones et neuro-flous ainsi que les concepts nécessaires à leurs conceptions, l'apprentissage et la validation.

Le troisième chapitre sera consacré à la présentation de notre travail, nous décrivons l'architecture du réseau de neurones proposé ainsi que le processus d'apprentissage adopté. Des tests et des simulations de performances du système global sont effectués afin de vérifier la marge de fiabilité de notre contrôleur.

Par la suite et dans le dernier chapitre et dans le but d'améliorer un tel processus nous avons pensé à introduire l'approche neuro-floue qui est susceptible de garder toujours de bonne performances même dans le cas extrême où le contrôle par le réseau de neurones se trouve moins efficace, ce qui est vérifié à travers une comparaison des résultats de simulation avec l'approche neuronale.

Nous terminerons notre mémoire par une conclusion générale dans laquelle nous donnerons les perspectives et les améliorations qui peuvent être prises en compte pour la continuité de ce travail.

Chapitre I

Techniques classiques et intelligentes de la commande MPPT

I.1.Introduction

L'énergie solaire photovoltaïque (PV) résulte de la conversion directe de l'énergie provenant de photons résidants dans le rayonnement solaire, en énergie électrique, par le biais de capteurs fabriqués avec des matériaux sensibles aux longueurs d'ondes du visible (nommés cellules PV). L'association de plusieurs cellules PV en série/parallèle donne lieu à un générateur photovoltaïque (GPV) ayant une caractéristique statique courant-tension I (V) non linéaire et présentant un point de puissance maximale (PPM). Cette caractéristique dépend du niveau d'éclairement et de la température de la cellule, ainsi que du vieillissement de l'ensemble. Le point de fonctionnement du GPV peut donc varier entre les points extrêmes correspondants au courant de court-circuit I_{cc} et la tension en circuit ouvert V_{oc} . La détermination du point de fonctionnement du GPV dépend directement de la charge à laquelle il est connecté et est plus ou moins éloigné du PPM caractérisé par le courant et la tension optimaux notés (I_{opt} , V_{opt}).

Bien que depuis les premières installations PV, les prix des GPV aient diminué fortement, le prix du kWh reste encore élevé. Dans ce contexte, encore aujourd'hui, la plupart des installations PV sont destinées à des sites isolés où la possibilité de raccordement au réseau est difficile voir impossible. Une manière d'optimiser le prix du kWh est d'extraire le maximum de puissance P_{max} disponible aux bornes du GPV. Pour cela, il est nécessaire d'introduire un étage d'adaptation entre le GPV et la charge DC dans le but d'imposer un point de fonctionnement du GPV le plus près possible du PPM.

Dans ce chapitre, nous allons présenter un bref rappel sur le concept de module et de l'influence des paramètres météorologique (flux lumineux, la température..) sur la caractéristique non linéaire des différents modules photovoltaïques lesquels devraient assurer un fonctionnement aux caractéristiques maximales quelque soient les comportements aléatoires de la source et de la charge avec un haut rendement. Nous allons également nous intéresser à l'optimisation de la puissance fournie par le générateur photovoltaïque par le biais d'un étage d'adaptation avec fonction de recherche du point de puissance maximale MPPT (hacheur MPPT). Par ailleurs, ce chapitre englobera une classification des principales commandes MPPT présentes dans la littérature, les critères de qualité de cette commande, ainsi que les différentes techniques intelligentes nouvellement introduites dans la commande MPPT à savoir : les réseaux de neurones artificiels, la logique floue, les systèmes neuro-flous et les algorithmes génétiques.

I.2. Le générateur photovoltaïque

I.2.1. Principe

Une cellule photovoltaïque est basée sur le phénomène physique appelé effet photovoltaïque qui consiste à produire une force électromotrice lorsque la surface de cette cellule est exposée à la lumière. La tension qui en résulte peut varier entre 0.3 V et 0.7 V en fonction du matériau utilisé et de sa disposition, ainsi que de la température de la cellule et du vieillissement de la cellule. La représentation schématique de base d'une cellule photovoltaïque est illustrée à la figure (I.1). [4]

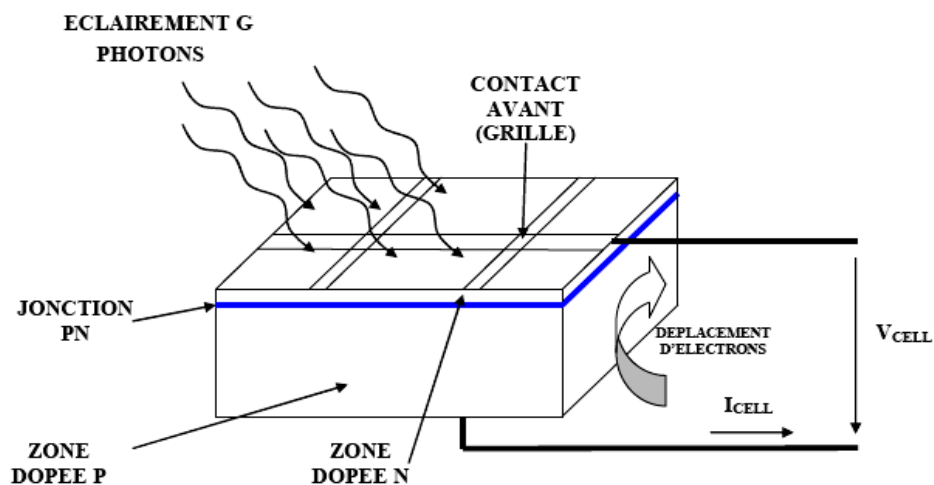


Figure I.1 : Coupe transversale d'une cellule PV typique.

Une cellule PV est réalisée à partir de deux couches de silicium, une dopée P (dopée au bore) et l'autre dopée N (dopée au phosphore) créant ainsi une jonction PN avec une barrière de potentiel. Lorsque les photons sont absorbés par le semi-conducteur, ils transmettent leur énergie aux atomes de la jonction PN de telle sorte que leurs électrons se libèrent et créent des électrons (charges N) et des trous (charges P). Ce qui crée une différence de potentiel entre les deux couches qui est mesurée entre les connexions des bornes positives et négatives de la cellule. La tension maximale de la cellule est d'environ 0.6 V pour un courant nul laquelle est nommée : tension de circuit ouvert (V_{OC}). Par ailleurs, le courant maximal se produit lorsque les bornes de la cellule sont court-circuitées et est appelé courant de court-circuit (I_{CC}) et dépend fortement du niveau d'éclairement. [4]

Chaque cellule peut être modélisée par le schéma électrique équivalent représenté à la figure (I.2).

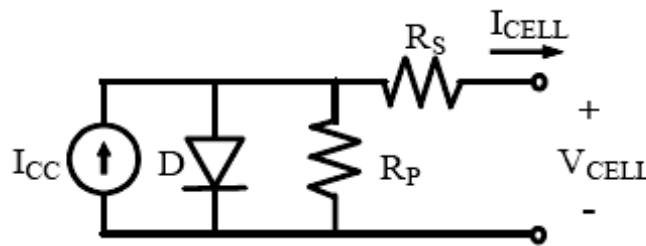


Figure I.2: Schéma équivalent électrique d'une cellule PV.

Les résistances R_s et R_p permettent de tenir compte des pertes liées aux défauts de fabrication. R_s représente les diverses résistances de contact et de connexion tandis que R_p caractérise les courants de fuite dus à la diode et aux effets de bord de la jonction.

I.3. Les performances du générateur photovoltaïque

Un module photovoltaïque est constitué d'un ensemble de cellules photovoltaïques élémentaires montées en série et/ou en parallèle afin d'obtenir des caractéristiques électriques désirées tels que : la puissance, le courant de court-circuit ou la tension en circuit ouvert. Un générateur photovoltaïque est constitué d'un ou plusieurs modules PV en série et / ou en parallèle pour obtenir une puissance, un I_{CC} et un V_{CO} désirés. [5]

I.3.1. Caractéristique Courant-Tension

La figure (I.3) représente la courbe $I = f(V)$ d'un module photovoltaïque typique dans des conditions constantes d'irradiation et de température.

L'irradiation standard adoptée pour mesurer la réponse des panneaux photovoltaïques est une intensité rayonnante de 1000 W/m^2 et une température de $25 \text{ }^\circ\text{C}$.

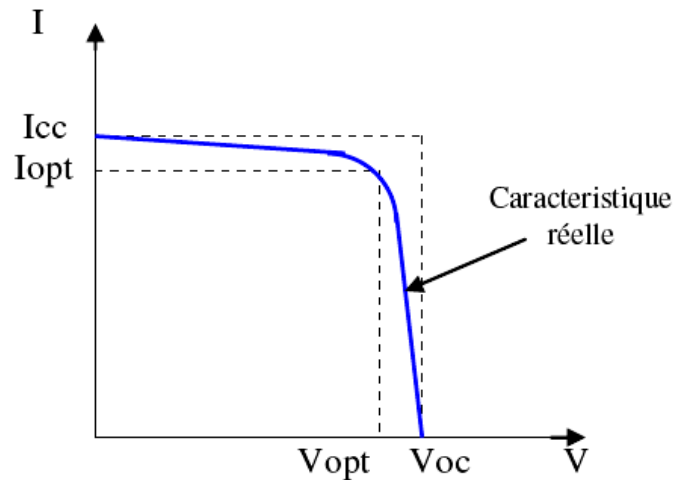


Figure I. 3 : Caractéristique $I = f(V)$ d'un module photovoltaïque.

Il est difficile de donner un caractère source de courant ou de tension à un panneau photovoltaïque sur toute l'étendue de la caractéristique courant-tension. Par conséquent, le panneau photovoltaïque est considéré comme une source de puissance avec un point P_m où la puissance se trouve être maximale. Il est donc intéressant de se placer sur ce point pour tirer le maximum d'énergie et ainsi exploiter au mieux la puissance crête installée. Il est important de noter que certains régulateurs solaires réalisent une adaptation d'impédance afin qu'à chaque instant on se trouve proche de ce point P_m . [5]

I.3.2. Influence de l'Éclairement

La figure (I.4) présente un exemple de courbes pour différents niveaux de rayonnement.

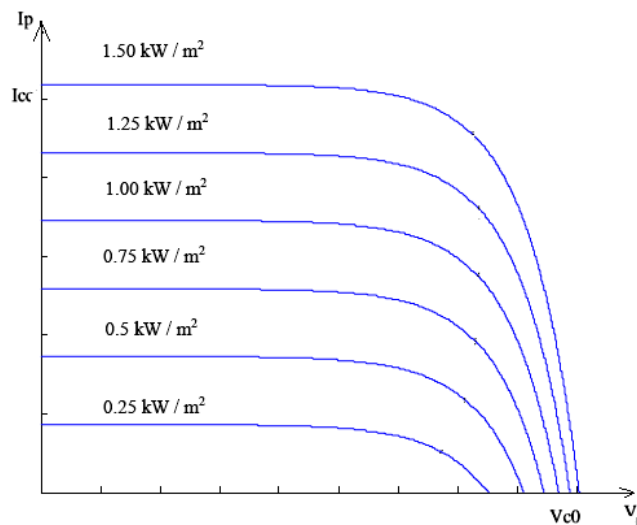


Figure I.4: Caractéristique d'un générateur photovoltaïque pour différents éclairements.

A partir de cette figure, il est clair que la valeur du courant de court-circuit est directement proportionnelle à l'intensité du rayonnement. Par contre, la tension en circuit ouvert ne varie pas dans les mêmes proportions, mais reste quasiment identique même à faible éclairement. [5]

I.3.3. Influence de la Température

La figure (I.5) présente des courbes courant-tension pour différentes températures de fonctionnement de la cellule PV.

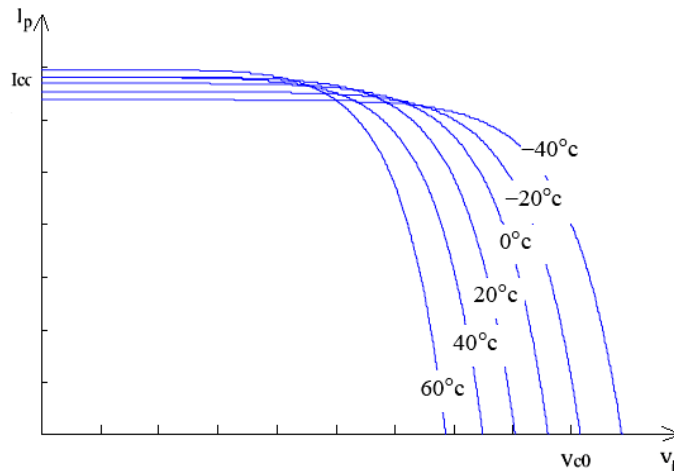


Figure I. 5 : Influence de la température sur la caractéristique électrique.

Nous remarquons que la température a une influence négligeable sur la valeur du courant de court-circuit. Par contre, la tension en circuit ouvert baisse assez fortement lorsque la température augmente, par conséquent la puissance extractible diminue. Lors du dimensionnement d'une installation, la variation de la température du site sera impérativement prise en compte. Il est important de savoir que la puissance du panneau diminue environ de 0,5% par chaque degré d'augmentation de la température de la cellule au dessus de 25 °C. [5]

Dans la figure (I.6), un exemple de réseau de caractéristiques puissance/tension d'un générateur photovoltaïque est illustré pour différentes valeurs de l'éclairement et de température auxquels est soumis le générateur. [3]

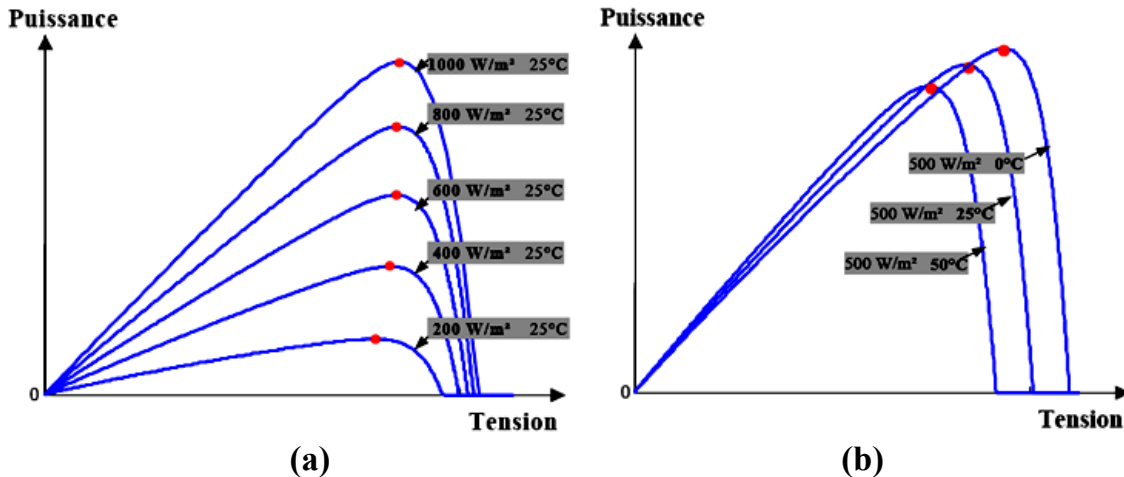


Figure I.6 : Exemple de réseau de caractéristiques puissance/tension d'un générateur photovoltaïque, (a) pour différents éclaircissements, (b) pour différentes températures de jonction.

Il est à noter que sur chacune des courbes, le point rouge représente le point de puissance maximale que peut fournir le panneau et que pour un éclaircissement et une température donnés, il existe une valeur de la tension qui maximise la puissance produite par le générateur. Ce qui nous contraint à insérer un convertisseur de puissance entre le générateur photovoltaïque et sa charge pour assurer un fonctionnement à puissance maximale quelles que soient la charge et les conditions d'éclaircissement et de température. [3]

I.4. Connexion directe source-charge

Actuellement, il reste encore beaucoup d'applications où une connexion directe entre un GPV et une charge est effectuée. Ce choix est principalement lié à la simplicité de l'opération et au très haut degré de fiabilité.

La configuration d'une connexion directe GPV-Charge par le biais d'une diode anti-retour est illustrée à la figure (I.7). Dans le cas où, la charge est une batterie, lorsque le module n'est pas éclairé, celui-ci pourrait fonctionner en tant que récepteur. La batterie pourrait donc se décharger sur le générateur PV et entraîner par conséquent un endommagement irréversible. De ce fait, pour éviter cette situation, la connexion doit être assurée à l'aide d'une diode anti-retour placée entre le GPV et la charge. L'inconvénient de cette configuration, c'est qu'elle n'offre aucun type de limitation et/ou réglage de la tension de la batterie. Le transfert de P_{max} disponible aux bornes du GPV vers la charge n'est pas non plus garanti. [4]

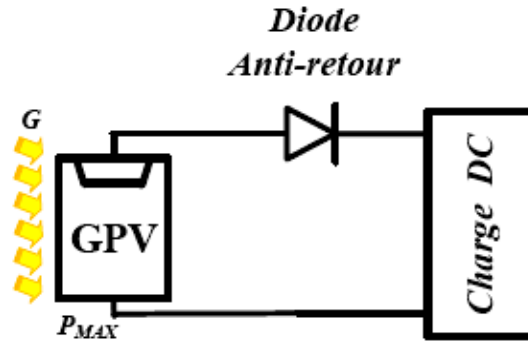


Figure I.7 : Connexion directe GPV-Charge par le biais d'une diode anti-retour.

En effet, le point de fonctionnement du GPV résulte de l'intersection entre la caractéristique I-V du GPV et la caractéristique I-V de la charge.

Il est à noter que dans ce type de connexion directe la nature de la charge est continue (DC) car une charge de type alternative n'est pas du tout compatible avec la connexion directe car le GPV fournit un courant continu. Pour la connexion d'un GPV à une charge alternative, nous avons obligatoirement besoin d'un étage d'adaptation spécifique de type onduleur.

Trois types de charges DC existent à savoir : charge purement résistive, charge de type source de tension et charge de type source de courant.

Sur la figure (I.8), nous avons représenté les caractéristiques I (V) et P (V) d'un GPV ainsi que les caractéristiques I (V) pour les trois types de charges. Le point de fonctionnement où la puissance fournie par le générateur est maximale (P_{MAX}) pour un courant optimal (I_{OPT}) et une tension optimale (V_{OPT}) est le point d'intersection entre les caractéristiques I (V) du générateur et celles des trois types de charges :

- 1) point A pour une charge résistive,
- 2) point B pour une charge de type source de tension,
- 3) point C pour une charge en source de courant.

Pour ces trois points, la puissance fournie par le générateur est respectivement PA' , PB' et PC' , correspondant à des valeurs de puissance inférieures à la puissance maximale disponible P_{MAX} . Ce qui nous permet de dire qu'une perte d'une partie de la puissance délivrable aux bornes du générateur PV entraînant à la longue des pertes de production énergétiques importantes.[4]

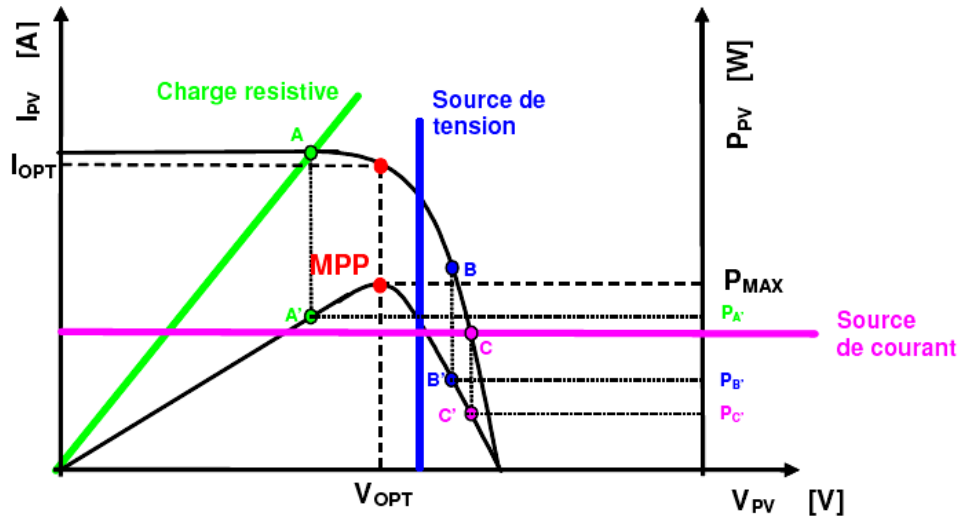


Figure I.8 : Points de fonctionnement d'un GPV en fonction de la charge en connexion directe.

I.5. Définition de l'étage d'adaptation entre (GPV) et une charge continue (Hacheur MPPT)

Comme nous l'avons vu dans la section 2.2, un GPV présente des caractéristiques $I(V)$ non linéaires avec des PPM. Ces caractéristiques dépendent entre autre du niveau d'éclairement et de la température de la cellule. De plus, selon les caractéristiques de la charge sur laquelle le GPV débite, nous pouvons trouver un très fort écart entre la puissance potentielle du générateur et celle réellement transférée à la charge en mode connexion directe.

Afin d'extraire à chaque instant le maximum de puissance disponible aux bornes du GPV et de la transférer à la charge, la technique la plus usuelle consiste à utiliser un étage d'adaptation entre le GPV et la charge (fig. I.9). Cet étage joue le rôle d'interface entre les deux éléments en assurant à travers une action de contrôle le transfert du maximum de puissance fournie par le générateur pour qu'elle soit la plus proche possible de P_{MAX} disponible. [4]

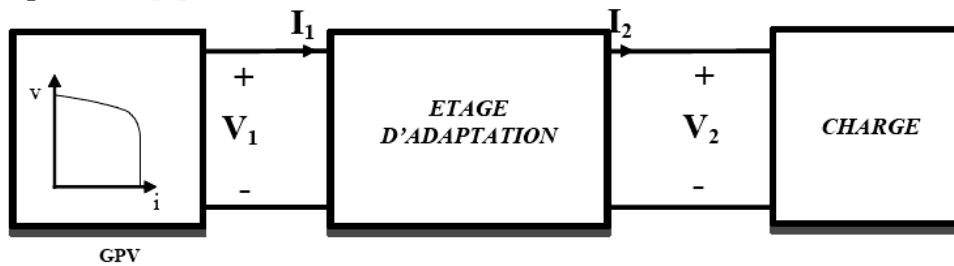


Figure I.9 : Etage d'adaptation jouant le rôle d'interface de puissance entre un GPV et une charge pour le transfert de P_{MAX} du GPV.

En résumé nous pouvons considérer que le système photovoltaïque n'est autre qu'un GPV et un étage d'adaptation avec fonction MPPT qui assure les fonctions de recherche du PPM et de protection de l'ensemble et dont la répartition des fonctions nécessaires au système photovoltaïque est représentée à la figure (I.10).

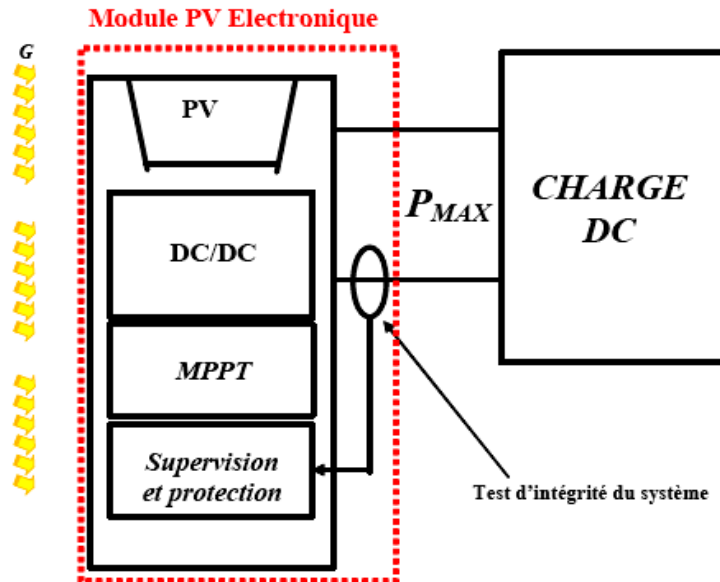


Figure I.10 : Schéma de principe d'un système photovoltaïque.

I.6.Principe de la recherche du point de puissance maximale (fonctionnement optimal du générateur photovoltaïque)

La chaîne de puissance d'un GPV où une charge DC est alimentée par un générateur à travers un convertisseur statique (CS) commandé par une commande MPPT peut être représentée comme indiquée sur la figure (I.11) et dont le rôle est de faire varier le rapport cyclique du CS de telle sorte que la puissance fournie par le GPV soit le P_{MAX} disponible à ses bornes. L'algorithme MPPT peut être plus ou moins compliqué pour rechercher le PPM et en général est basé sur la variation du rapport cyclique du hacheur dans l'objectif d'atteindre le PPM en fonction des évolutions des paramètres d'entrée du CS (I_{PV} et V_{PV}). Dans la section suivante, nous présentons un bilan sur les différents types d'algorithmes MPPT présents dans la littérature. [4]

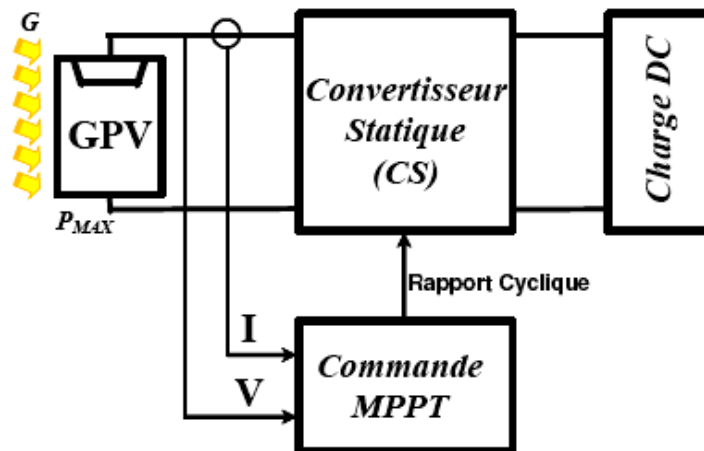


Figure I.11 : Chaîne élémentaire de conversion photovoltaïque avec CS contrôlé par une commande MPPT sur charge DC.

Dans notre cas l'étage d'adaptation adopté est un hacheur MPPT qui est un dispositif d'asservissement de la puissance couramment appelé à pilotage MPPT. Les hacheurs les plus couramment utilisés sont : le Buck, le Boost ou le Buck-Boost. Le choix de la structure de conversion est effectué en fonction de la charge DC à alimenter. Par exemple, si nous supposons que la charge est une batterie au plomb, les plages de tension de charge et de décharge de celui-ci établissent ou déterminent la structure la plus adéquate.

I.7. Classification des commandes MPPT

Nous pouvons classer d'une manière générale les commandes MPPT selon le type d'implémentation électronique : analogique, numérique....etc. [1]

I.7.1. Les méthodes analogiques

En les comparant aux méthodes micro programmées, les méthodes analogiques sont souvent simples à réaliser et à bas prix. Parmi ces dernières, nous rappelons quelques méthodes souvent utilisées :

A) la première méthode consiste à commander un convertisseur à travers un circuit électronique utilisant des cellules PV étalons comme consigne de référence afin de déterminer la tension optimale du générateur PV ou le courant optimal. Malgré le fonctionnement à une puissance maximale, quelques inconvénients sont à soulever tels que :

- La non prise en compte des dérives des caractéristiques du générateur PV dues aux paramètres extérieurs (vieillessement, vent, température, poussière),
 - Dépendance totale de la cellule étalon qui peut subir des modifications ou anomalies,
- B) Une deuxième méthode analogique se basant sur la recherche extrême analogique de la puissance maximale. Cette méthode est basée sur la dérivation de la puissance P du générateur. L'action de la commande du convertisseur permet de changer la résistance apparente vue du générateur et par conséquent on aura une variation de la puissance débitée par ce dernier. Le générateur travaille à son optimum de puissance lorsque le signal dérivé dp/dt s'annule. Cette méthode, quoique théoriquement séduisante et facile à étudier présente par contre des difficultés de mise en œuvre et ceci pour différentes raisons :
- Nécessité de dériver le signal analogique $P(t)$ ce qui engendre des bruits et des dérives importantes difficiles à filtrer,
 - Instabilité due aux bruits et aux perturbations,
 - Nécessité de tenir compte de la dynamique de mesure de puissance,
- C) La troisième méthode concerne celle de modulation par détection synchrone. Elle consiste à superposer, en série avec le générateur une source de tension sinusoïdale de faible amplitude. La puissance débitée par le générateur sera alors modulée d'une manière sinusoïdale et la phase de la modulation nous renseigne sur la position du point de fonctionnement par rapport au point de puissance maximale. Le positionnement du point de fonctionnement se fera aussi par incrémentation ou décrémentation de la tension aux bornes du générateur PV. [1]

1.7.2 Méthodes micro programmées

L'évolution du domaine de la micro-informatique a permis d'améliorer les conditions de fonctionnement des systèmes photovoltaïques. Parmi ces conditions, nous citons les procédures micro programmées permettant un fonctionnement du système à puissance maximale. L'élaboration de ces procédures se base sur plusieurs principes tels que :

- a) l'utilisation d'une base de données climatique permettant de déterminer pour chaque température et éclairement la puissance maximale de référence prévue.
- b) La simulation des équations caractéristiques du générateur PV pour chercher chaque nouvelle référence (puissance référence).
- c) l'utilisation des techniques de recherches en temps réel du point de fonctionnement à puissance maximale suivant les variations des conditions climatiques.

Les deux premières méthodes nécessitent un modèle de la source PV ainsi que des mesures de la température et de l'éclairement. Quant à la troisième, on n'a besoin ni du modèle de la source ni de la température ni de l'éclairement. Cependant cette dernière méthode nécessite la mesure de la tension et du courant du générateur PV en temps réel.

Une procédure de commande micro-programmée calcule la puissance délivrée par le générateur PV ainsi que sa dérivée puis élabore une commande du système qui ne se stabilise que dans le cas d'une dérivée nulle de la puissance correspondant ainsi à une puissance maximale. [1]

1.7.3. Recherche extrémale adaptative

1.7.3.1. Asservissement de la tension du générateur PV

Tension de référence fixe Cette méthode suppose que les variations de l'éclairement et de la température au niveau du générateur PV entraînant des variations insignifiantes de la tension optimale, et qu'une tension constante de référence constitue une approximation adéquate du point de puissance maximale. La tension aux bornes du générateur PV est prise variable à contrôler. La régulation de cette tension permet de maintenir un fonctionnement du système proche du point de puissance maximale. Malgré la simplicité de la mise en œuvre de cette méthode, cette dernière présente quelques inconvénients :

- Elle néglige l'effet de l'éclairement et de la température sur le générateur PV.
- Elle ne tient pas compte des dérives des caractéristiques du générateur dues aux paramètres extérieurs (vieillessement, vent, occultation).
- Elle ne peut pas être utilisée pour la régulation des systèmes à stockage d'énergie.

C'est pour cela que cette méthode de régulation est bien adaptée aux applications où les conditions climatiques sont stables telles que les systèmes satellites. [1]

Tension de référence variable La caractéristique statique d'un générateur photovoltaïque dans le plan $[P, V]$ présente un maximum. En première approximation, le lieu des points de fonctionnement optimal lorsque l'éclairement varie correspond à une tension constante V_p aux bornes du générateur.

La méthode d'adaptation consiste à réguler cette tension et à asservir la consigne en fonction de la variation de température des cellules. Malgré sa simplicité, cette méthode présente aussi quelques inconvénients :

- Elle ne tient pas compte des dérives des caractéristiques du générateur dues aux paramètres extérieurs (vieillessement, vent, poussière etc.).
- Difficulté de mesurer avec précision la température de jonction de la cellule,
- Nécessité de refaire le calcul de la boucle d'asservissement lorsqu'on change de générateur. [1]

I.7.3.2. Asservissement de la puissance du générateur PV

Cette méthode est basée sur la recherche du point de fonctionnement optimum par le principe dit de perturbation et observation ou ‘aveugle’. Un microprocesseur compare entre deux instants d’échantillonnage, la puissance débitée par le générateur et élabore une consigne en fonction du signal d’écart. Dans cette approche, on maximise la puissance délivrée à la charge et non la puissance permettant un rendement maximal de cette charge. [1]

❖ Méthode de perturbation et observation

Cette méthode a la particularité d’avoir une structure de régulation simple, et peu de paramètre de mesure. Elle opère en perturbant périodiquement la tension du panneau, et en comparant l’énergie précédemment délivrée avec la nouvelle après perturbation. Si la perturbation (l’ajout d’un dV de tension) implique une augmentation de la puissance alors nous nous trouvons dans la phase ascendante de la courbe de puissance, donc la tension de sortie devra être augmentée (et inversement). Dans ces conditions le tracker cherche en permanence le maximum de puissance.

❖ Algorithme d’incrémental de l’inductance

Cette méthode utilise la conductance incrémentale de la source comme MPPT. Cette méthode est plus efficace que la méthode de perturbation, et indépendante des caractéristiques des différents composants utilisés. Les tensions et courants du panneau sont monitorés, de telle manière que le contrôleur peut calculer la conductance et la conductance incrémentale, et décider de son comportement. Cet algorithme comportant un nombre important de calculs de dérivées.

I.8. Critères de qualité d’une commande MPPT

Comme nous l’avons vu dans les paragraphes précédents, il existe une grande quantité de commandes MPPT dans la littérature. Par contre, dans la plupart des cas, les performances de ces commandes ne sont pas fournies ou bien, elles le sont simplement pour un certain point d’opération avec une puissance donnée et pas pour une journée complète de mesures. Sans ces critères d’évaluation, il est difficile de savoir si un gain est réellement apporté en raison de l’utilisation d’une commande MPPT d’un type ou d’un autre. Pour ces raisons, nous avons défini certains critères de qualité qualifiant une commande MPPT en régime établi et en régime dynamique.

Dans un premier temps, la commande MPPT doit avoir un niveau de simplicité important favorisant une faible consommation et un coût raisonnable. En effet, il ne faut

pas oublier que dans un contexte énergétique, le fait d'insérer un étage d'adaptation avec une commande MPPT doit avoir un gain du point de vue énergétique recouvrant le surcoût économique. Dans le cas contraire, la commande aussi performante qu'elle soit, n'est pas recevable pour l'utilisation d'un tel type de commande.

De plus, concernant ses performances, la commande MPPT doit avoir un bon comportement en dynamique et en statique pour piloter l'étage d'adaptation auquel elle est associée et pour assurer que l'adaptation aux changements d'éclairement soit faite le plus rapidement possible. En même temps, elle doit pouvoir piloter l'étage d'adaptation de telle sorte qu'en régime statique, le point de fonctionnement du GPV soit le plus près possible du PPM dans n'importe quelle condition météorologique ou état de la charge DC alimentée par l'étage d'adaptation.

En résumé, l'utilisation d'un type de commande MPPT par rapport à un autre ou bien à une connexion directe doit apporter un gain énergétique et économique quantifiable.[4]

Malheureusement, il n'existe pas encore de standard international qui définisse comment il faut mesurer les performances d'une commande MPPT.

I.9. Techniques intelligentes pour la commande MPPT

Dans la partie précédente, les méthodes classiques et les mécanismes de poursuite de puissances maximales les plus rencontrées dans la littérature ont été présentés. Cette partie quant à elle, décrit des techniques intelligentes nouvellement introduites dans le monde de contrôle des MPPT. Il s'agit principalement des réseaux de neurones artificiels et les différentes structures de commande qui leurs sont associées, la logique floue, les systèmes neuro-flous et, à un degré moindre, les algorithmes génétiques lesquels font actuellement leur apparition dans la commande électrique.

I.9.1. Commande par les algorithmes génétiques

Les algorithmes génétiques sont des algorithmes d'optimisation stochastiques fondés sur les mécanismes de la sélection naturelle et de la génétique, inspirés de la théorie de survie de Darwin. En raison des performances remarquables des AGs, ces derniers ont envahi plusieurs domaines de recherche dans lesquels ils ont apporté des satisfactions appréciables en raison de leurs avantages à savoir : la rapidité et la possibilité de résoudre des équations non linéaires à plusieurs variables.

Les algorithmes génétiques comptent parmi les approches intelligentes utilisées pour la poursuite du point de puissance maximale (PPM) des cellules photovoltaïques pour pouvoir ainsi profiter au maximum de l'énergie solaire et suivre rapidement ce point. Beaucoup de travaux ont montrés que les AGs donne des résultats très intéressants dans lesquels la convergence est assurée, avec des temps de calculs et des opérations simples.

Néanmoins, les algorithmes génétiques présentent un inconvénient qui réside dans l'utilisation des relations de probabilité rendant ainsi le résultat parfois imprévisible et les performances moins contrôlées. [7], [18]

I.9.2. Commande par la logique floue

La commande par logique floue a été appliquée avec succès pour extraire le point maximum de puissance dans des systèmes de conversion d'énergie, et pour transférer l'énergie électrique maximum disponible à partir d'un PVG à une charge. [16], [17]

I.9.3. Commande par réseau de neurones

Les réseaux de neurones artificiels (ANN) sont des modèles électroniques basés sur la structure neuronale du cerveau lesquels sont souvent utilisés pour l'optimisation de l'algorithme de MPPT. En effet les réseaux de neurone peuvent être utilisés afin de trouver la position du point maximum de puissance avec un nombre réduit d'itérations pour atteindre le MPP et une réduction au minimum des oscillations autour de ce dernier.

I.9.4. Commande par réseaux neuro-flous

En outre, les réseaux neuro-flous ont été employés dans le même contexte. Énonçons par exemple l'étude présentée par Della et autre, ce qui mène à un cheminement optimal en ligne de point d'opération d'un PVG qui alimente un moteur à induction par l'intermédiaire d'un onduleur de tension de PWM.

I.10. Conclusion

Dans ce chapitre, nous avons mis en évidence que la conception de systèmes photovoltaïques optimisés est par nature difficile. En effet, côté source, pour un générateur photovoltaïque, la production de puissance varie fortement en fonction de l'éclairement, de la température, mais aussi du vieillissement global du système. Côté charge, que ce soit de nature continue (DC), ou bien alternative (AC), comme dans le cas du réseau électrique, chacune a son comportement propre qui peut être aléatoire. En effet il a été montré pour qu'un tel générateur fonctionne le plus souvent possible dans son régime optimal, la solution communément adoptée est alors d'introduire un convertisseur statique qui jouera le rôle d'adaptateur source-charge. La commande MPPT permet de faire fonctionner un générateur photovoltaïque de façon à produire en permanence le maximum de sa puissance. Ainsi, quels que soient les conditions météorologiques (température et irradiation), et

quelque soit la tension de la batterie, la commande du convertisseur place le système au point de fonctionnement maximum (V_{OPT} - I_{OPT}),

Dans ce contexte plusieurs méthodes de poursuite et algorithmes ont été mis en application, chacun présente des avantages comme des inconvénients.

Le chapitre suivant sera consacré à l'étude des réseaux de neurones, la logique floue ainsi que l'approche neuro-floue.

Chapitre II

Concepts des réseaux de neurones et de la logique floue

II.1.Introduction

Nous présentons dans ce chapitre certains modèles de réseaux de neurones artificiels (RNA) qui intègrent des aspects issus de la logique floue. Pour décrire ce type de modèle, la terminologie utilisée est *systèmes hybrides neuro-flous*.

Dans ce chapitre, nous présenterons les principes et concepts de base des réseaux de neurones, ceux de la logique floue et neuro-floue.

II.2.Les réseaux de neurones

Le terme "*Réseau de neurones artificiels*" fait rêver beaucoup de chercheurs soucieux d'exploiter au maximum les données dont ils disposent. Cet outil s'inspire essentiellement du processus de traitement de l'information effectué par le cerveau humain.

Dans cette section, nous aborderons dans un premier temps les principes et les définitions des réseaux de neurones, puis la conception des RNA à savoir : choix de la topologie, la paramétrisation,...etc. ainsi que leurs limitations.

II.2.1.Présentation générale

II.2.1.1.Le neurone biologique et le neurone formel

Les réseaux connexionnistes sont des assemblages fortement connectés d'unités de calcul, appelés les *neurones formels* lesquels ont pour origine un modèle du *neurone biologique*, dont ils ne retiennent d'ailleurs qu'une vision fort simplifiée (Fig.II.1 et Fig.II.2). Le neurone, comme toute cellule, est composé d'un corps (ou *soma*), qui contient son noyau et où se déroulent les activités propres à sa vie cellulaire. Cependant, il est aussi doté d'un *axone* et de *dendrites*, considérées comme des structures spécialisées dans la communication avec les autres neurones. Cette communication entre cellules nerveuses s'effectue via des impulsions nerveuses générées à l'extrémité somatique de l'axone et qui se propagent vers les terminaisons axonales où elles affecteront tous les neurones reliés au neurone générateur, par l'intermédiaire de jonctions entre les terminaisons axonales et les autres cellules. Cette jonction est appelée *synapse*. [27]

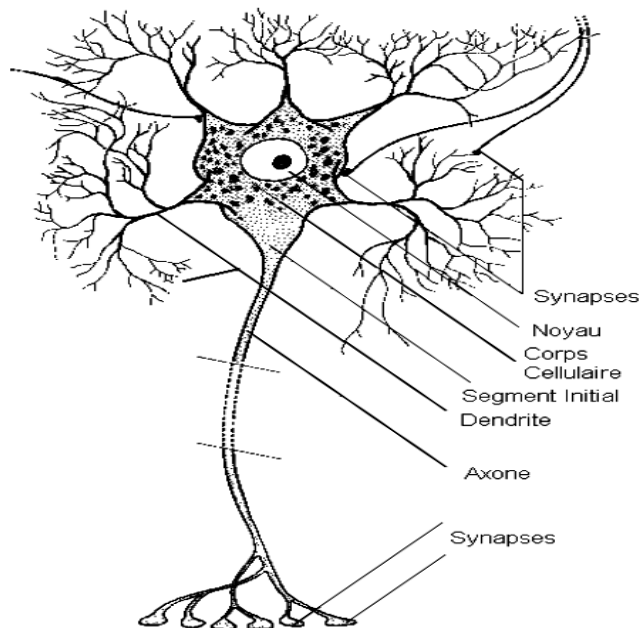


Figure.II.1. Structure d'un réseau de neurones biologiques.

Cet héritage de la neurobiologie forme une composante importante de l'étude des réseaux connexionnistes, et le souci de maintenir une certaine correspondance avec le système nerveux humain a animé et continue à animer une part importante des recherches dans ce domaine. Malgré cet héritage, l'essentiel des travaux réalisés aujourd'hui ont pour objet, les réseaux de neurones formels et non leur corrélat neurobiologique. Par ailleurs, et considérés comme des systèmes de calcul, les réseaux de neurones possèdent plusieurs propriétés qui les rendent intéressants d'un point de vue théorique et fort utile en pratique.

Un réseau connexionniste est constitué par un graphe orienté et pondéré, dans lequel les noeuds sont des automates simples nommés *neurones formels* ou tout simplement *unités du réseau*. Ces unités sont dotées d'un *état interne*, que l'on appelle *état*

d'activation dont le rôle est de propager son état d'activation aux autres unités du graphe en passant par des arcs pondérés appelés *connexions*, *liens synaptiques* ou bien *poids synaptiques*. La règle qui détermine l'activation d'un neurone en fonction de l'influence venue de ses entrées et de leurs poids respectifs s'appelle *règle d'activation* ou *fonction d'activation*. Les *changements* apportés aux valeurs *des poids synaptiques* ou dans la structure d'interconnexion des unités du réseau sont responsables des changements de comportement d'activation du réseau, lui permettant d'*apprendre* un nouveau comportement. En effet, le réseau est capable d'établir des associations entrée-sortie (stimulus et réponse) afin de bien résoudre un problème, par ce qu'on appelle règle d'apprentissage. [27]

II.2.1.2.Modélisation

Le modèle de neurone formel est un modèle mathématique très simple dérivé de l'analyse de la réalité biologique dont la structure est donnée par la figure (II.2). [20]

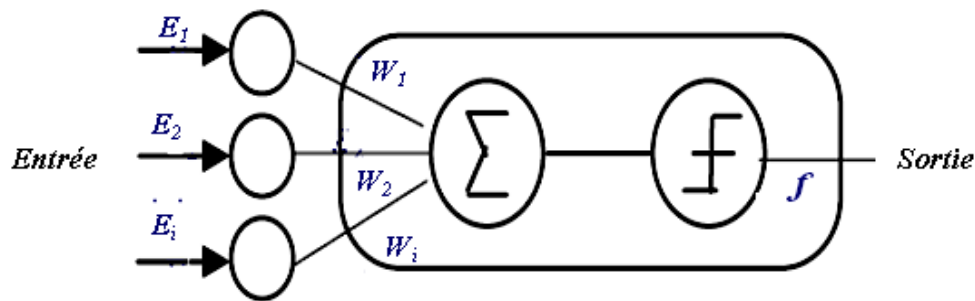


Figure II. 2 : Schéma général d'un neurone artificiel.

La formulation mathématique de ce neurone est décrite par la relation suivante:

$$S = f\left(\sum_{j=1}^i W_j E_j\right) \quad (\text{II.1})$$

Où

S : est la sortie du neurone

E_i : est la $i^{\text{ème}}$ entrée du neurone, elle caractérise la donnée communiquée par le capteur (test de détection) ou par l'expert (diagnostic et base de faits et de connaissance).

f: fonction d'activation (la sigmoïde)

W_i : poids de connexions avec les entrées.

Différents types de fonction d'activation peuvent être utilisés à savoir :

- La fonction Heaviside (Fig.3.a).
- La fonction signe (Fig. 3.b).
- La fonction linéaire à seuil (Fig.3.c).
- La fonction linéaire (Fig. 3.d).
- La Fonction sigmoïde (Fig.3.e). [19]

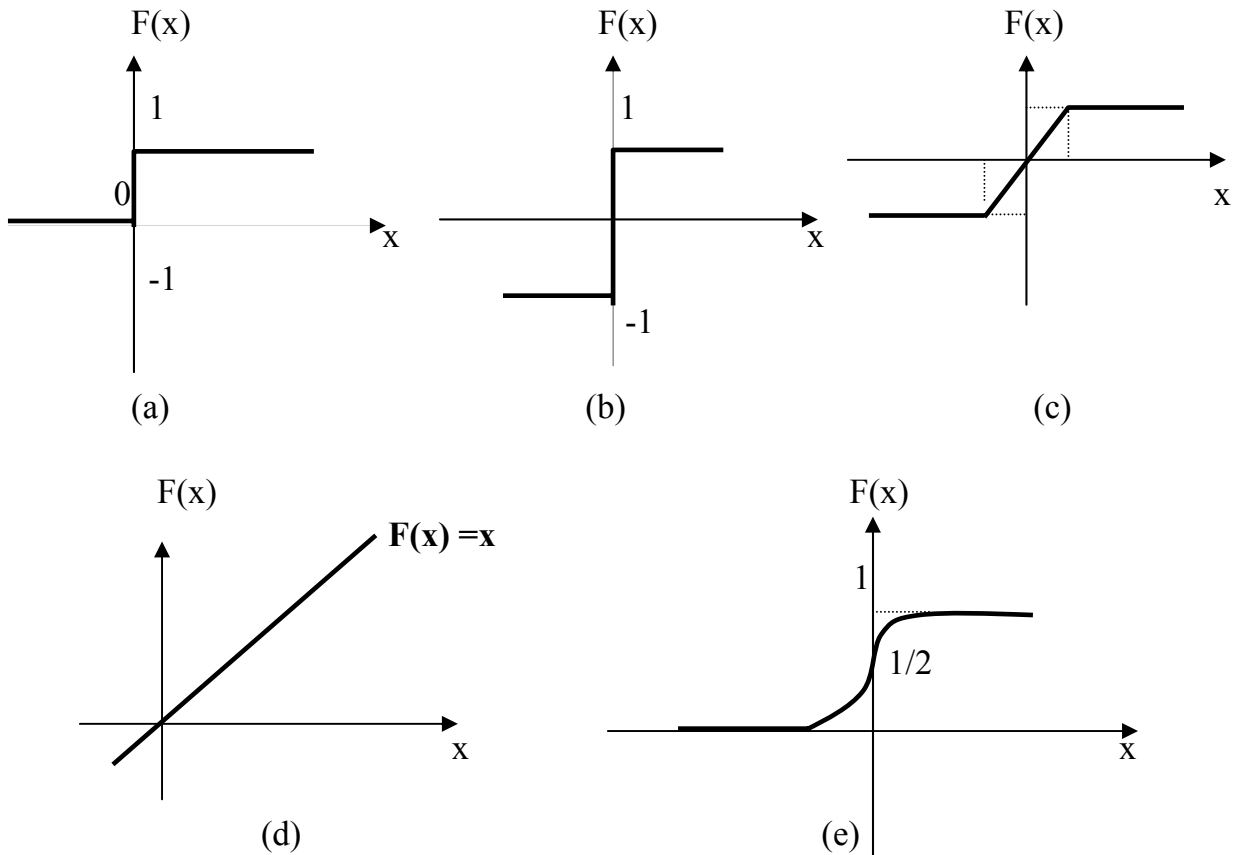


Figure.II.3 : Les différentes fonctions d'activation pour les neurones.

II.2.2. Classification et propriétés des RNA

La grande quantité de modèles connexionnistes existants, a conduit d'abord les chercheurs à les classer selon leurs différentes propriétés afin de mieux comprendre les avantages et les inconvénients du choix d'un modèle plutôt que l'autre. En effet, il existe plusieurs modèles de réseaux de neurones établis, selon les attributs adoptés, tels que : le type d'apprentissage, l'architecture d'interconnexion des unités du réseau, la forme pour traiter et représenter les données, etc. [27]

II.2.2.1. Apprentissage Connexionniste

L'apprentissage est un processus graduel, itératif, où les poids du réseau sont modifiés plusieurs fois selon une règle d'apprentissage avant d'atteindre leurs valeurs finales. Les approches d'apprentissage connexionniste peuvent être réparties en trois grandes classes, selon le degré de contrôle donné à l'utilisateur. [27]

1) Apprentissage Supervisé

L'utilisateur dispose d'un comportement de référence précis qu'il désire faire apprendre au réseau lequel est censé être capable non seulement de mesurer la différence entre son comportement actuel et le comportement de référence mais ainsi de corriger ses poids de façon à réduire cette erreur. Ainsi, un tel type d'apprentissage supervisé utilise des connaissances empiriques, habituellement représentées par des ensembles d'exemples étiquetés par la classe à laquelle ils appartiennent.

2) Apprentissage Semi-Supervisé (reinforcement learning)

L'utilisateur ne possède que des indications imprécises (par exemple, échec/succès du réseau) sur le comportement final désiré. En effet, souvent nous ne disposons que d'une évaluation qualitative du comportement du système réel.

3) Apprentissage non supervisé

Les poids du réseau sont dans ce cas modifiés en fonction de critères internes comme la coactivation des neurones. Les comportements qui en résultent sont en général comparables à des techniques d'analyse de données appelées aussi *auto organisation*.

L'apprentissage connexionniste nécessite en général une grande quantité de données que l'on regroupe dans un *ensemble d'exemples* dits *d'apprentissage*. Selon la technique utilisée, d'autres ensembles de données sont également employés, notamment pour mesurer la validité de la solution trouvée par le réseau appelé *ensembles d'exemples de test* ou de *généralisation*. Cette étape est généralement effectuée afin de tester la capacité du réseau à réagir correctement lorsqu'on lui présente des entrées qui n'ont pas été vues lors du processus d'apprentissage. Dans la pratique, ce sont les *capacités de généralisation* d'un réseau connexionniste qui vont établir les possibilités de son application à différents problèmes.

Il est cependant important de noter qu'un réseau peut trop se spécialiser à un ensemble d'apprentissage, ce qui va probablement mener à un problème de mauvaise généralisation, ou ce qu'on appelle sur-apprentissage (*overfitting*). Dans ces conditions, une façon d'éviter un tel problème c'est d'utiliser de la procédure de la validation croisée (*cross validation*) qui consiste à diviser les données en deux ensembles distincts : un ensemble d'apprentissage et un ensemble de test.

L'apprentissage d'un ensemble de données peut être réalisé de différentes façons, selon la manière dont le réseau est alimenté par les données :

- *Apprentissage Instantané*

L'ensemble de données d'apprentissage est analysé une seule fois et permet d'établir les poids du réseau d'un seul coup. Ce mode d'apprentissage requiert un seul passage sur les données et est également appelé *one-shot-learning*. Cette méthode n'est pas très utilisée.

- *Apprentissage par Paquets*

L'ensemble des données d'apprentissage est présenté au réseau plusieurs fois de façon à optimiser les poids synaptique et minimiser de ce fait l'erreur en sortie. Chaque présentation de l'ensemble complet de données d'apprentissage est appelée une époque (*epoch*).

La procédure d'apprentissage est réalisée par un algorithme d'apprentissage dont le rôle est de réduire petit à petit l'erreur de sortie à chaque présentation des données de sorte que l'état du réseau converge vers la solution du problème.

Il existe plusieurs types d'algorithme d'apprentissage :

- *Apprentissage Continu*

Ce processus fait appel à un algorithme d'apprentissage qui prend en compte et au fur et à mesure les exemples qui lui arrivent d'où son appellation (*continuous/on-line learning*). Cette méthode est aussi appelée apprentissage incrémental (par rapport aux données), cependant, afin d'éviter toute confusion avec l'apprentissage incrémental dans le cas où des unités sont ajoutées à la structure du réseau, nous allons utiliser plutôt le terme apprentissage continu. Un des principaux problèmes de l'apprentissage continu réside dans la difficulté de trouver un bon compromis entre "adaptation et stabilité", ce qui peut amener le réseau à 'oublier' les données apprises (trop d'adaptation) ou bien

d'être incapable de s'adapter aux nouvelles données qui lui sont présentées (trop de stabilité).

- *Apprentissage Actif*

Cette approche est basée sur l'idée que l'algorithme d'apprentissage peut lui aussi imposer des choix par rapport aux données à utiliser.

Dans ce cas l'apprentissage passe d'un comportement passif, où le réseau ne peut pas intervenir sur les données présentées à son entrées, à un comportement actif, pour lequel il peut déterminer quelles données vont être prises en compte et aussi dans quel ordre elles doivent être considérées. Il s'agit d'une voie de recherche relativement récente. [27]

II.2.2.2. Les règles d'apprentissage

Les trois types d'apprentissage présentés ont pour but comme il a été mentionné d'ajuster les poids de connexions entre les neurones en s'accordant certaines règles dont les plus répandues sont :

➤ **La règle de Hebb**

Cette règle a pour rôle de modifier la valeur des poids synaptiques en fonction de l'activité des unités qui les relient c'est-à-dire "dans le cas où deux unités s'activent en même temps la connexion qui les lie est renforcée (c'est une connexion excitatrice) sinon elle est affaiblie (c'est une connexion inhibitrice)". [25]

➤ **La règle delta**

Calcule la différence entre la valeur de sortie du réseau avec celle désirée et cela afin d'ajuster les poids synaptiques. Elle emploie une fonction d'erreur, nommée le moindre carré moyen, qui est basée sur les différences utilisées pour l'ajustement des poids. [25]

➤ **La règle d'apprentissage compétitive**

Elle ne concerne qu'un seul neurone. Dans ce cas les données sont regroupées en catégorie de sorte que les neurones similaires soient rangés dans une même classe en se basant sur des corrélations des données et sont par la suite représentées par un neurone unique.

L'architecture d'un tel réseau possède une couche d'entrée et une couche de compétition. Une forme est présentée à l'entrée du réseau et est projetée sur chacun des neurones de la

couche compétitive. Le neurone sélectionné est celui qui possède un vecteur de poids le plus proche de la forme présentée à l'entrée. D'autre part, chaque neurone de sortie est connecté aux neurones de la couche d'entrée et aux autres cellules de sortie (c'est une connexion inhibitrice) ou à elle-même (c'est une connexion excitatrice). La sortie dépend alors de la compétition entre les connexions inhibitrices et excitatrices. [25]

➤ **La règle de correction d'erreurs**

Elle s'effectue en trois étapes comme suit :

1. On commence avec des valeurs des poids de connexions qui sont pris au hasard.
2. On introduit un vecteur d'entrée de l'ensemble des échantillons pour l'apprentissage.
3. Si la sortie ou la réponse n'est pas correcte, on modifie toutes les connexions pour atteindre la bonne réponse. [25]

➤ **La règle de rétro propagation**

Inventée par Rumelhart, Hinton et Williams en 1986, utilisée pour ajuster les poids de la couche d'entrée à la couche cachée. Cette règle peut aussi être considérée comme une généralisation de la règle delta pour des fonctions d'activation non linéaire et pour des réseaux multicouches. Les poids dans le réseau de neurones dans ce cas sont au préalable initialisés avec des valeurs aléatoires. Par la suite, un ensemble de données qui vont servir à l'apprentissage sont considérés, sachant que chaque échantillon possède des valeurs cibles qui sont celles que le réseau de neurones doit à terme prédire lorsqu'on lui présente le même échantillon. [25]

II.2.2.3.Type des Unités

Les unités d'un réseau (neurones formels) peuvent être de différents types, et cela suivant la fonction interne utilisée pour calculer l'activation. Les principales différences résident dans le type de fonction d'activation employé (linéaire, sigmoïde exponentielle asymétrique, sigmoïde tangentielle symétrique, gaussienne), et dans la fonctionnalité des unités (unités à prototypes, unités du type Perceptron). En effet, selon la fonctionnalité des unités nous pouvons distinguer deux types majeurs de réseaux :

• **Les réseaux à Prototypes**

Ce type de réseau utilise des unités qui servent à représenter des prototypes d'exemples appris dont les unités ont une représentation interne regroupant les caractéristiques typiques d'un ensemble d'exemples. Ces réseaux ont normalement un apprentissage non-supervisé, mais ils peuvent aussi avoir un apprentissage supervisé

(avec un ou plusieurs prototypes associés à chaque classe). Un des avantages de ce type de réseau réside dans la possibilité de faire un apprentissage continu (incrémental), étant donné qu'il n'est pas très difficile de concevoir des méthodes qui permettent d'augmenter le réseau par ajout de nouveaux prototypes, lesquels sont aussi appelés des noyaux.

- **Les réseaux de type "Perceptron"**

Les unités du type "Perceptron" ont été proposées par Frank Rosenblatt en 1958. Il s'agit d'un des modèles d'unités les plus utilisés actuellement. Il est à l'origine de plusieurs autres réseaux de neurones avec apprentissage supervisé par correction d'erreur. Le modèle du Perceptron Multi-Couche (PMC) est devenu très connu, tout en étant associé à la règle d'apprentissage de la Rétro-Propagation.

II.2.2.4.Type d'Architecture du Réseau

La façon d'interconnecter les unités d'un réseau, peut également conduire à différentes architectures, entre autre :

- **Réseaux à une seule couche**

Dans ce type d'architecture, les unités sont toutes sur le même niveau et sont connectées directement aux entrées et sont aussi les sorties du réseau. Les réseaux à une seule couche ont normalement des connexions latérales (entre les neurones d'une même couche). Un exemple de ce type d'architecture sont les réseaux du type "Kohonen Feature Map".

- **Réseaux à couches unidirectionnels**

Les unités sont dans ce type de réseaux organisées en plusieurs niveaux bien définis, que l'on appelle des couches. Chaque unité d'une couche reçoit ses entrées à partir de la couche précédente et envoie ses sorties vers la couche suivante (*feed-forward nets*). Cette architecture à trois couches (entrée, couche cachée et sortie) est très utilisée dans la pratique. Le modèle du perceptron multicouche (PMC) se compose en général d'une architecture de ce type, c'est-à-dire avec une seule couche cachée (*hidden layer*), mais rien n'empêche d'avoir plus d'une seule couche cachée dans ce modèle. Un autre type d'interconnexions dans les réseaux à couches sont les raccourcis (*short-cuts*) qui permettent de lier des unités en passant à travers des niveaux. Ainsi, on peut 'passer' d'une couche à l'autre à travers des raccourcis, à condition de ne pas créer une boucle.

- **Réseaux récurrents**

Les réseaux récurrents peuvent avoir une ou plusieurs couches, mais leur particularité réside dans la présence d'interconnexions depuis la sortie d'une unité vers une autre de la même couche ou d'une couche inférieure. Ce type d'interconnexions permet de modéliser des aspects temporeux et des comportements dynamiques dans lesquels la sortie de chaque unité dépend de son état antérieur. Les boucles internes rendent ce type de réseaux instable, ce qui nous oblige à utiliser des algorithmes plus spécifiques (et usuellement plus complexes) pour l'apprentissage. Un type particulier de réseau récurrent sont les réseaux totalement connectés, tels que le modèle de Hopfield.

- **Réseaux d'ordres supérieurs**

Les unités de ce type de réseaux assurent la connexion directe entre deux ou plusieurs entrées et cela avant d'appliquer la fonction de calcul de l'activation. Ce type de réseau sert à modéliser les synapses de modulation, c.à.d, quand une entrée peut moduler ou agir sur le signal provenant d'une autre entrée. Un modèle particulier de réseaux d'ordre supérieur (*high-order neural net*) est le réseau *Sigma-Pi* proposé par Rumelhart.

De plus, l'architecture d'un réseau peut être aussi classée selon son évolution au cours de l'apprentissage et de son utilisation. En fonction de ce critère, les réseaux sont divisés en :

- **Structure statique**

Le réseau a sa structure définie avant l'apprentissage. La quantité de neurones, ainsi que sa structure d'interconnexions ne changent pas. Les seuls changements subis par le réseau se situent au niveau des poids synaptiques, modifiés au cours de l'apprentissage. Ce type de modèle pose une difficulté : la détermination du bon nombre de neurones et interconnexions à utiliser. Un réseau avec peu d'unités et d'interconnexions risque de ne pas pouvoir apprendre à résoudre correctement un problème et un réseau avec trop d'unités peut avoir des problèmes de convergence, de complexité et de généralisation. Il n'existe pas de méthode optimale qui permet de spécifier automatiquement le nombre exact d'unités à employer pour bien résoudre un problème donné. Les réseaux PMC à Rétro-Propagation classiques sont des réseaux statiques.

- **Structure dynamique**

Les réseaux ayant une telle structure sont des réseaux dont le nombre d'unités et d'interconnexions peut varier. Ils sont aussi appelés réseaux ontogéniques. Les

changements dans la structure du réseau peuvent être de type *génératif (incrémental)* ou de type *destructif (élagage)*.

II.2.2.5.Type d'Application du Réseau

Les réseaux de neurone peuvent être appliqués à plusieurs types de tâches, telles que : la reconnaissance de formes (reconnaissance de visages), la classification, la transformation de données (compression), la prédiction (prédiction de séries temporelles), le contrôle de processus et l'approximation de fonctions. Toutes ces tâches peuvent être regroupées en deux groupes principaux et cela selon le type des sorties fournies par le réseau et le comportement qui est recherché :

- **Réseaux pour l'Approximation de Fonctions**

Ce type de réseau doit avoir une sortie continue et est généralement employé pour l'approximation exacte (*interpolation*) ou pour l'approximation approchée d'une fonction représentée par les données d'apprentissage. Cette fonction acquise par le réseau permet de prédire les sorties étant données les entrées. On appelle ce type de problème, un problème de *régression*. En général, les fonctions représentées sont des fonctions avec des variables d'entrée et de sortie continues.

- **Réseaux pour la Classification**

Ce type de réseau doit attribuer des classes (valeur de sortie non continue) aux exemples qui lui sont fournis. La classification est un cas particulier de l'approximation de fonctions où la valeur de sortie est discrète et appartient à un ensemble limité de classes. Cet ensemble de classes peut être connu d'avance dans le cas de l'apprentissage supervisé.

Remarque :

Il est un peu délicat de classer précisément tous les différents modèles de réseaux de neurones dans une seule de ces deux classes. La plupart des modèles peuvent être adaptés pour être utilisés dans l'un ou l'autre type d'application. Mais certains modèles sont plus adaptés à une tâche de classification que d'approximation et vice versa.

Il existe donc de très nombreuses sortes de réseaux de neurones, nous nous intéressons à l'étude du réseau multicouche objet de notre travail.

II.2.3. Le perceptron multicouches (PMC)

C'est une extension du perceptron monocouche avec une ou plusieurs couches cachées entre l'entrée et la sortie. Les fonctions d'activation utilisées dans ce type de réseaux sont principalement les fonctions à seuil ou sigmoïdes.

Ce type de réseau peut résoudre des problèmes non linéairement séparables et des problèmes logiques plus compliqués, et notamment le fameux problème du XOR.

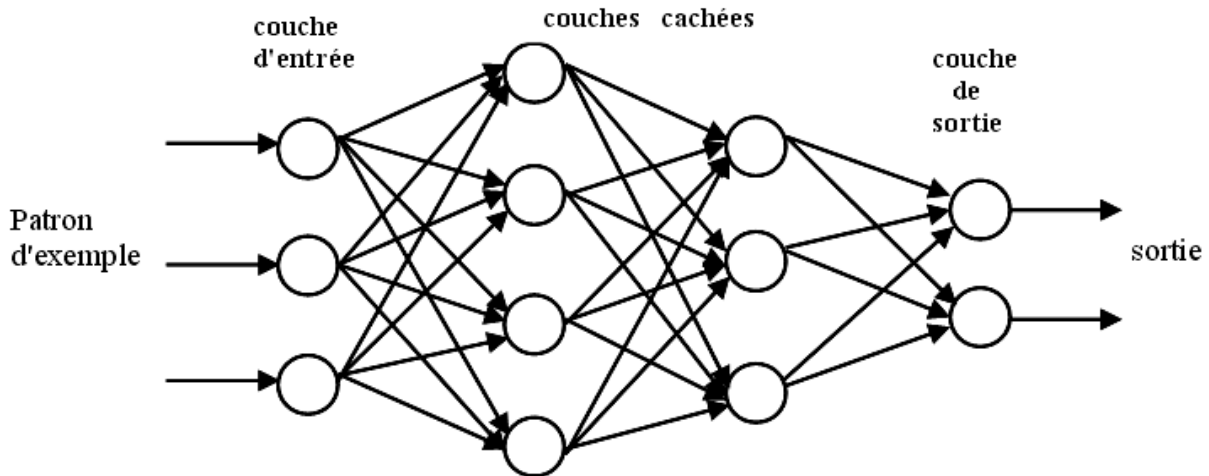


Figure.II.5: Architecture du perceptron multicouches.

Les réseaux PMC sont constitués par des unités organisées en couches à savoir: une couche d'entrée, une ou plusieurs couches cachées, et une couche de sortie. Les unités de la couche d'entrée reçoivent les signaux provenant de l'environnement. Ces signaux représentent les informations traitées par le réseau, telles que : la description de l'état du système, les valeurs des attributs associés aux variables d'entrée, les mesures obtenues à partir de capteurs externes ou les faits qui composent les prémisses d'un système expert. La couche de sortie rend disponible le résultat de l'activation du réseau à l'environnement extérieur. Enfin, les unités cachées, comme leur nom l'indique, n'ont pas d'interaction directe avec l'environnement. Dans ce cas les unités sont connectées selon une architecture sans boucles (connexions unidirectionnelles), mais on peut avoir des connexions inter-couches du type "raccourcis".

Les réseaux PMC utilisent une méthode d'apprentissage de type supervisé réalisé par un algorithme de minimisation sur l'erreur estimée à partir des sorties du réseau. L'emploi de cette méthode permet d'entraîner les unités de sortie du réseau, où l'erreur peut être obtenue directement par simple différence entre la valeur de sortie désirée et la valeur obtenue par le réseau ; mais il permet aussi d'entraîner les unités cachées grâce à une

technique de propagation en arrière de l'erreur à travers les couches du réseau, d'où le nom de "Rétro-Propagation" (*Back-Propagation Learning*). [19],

II.2.4. Certaines considérations pratiques pour le choix du réseaux de neurone adéquat

Avant de procéder à l'apprentissage d'un réseau de neurones artificiels, un certain nombre de paramètres doivent être fixés (dimensionnement du réseau, pas d'apprentissage..., etc.), au risque d'engendrer et de compromettre le processus d'entraînement dans le cas d'un mauvais choix. [28]

1. Choix et dimensionnement d'un réseau

D'une manière générale, il n'existe pas de méthodes systématiques pour la détermination du nombre de couches dans le réseau ou le nombre de neurones par couche. Néanmoins, il existe dans ce sens des règles pratiques tirées à partir de l'expérience, parmi lesquelles nous pouvons citer:

Règle 1 : le nombre de neurones par couche doit être d'autant plus élevé que la relation entre les données en entrée et les sorties désirées est complexe.

Règle 2 : pour les réseaux à une seule couche cachée, le nombre de neurones dans cette couche doit être égal à :

$$h = \frac{c}{10 (M + N)} \quad (\text{II.2})$$

Avec :

c : le nombre de vecteurs dans le fichier d'exemples.

M : nombre de neurones dans la couche de sortie.

N : nombre d'entrées du réseau.

Règle 3 : le nombre de couches cachées doit être égal au nombre de relations qui existent entre les neurones de la couche d'entrée et ceux de la couche de sortie.

2. Paramètre d'apprentissage ou pas de correction η

Le choix du paramètre d'apprentissage η a un effet important sur les performances du réseau. Généralement, il prend des valeurs petites de l'ordre de 0.05 à 0.25, afin d'assurer une certaine convergence. Ce paramètre présente en réalité un dilemme : s'il est trop grand, il peut entraîner des oscillations, s'il est trop petit, le temps d'apprentissage

devient infini. Par conséquent, ce paramètre η est choisi variable à partir d'une valeur grande qu'on diminue progressivement. [28]

3. Poids initiaux

Les poids initiaux sont choisis aléatoirement entre -0.5 et 0.5. Par ailleurs, il est recommandé de ne pas initialiser les poids synaptiques à une même valeur.

4. Minimums locaux

L'algorithme d'apprentissage peut éventuellement, converger vers un minimum local du critère d'erreur. Dans ce cas, l'erreur stagnera autour d'une valeur relativement grande, et les performances requises ne seront jamais atteintes. Pour remédier à ce problème, il existe plusieurs alternatives à savoir : changer le pas de correction, changer les poids initiaux, utiliser la rétro-propagation avec momentum, changer la fonction d'activation, augmenter le nombre de couches cachées ou bien augmenter le nombre de neurones par couche.

5. Saturation du réseau

Etant donné que, les fonctions d'activation dans un réseau de neurones sont généralement bornées, et fournissent des sorties entre 0 et 1 ou entre -1 et 1, il est de ce fait nécessaire de mettre à l'échelle les données présentées en entrée.

II.2.5. Paramétrage et conception d'un réseau de neurone

Pour paramétrer un réseau de neurone, nous devons déterminer les variables suivantes :

- Le nombre des neurones d'entrée qui correspond au nombre des variables du vecteur d'entrée utilisé pour résoudre le problème donné. Cependant il n'y a pas une suggestion systématique pour déterminer ce nombre. La sélection de ce paramètre doit être incluse dans le modèle de construction du processus. Idéalement on choisit un petit nombre pour les neurones d'entrée.
- Le nombre de couches cachées et le nombre de neurones cachés ; ils ont un rôle important pour l'application correcte des réseaux de neurones. Ce sont les neurones de la couche cachée qui prennent le modèle des données et effectuent une mise en correspondance non linéaire complexe entre les variables d'entrée et les variables de sortie.
- Le nombre d'échantillons d'apprentissage ; la plupart des scientifiques utilisent dans leurs réseaux de neurones seulement une couche cachée. Cependant il s'avère que dans ce cas, la couche cachée nécessite un nombre élevé de neurones cachés, ce qui n'est pas recommandé car cela provoque un temps d'apprentissage et une capacité de généralisation du réseau très mauvais.

En général pour déterminer le nombre des neurones cachés, on réalise des expériences et des essais et parmi eux on choisit le nombre qui provoque la plus petite erreur. [25]

Par ailleurs, réaliser une application, c'est d'abord exprimer le problème sous la forme d'une mise en correspondance de deux espaces, puis construire une base d'apprentissage représentative de données et enfin choisir, en se référant à son expérience, le modèle de réseau et ses paramètres. Il faut aussi préalablement définir les critères de mesure des performances (construction de la base de test), les prétraitements et le codage sur le réseau. [29]

Les quatre grandes étapes de conception d'un réseau de neurones sont :

1. Choix des échantillons

Cette étape aide le concepteur à déterminer le type de réseau le plus approprié pour résoudre son problème. La façon dont se présente l'échantillon conditionne : le type de réseau, le nombre de cellules d'entrée, le nombre de cellules de sortie et la façon dont il faudra mener l'apprentissage, les tests et la validation.

2. Elaboration de la structure du réseau

La structure du réseau dépend du type des échantillons et bien sûr de l'application proprement dite sachant que certaines architectures conviennent mieux à des problèmes donnés.

3. Apprentissage

L'apprentissage est l'avant dernière phase d'élaboration d'un réseau de neurones. Elle consiste tout d'abord à calculer les pondérations optimales des différentes liaisons en utilisant une partie des échantillons. La méthode la plus utilisée est la rétropropagation. Il faut bien prendre garde à ne pas surentraîner un réseau de neurone qui deviendra alors moins performant.

4. Validation

Une fois le réseau calculé, il faut procéder à des tests pour vérifier ses performances : c'est la validation. La méthode la plus simple étant de garder une partie des échantillons réservés à l'apprentissage, pour la validation. [29]

II.2.6. Les limitations d'un réseau de neurone

Les réseaux de neurones peuvent implémenter n'importe quelle fonction non linéaire jusqu'à un certain degré de fiabilité. Nous allons lister quelques avantages et inconvénients qui les caractérisent : [25]

Pour ce qui est des avantages d'utilisation des réseaux de neurones on peut citer :

- ❖ une tolérance à l'incertitude très élevée ;
- ❖ étant une multiple copie d'unités simples (les neurones), ils sont donc facilement extensibles ;
- ❖ une facilité d'application car ne nécessitant pas une compréhension approfondie ;
- ❖ un choix de types, d'architecture et de fonction d'activation de réseaux diverses ;
- ❖ ils possèdent une capacité de généraliser ;

Les inconvénients par contre sont :

- ❖ une facilité d'application donnant lieu à de nombreuses implémentations et des choix pas toujours justifiés ;
- ❖ malgré une solide base théorique, le choix du réseau appartient souvent à l'utilisateur car il n'existe pas de guide approprié et reconnu ;
- ❖ la surface d'erreur des réseaux complexes possède beaucoup de sommets (des maximums locaux) et des vallées (des minimums locaux). A cause de leur nature non linéaire ils peuvent être piégés dans un minimum local où les performances des réseaux sont nettement sous optimales. Pour éviter les minima locaux on doit envisager les solutions suivantes :
 1. modifier le pas d'apprentissage du réseau pour pousser le réseau hors des minima locaux. C'est un paramètre qui règle la taille de la surface d'erreur.
 2. entraîner un même réseau à partir de plusieurs choix initiaux de poids, pour ensuite ne garder que le meilleur d'entre eux. [25]

II.2.7. Conclusion

Les réseaux de neurones artificiels constituent un moyen de modélisation et de résolution de problèmes dans plusieurs domaines notamment la reconnaissance de formes (images ou signaux), le diagnostic, le contrôle, la traduction automatique, la compréhension du langage etc..... [29]

La puissance de ces systèmes réside dans leur capacité à manipuler de l'information implicite et n'ont pas besoin qu'elle soit exprimée explicitement pour la modéliser.

Donc, plutôt que de représenter les connaissances explicitement et de les manipuler à travers un moteur d'inférence, les programmes connexionnistes représentent les connaissances implicitement dans des patrons d'interactions à travers leurs composantes. Pour notre système, nous avons opté pour un réseau de neurone multicouche car il semble bien adapté au problème posé.

II.3.La logique floue

II.3.1. Introduction

L'une des caractéristiques d'un expert humain, en traitant une question, est son jugement avec des données imprécises, inachevées et incertaines. Etudier un tel problème par des méthodes classiques exige la connaissance des différents paramètres qui peuvent influencer sur celui-ci.

Cette étude est souvent entachée d'erreurs et d'imprécisions qui accompagnent toute modélisation de processus.

Il est possible de résoudre ce problème en utilisant des méthodes dites robustes et la logique floue en est une. Son idée est de capturer l'imprécision de la pensée humaine et de l'exprimer avec des outils mathématiques appropriés. La théorie des ensembles flous a donné naissance à un traitement original de l'incertitude, fondé sur l'idée d'ordre, et qui permet de formaliser le traitement de l'ignorance partielle et de l'inconsistance dans les systèmes d'informations.

Son intérêt réside dans sa capacité de traiter l'imprécis, l'incertain et le vague, elle est issue de la capacité de l'homme à décider et agir de façon pertinente malgré le flou des connaissances disponibles, elle est appliquée dans les systèmes de réglage ainsi que dans les systèmes experts; dans les systèmes de décision et pour la reconnaissance de formes etc.

Dans cette section nous allons présenter la base de la logique floue ainsi que, les étapes de construction d'un système flou.

II.3.2.Structure générale d'un système flou

Chaque système basé sur la logique floue est composé des blocs suivants :

- a) Base de connaissances (paramètres des fonctions d'appartenance), Elle est composée :
- D'une base de données qui fournit les définitions des concepts nécessaires utilisées pour définir les règles linguistiques et la manipulation des données floues. Ces concepts sont subjectivement définis et basés sur l'expérience et le jugement de l'expert. [34]
 - D'une base de règles qui est l'ensemble d'expressions linguistiques basé sur la connaissance d'un expert ou bien la logique de fonctionnement du processus. Cette connaissance et d'habitude formulée sous forme de règles " *SI-ALORS* ". Chacune de ces

règles est composée d'une ou plusieurs prémisses reliées entre elles généralement par l'opérateur flou "ET" ou "OU" ; et une conclusion précédée par l'opérateur "ALORS". L'ensemble des règles floues sont reliées entre elles généralement par l'opérateur flou "OU".

b) Fuzzification qui est la transformation des entrées précises en degrés d'appartenance

c) Defuzzification qui consiste en la transformation des résultats flous en sorties précises.

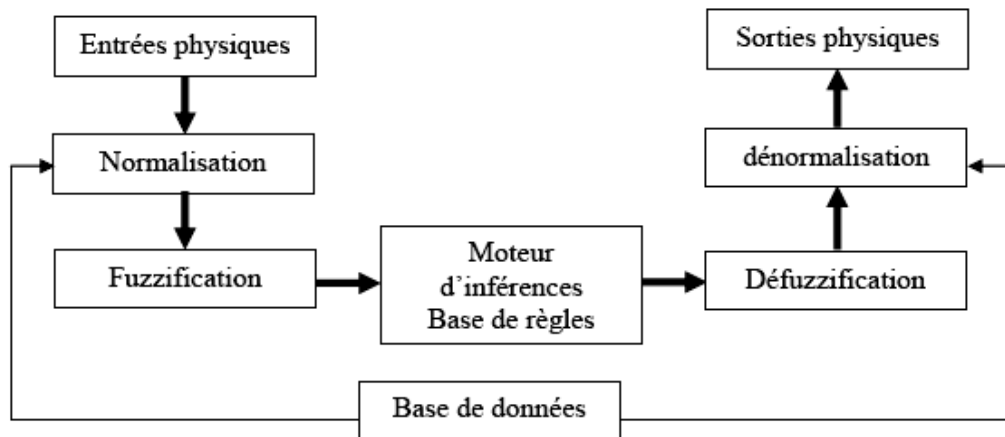


Figure II.6 : Schéma général d'un système flou.

II.3.3. Conception d'un système flou

II.3.3.1. Univers de discours

Un des premiers pas dans la conception d'un système flou est de définir l'ensemble de référence ou univers du discours pour chaque variable linguistique. Pour ces variables x on définit un ensemble i sur un univers de discours X par une fonction « degré d'appartenance »

Par exemple :

$$\mu : X \rightarrow [0, 100]$$

$$x \rightarrow \mu_i(x)$$

L'univers de discours est l'ensemble des valeurs réelles que peut prendre la variable floue x et $\mu_i(x)$ est le degré d'appartenance de l'élément x à l'ensemble floue i .

Le pas suivant, est le choix des variables linguistiques qui déterminent son état, puis des règles linguistiques qui établissent les relations d'inférence entre ces variables. En général, les règles sont proposées par un expert. On partitionne ensuite le domaine de chaque variable linguistique en un ensemble de fonctions d'appartenance, qui expriment les valeurs de façon approximative : petit, moyen, grand, énorme par exemple.

Les notions de valeurs « moyen » ou « faible » par exemple, sont relativement difficiles à spécifier de manière précise. On peut fixer des seuils et considérer que l'on attribue tel ou tel qualitatif en fonction de la valeur de la variable par rapport à ces seuils.

On peut définir le degré d'appartenance de la variable A par exemple à l'ensemble «faible» comme le degré de vérité de la proposition " A est faible "

En logique booléenne, le degré d'appartenance μ ne peut prendre que deux valeurs (0 ou 1). La valeur A peut être :

Faible : $\mu_{\text{faible}}(A) = 1$; $\mu_{\text{moyen}}(A) = 0$; $\mu_{\text{élevé}}(A) = 0$

Moyen : $\mu_{\text{faible}}(A) = 0$; $\mu_{\text{moyen}}(A) = 1$; $\mu_{\text{élevé}}(A) = 0$

Elevé : $\mu_{\text{faible}}(A) = 0$; $\mu_{\text{moyen}}(A) = 0$; $\mu_{\text{élevé}}(A) = 1$

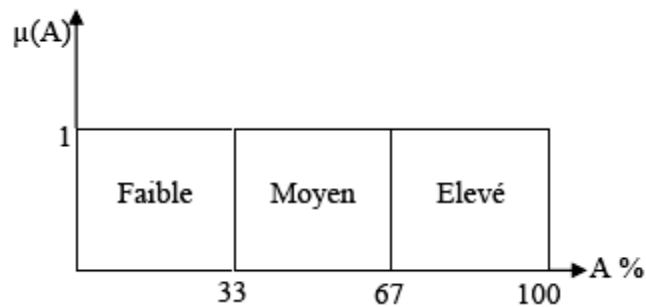


Figure II.7 : Degré d'appartenance aux ensembles considérés en logique booléenne.

Il ne peut pas prendre deux quantitatifs à la fois :

En logique floue, le degré d'appartenance devient une fonction qui peut prendre une valeur réelle comprise entre 0 et 1 inclus, c'est la fuzzification. [34]

II.3.3.2.Fuzzification

La fuzzification des variables d'entrée est une phase délicate du processus mis en oeuvre par la logique floue. Elle est souvent réalisée de manière itérative et requiert de l'expérience.

La fuzzification comprend

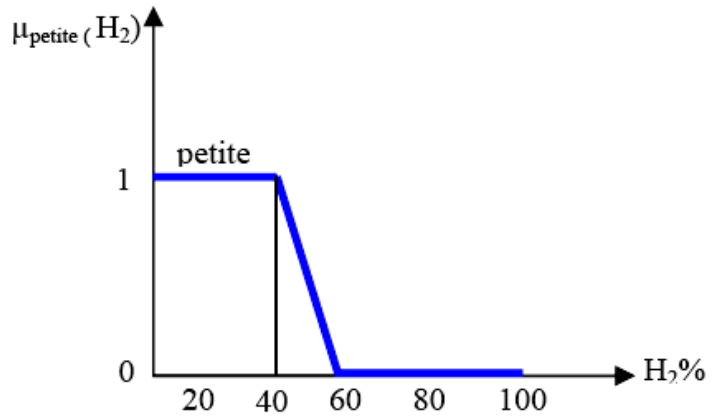
- Les mesures des variables d'entrées : En pratiques, les valeurs des variables d'entrées sont en général obtenus à l'aide de divers capteurs, organes de mesures, ou grâce à l'avis de différents opérateurs qui sont le plus souvent de type analogique.
- La conversion analogique digitale A/D (numérique/symbolique) ainsi que le traitement des grandeurs mesurées
- L'attribution de degrés d'appartenance à chaque valeur d'entrée est un passage des grandeurs physiques aux variables linguistiques (variables floues) qui peuvent être alors traitées par les inférences. Les variables linguistiques sont définies par leurs valeurs linguistiques.

Prenant l'exemple d'une valeur A = concentration en H₂

La concentration est petite au dessous de 40%.

A 50 %, elle n'est "qu'à moitié" petite.

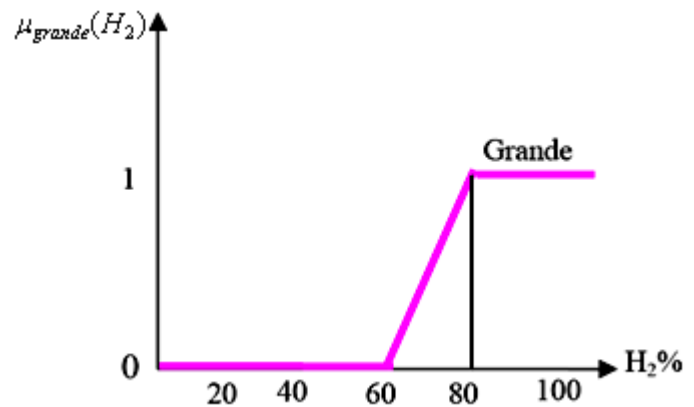
Il ne l'est plus du tout au delà de 60%.



La concentration est grande au dessus de 80%.

A 70 %, elle n'est "qu'à moitié" grande.

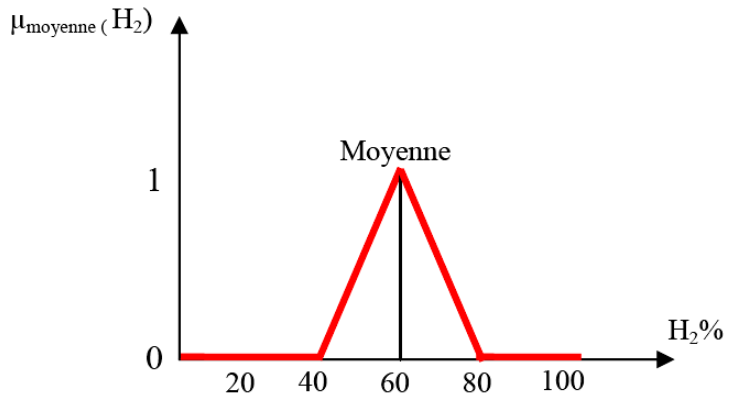
Il ne l'est plus du tout au dessous de 60 %.



La concentration est tout à fait moyenne à 60%

En dessous de 40%, elle n'est pas assez grande pour être moyenne.

Au delà de 80%, elle ne l'est plus non plus.



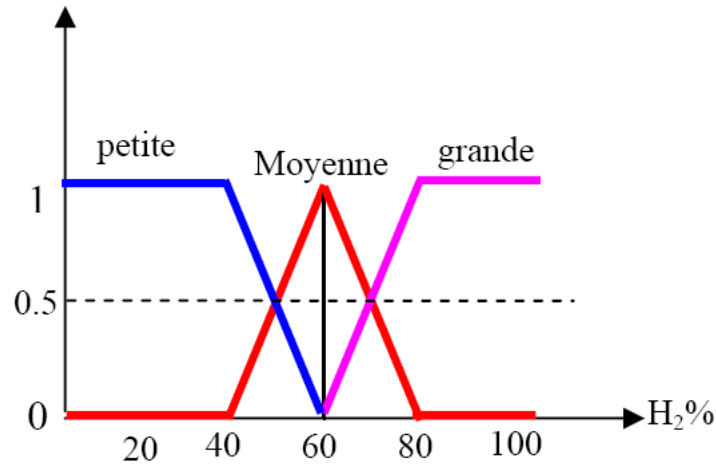


Figure II.8 : Exemple de fonction d'appartenance de H2%.

❖ Les fonctions d'appartenance

Le choix correct des fonctions d'appartenance d'un ensemble de termes joue un rôle essentiel dans la réussite d'une application. En ce qui concerne l'univers de discours, on peut ramener les grandeurs d'entrées dans l'intervalle $[-1 +1]$, on parle donc d'une **normalisation** de l'univers de discours. Ainsi, la normalisation d'un univers de discours nécessite une connaissance antérieure de l'espace d'entrées/sorties.

Il n'y a pas de règles précises pour la définition de fonctions d'appartenance. On peut introduire pour une variable x trois, cinq ou sept valeurs linguistiques. Ce nombre dépend de la résolution du réglage désiré. La désignation standard des ensembles flous est par exemple :

Symbole	Signification
NG	Négatif Grand
NM	Négatif Moyen
EZ	Environ Zéro
PM	Positif Moyen
PG	Positif Grand

Tableau II.1 : Exemple d'intervalle flou.

Un nombre de valeurs linguistiques supérieur à sept n'apporte en général aucune amélioration au comportement du système. Par contre, un tel choix compliquerait la

formulation des règles d'inférences. Les fonctions d'appartenance peuvent être symétriques, régulièrement distribuées ou avoir une distribution non uniforme. Il est important d'éviter le vide (lacunes) ou grand chevauchement de deux ensembles voisins, ceci provoque des zones de non intervention (zones mortes), ce qui conduit à une instabilité de la sortie. De même, il faut éviter un chevauchement trop important car il conduit à un aplatissement des caractéristiques de l'application.

Pour les formes de fonction d'appartenance, le plus souvent, on utilise des formes trapézoïdales ou triangulaires. Il s'agit des formes les plus simples, composées par morceaux de droites. Dans la plupart des cas, ces deux formes sont suffisantes pour délimiter des ensembles flous. [34]

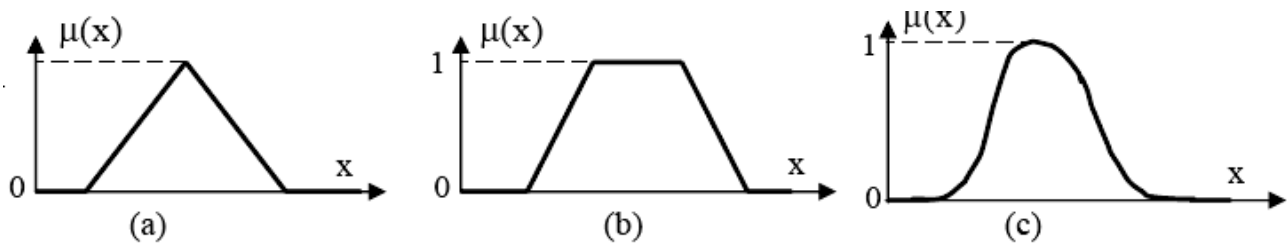


Figure II.9 : Les différentes formes de la fonction d'appartenance.

- a) Fonctions d'appartenance de forme triangulaire; b) Fonctions d'appartenance de forme trapézoïdale ; c) Fonctions d'appartenance de forme en cloche.

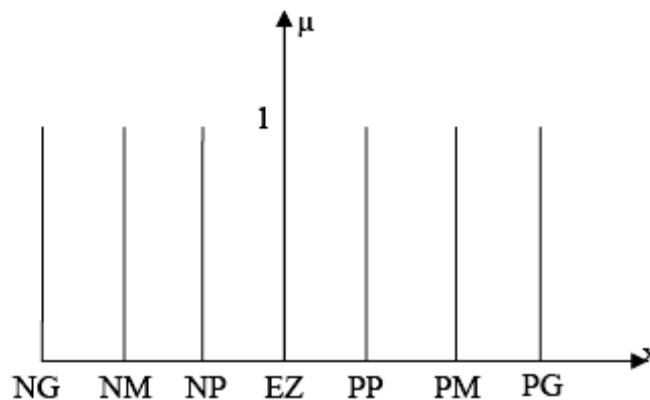


Figure II.10 : Sous ensembles flous de Sugeno.

Il ne faut pas oublier qu'il est indispensable de créer les fonctions d'appartenance pour les variables de sortie. D'habitude, on les crée de la même manière que les fonctions des entrées, c'est à dire en forme de triangle, de trapèze ou de cloche. Il est aussi possible de créer les sous-ensembles comme ceux de la de la figure II.10 qui désignent des valeurs précises. Ces sous ensembles sont connus dans la littérature comme les ensembles de

Sugeno. On peut utiliser ce type de fonction lorsqu'on a des sorties précises (qui prennent des valeurs binaires) et qu'on ne peut pas les mettre sous des ensembles flous. [34]

II.3.3.3. Inférences floues

Les inférences lient les grandeurs mesurées et les variables de sortie par des règles linguistiques. Ces règles sont combinées en utilisant les connexions **ET** et **OU**.

Les règles d'inférences peuvent être décrites de plusieurs façons. Linguistiquement, on écrit les règles de façon explicite comme dans l'exemple suivant :

Si (la vitesse est très grande) et (le feu rouge est proche) ALORS on freine fort.

Symboliquement, il s'agit au fait d'une description linguistique où l'on remplace la désignation des ensembles flous par des abréviations. Dans ce cas nous allons considérer par exemple un système flou avec n règles linguistiques

Si $x = A1$ et $y = B1$ alors $z = C1$

Si $x = A2$ et $y = B2$ alors $z = C2$

..

Si $x = An$ et $y = Bn$ alors $z = Cn$

Où x, y et z sont des variables linguistiques qui représentent les variables d'état du processus et les variables A_i , B_i et C_i ($i=1, n$) sont les sous ensembles flous définis dans les ensembles de références pour x, y, z respectivement.

On peut former une matrice d'inférence qui rassemble toutes les règles d'inférences sous forme de tableau (Tab.II.2). Dans le cas d'un tableau à deux dimensions, les deux entrées du tableau représentent les ensembles flous des variables d'entrées. L'intersection d'une colonne et d'une ligne donne l'ensemble flou de la variable de sortie définie par la règle. Il y a autant de cases que de règles

x	y				
	NG	NP	Z	PP	PG
NG	NG		PG		
NP		NG	PP		
Z			Z		
PP			PG	PG	
PG			NG		PG

Tableau II.2 : Exemple d'un tableau d'inférence.

NG Négatif Grand, **NP** Négatif Petit, **Z** Zéro, **PP** Positif Petit, **PG** Positif Grand

Dans un espace d'entrées/sorties, le cardinal d'un ensemble de termes détermine le nombre maximum de règles de l'application par logique floue qu'on peut construire. Dans le cas d'un système à deux entrées et une seule sortie, si les cardinaux de ses entrées sont respectivement n et m , alors le nombre de règles maximum est $n \times m$. [34]

II.3.3.3.1. Opérateurs de la logique floue

Puisque les fonctions caractéristiques des ensembles flous sont graduelles, l'extension de certains opérateurs définis sur les ensembles de la logique classique aux cas des ensembles flous pose certaines difficultés. Donc, il n'est pas possible d'appliquer directement la logique booléenne qui manipule des ensembles à fonctions caractéristiques binaires. Pour pallier à ce problème, sont introduites quelques définitions d'outils nécessaires au traitement des ensembles flous.

Il existe plusieurs lois de composition, mais la plus couramment utilisée est la suivante :

- Quand des conditions sont liées par une logique "**OU**", on considère le degré d'appartenance **maximum** parmi les conditions d'entrée.
- Quand des conditions sont liées par une logique "**ET**", on considère le degré d'appartenance **minimum** parmi les conditions d'entrée.

Alors, les opérateurs flous "**OU**", "**ET**" et "**ALORS**" appliqués aux deux ensembles flous sont définis à l'aide de leurs fonctions d'appartenance de la manière suivante :

- Opérateur ET :

Il est réalisé par plusieurs méthodes. On se limite à celle du produit des deux degrés d'appartenance $\mu(x_1)$ et $\mu(x_2)$ des deux ensembles flous en question soit :

$$\mu(x_1, x_2) = \mu(x_1) \cdot \mu(x_2)$$

- Opérateur OU :

La réalisation de cet opérateur se fait par exemple par la formation de la moyenne arithmétique des deux degrés d'appartenance $\mu_{1i}(x_1)$ et $\mu_{2j}(x_2)$ des deux ensembles flous i et j de x_1 et x_2 respectivement, c'est-à-dire :

$$\mu(x_1, x_2) = [\mu(x_1) + \mu(x_2)]/2$$

• Opérateur ALORS :

Il permet d'évaluer le degré de vérité des propositions floues, il est réalisé par plusieurs méthodes telles que le produit des deux fonctions d'appartenance $\mu_{1i}(x_1)$, et $\mu_{2j}(x_2)$ des deux ensembles flous i et j de x_1 et x_2 respectivement, c'est-à-dire :

$$\mu(x_1, x_2) = \mu(x_1) \cdot \mu(x_2)$$

Les opérations minimum et maximum présentent l'avantage de la simplicité lors du calcul, par contre, elles privilégient l'une des deux variables. Les opérations de produit et valeurs moyennes sont plus complexes à calculer mais elles produisent un résultat qui tient compte des valeurs des deux variables. [34]

II.3.3.3.2. méthodes d'inférences floues

Il existe plusieurs possibilités pour réaliser les opérateurs qui s'appliquent aux fonctions d'appartenance. On introduit la notion de méthode d'inférence, qui dépend des relations utilisées pour réaliser les différents opérateurs dans une inférence, permettant ainsi un traitement numérique de cette dernière. On cite les trois méthodes d'inférences les plus usuelles : Max-Min, Max-Produit et Somme Produit.

➤ Méthode de somme produit

La méthode d'inférence somme-produit réalise, au niveau de la condition, l'opérateur " OU " par la formation de la somme, plus précisément par la valeur moyenne, tandis que l'opérateur " ET " est réalisé par la formation du produit.

La conclusion de chaque règle, précédée par " ALORS ", liant le facteur d'appartenance de la condition avec la fonction d'appartenance de la variable de sortie par l'opérateur " ET ", est réalisée par la formation du produit. L'opérateur " OU " qui lie les différentes règles est réalisé de nouveau par la formation de la moyenne arithmétique. Pour mieux comprendre on va traiter l'exemple suivant :

Soit les deux entrées x et y : $x=0.44$, $y=-0.67$

- ❖ Si x est positif grand (PG) et y est environ zéro (EZ) alors z est EZ
- ❖ Si x est EZ ou y est négatif grand (NG) alors z est NG

Avec les valeurs d'appartenance :

$$\begin{aligned} \mu_{PG}(x=0.44) &= 0.67, \mu_{EZ}(y=-0.67) = 0.2 \\ \mu_{EZ}(x=0.44) &= 0.33, \mu_{NG}(y=-0.67) = 0.67 \end{aligned}$$

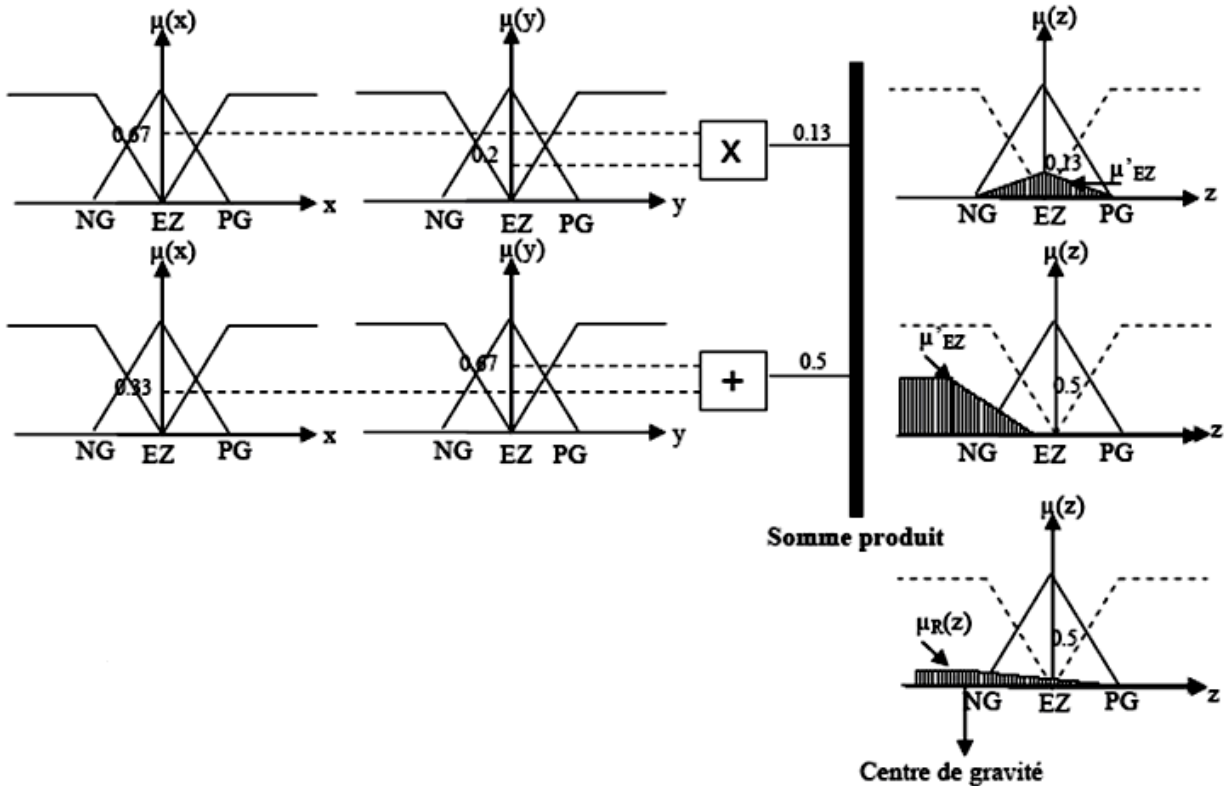


Figure II.11: Méthode d'inférence Somme-produit.

La première règle d'inférence se réalise par le produit de deux fonctions d'appartenance $\mu(x)$ et $\mu(y)$, ce produit nous donne 0.13. Tandis que la deuxième règle sera réalisée par la moyenne arithmétique des deux fonctions d'appartenance $\mu(x)$ et $\mu(y)$ ce qui nous donne 0.5.

La fonction d'appartenance résultante $\mu_R(z)$ s'obtient par la formation de la valeur moyenne de deux fonctions d'appartenance de sorties partielles. [34]

➤ Méthode Min Max

Cette méthode réalise, au niveau de la condition, l'opérateur **OU** par la formation du maximum et l'opérateur **ET** par la formation du minimum. La conclusion dans chaque règle, introduite par **ALORS**, lie le facteur d'appartenance de la condition avec la fonction d'appartenance de la variable de sortie (z) par l'opérateur **ET**, réalisé dans le cas présent par la formation du minimum. Enfin l'opérateur **OU** qui lie les différentes règles est réalisé par la formation du maximum.

A noter que la désignation de la méthode d'inférence (max-min dans le cas présent) se rapporte à la réalisation de **OU** liant les règles (max) et la réalisation de **ALORS** (min).

La figure II.12.b représente graphiquement le principe de la méthode d'inférence max-min. La condition (x est A_1 ET y est B_1) de la première règle implique pour $x=x^*$ et $y=y^*$ les facteurs d'appartenance $\mu_{A_1}(x^*)$ et $\mu_{B_1}(y^*)$ ce qui implique que la condition prend le facteur d'appartenance $\mu_{B_1}(y^*)$ (minimum des deux valeurs à cause de l'opérateur ET) (Fig. II.12.b), la fonction d'appartenance $\mu_{C_1}(z)$ est donc écrêtée à $\mu_{B_1}(y^*)$ (à cause de la formation du minimum lié à **Alors**).

De même pour la deuxième règle on trouve que la fonction d'appartenance $\mu_{C_2}(z)$ est écrêtée à $\mu_{A_2}(x^*)$ (Fig.II.12.b). Il faut encore déterminer la fonction d'appartenance résultante. Elle s'obtient par la formation du maximum des deux fonctions d'appartenances partielles, étant donné que les règles sont liées par l'opérateur **OU**. [33]

➤ **Méthode Max-prod**

Cette méthode, réalise en générale, au niveau de la condition, l'opérateur **OU** par la formation du maximum et l'opérateur **ET** par la formation du minimum. Par contre, la conclusion dans chaque règle introduite par **ALORS**, qui lie le facteur d'appartenance de la condition avec la fonction d'appartenance de la variable de sortie (z) par l'opérateur **ET**, est réalisé cette fois-ci par la formation du produit. L'opérateur **OU** qui lie les différentes règles est réalisé de nouveau par la formation du maximum.

Comme on le voit le **OU**, liant les règles est réalisé par la formation du maximum et le **ALORS** est réalisé par la formation du produit. D'où la désignation de cette méthode d'inférence par max-prod.

La représentation graphique du principe de la méthode d'inférence max-prod se trouve à la figure II.12.a.

Comme dans le cas du paragraphe précédent, la première condition prend le facteur d'appartenance $\mu_{B_1}(y^*)$. La fonction d'appartenance $\mu_{C_1}(z)$ pour la variable de sortie est cette fois-ci multipliée par ce facteur ($\mu_{B_1}(y^*)$), (à cause de la formation du produit lié à **ALORS**).

De la même manière pour la deuxième règle, la fonction $\mu_{C_2}(z)$ sera multipliée par $\mu_{A_2}(x^*)$ (formation du produit).

La fonction d'appartenance résultante s'obtient également par la formation du maximum des deux fonctions d'appartenance partielles, réalisent ainsi l'opérateur **OU** entre les règles. [33]

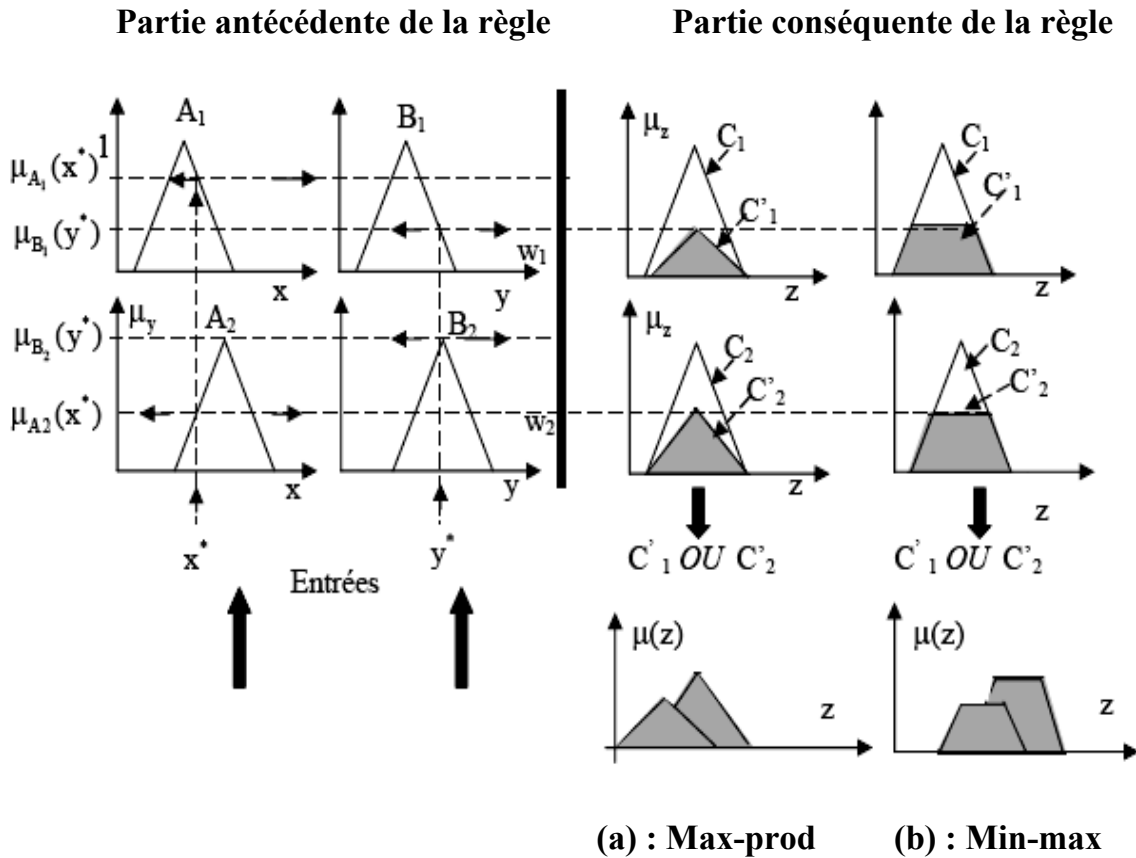


Figure II.12 : Méthode d'inférence Max Min et Max prod.

II.3.3.4. Defuzzification

Comme on l'a vu, les méthodes d'inférences fournissent une fonction d'appartenance résultante $\mu(z)$ pour la variable de sortie z . Il s'agit donc d'une information floue, il convient de la transformer en une grandeur précise.

La transformation d'une information floue en une information déterminée est la défuzzification (concrétisation). Pendant cette étape se fait la déduction de la grandeur de sortie numérique à partir de l'inférence floue, il s'agit de calculer à partir des degrés d'appartenance à tous les ensembles flous des variables d'entrées et des ensembles flous de la variable de sortie, une valeur numérique de la variable de sortie. Parmi les stratégies de défuzzification, nous citons la méthode de valeur maximale, la méthode de moyenne pondérée et la méthode de centre de gravité. [34]

1. Technique du maximum est la plus simple, elle consiste à ne considérer, pour chaque sortie, que la règle présentant le maximum de validité, c à d que la valeur de sortie est définie par la valeur maximale des variables floues de sortie.

Cette méthode est simple à appliquer mais, étant basée sur l'opérateur **maximum**, elle privilégie une seule règle parmi celles présentes, et ignore les règles secondaires qui peuvent néanmoins être importantes pour le fonctionnement et la stabilité du système. Elle est peu employée.

2. Technique de la moyenne pondérée est plus évoluée. Elle considère, comme valeur de sortie, la moyenne des valeurs préconisées par chaque règle, pondérées par leurs degrés respectifs de validité.

3. Technique du centre de gravité est plus performante : il s'agit de calculer le centre de gravité de la fonction d'appartenance de la variable de sortie.

Le calcul du centre de gravité permet bien d'obtenir une seule valeur pour la grandeur de sortie. Son calcul est cependant relativement complexe puisqu'il nécessite le calcul d'une intégrale, ou dans le cas simple de fonctions d'appartenance en raies (Fig.II.13), d'une somme pondérée.

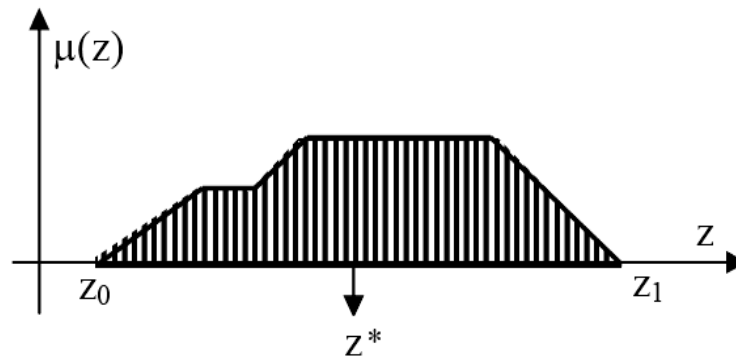


Figure II.13: Défuzzification par centre de gravité.

L'abscisse de centre de gravité Z^* peut être déterminé à l'aide de la relation générale:

$$z^* = \frac{\int_{z_0}^{z_1} z \mu(z) dz}{\int_{z_0}^{z_1} \mu(z) dz}$$

L'intégrale au dénominateur donne la surface, tandis que l'intégrale au numérateur correspond au moment de la surface.

Dans le cas d'un univers de discours discret, cette méthode donne :

$$z^* = \frac{\sum_{i=1}^n \mu(z_i) \cdot z_i}{\sum_{i=1}^n \mu(z_i)}$$

Avec :

n : nombre des niveaux de quantification.

Zi : la valeur de la sortie pour le niveau i.

II.3.3. Commande floue

La commande par logique floue est en pleine expansion. En effet, cette méthode permet d'obtenir une loi de réglage souvent très efficace sans devoir faire des modélisations approfondies. Par opposition à un régulateur standard ou à un régulateur à contre-réaction d'état, le régulateur par logique flou (RLF) ne traite pas une relation mathématique bien définie, mais utilise des inférences avec plusieurs règles, se basant sur des variables linguistiques. Par des inférences avec plusieurs règles, il est possible de tenir compte des expériences acquises par les opérateurs d'un processus technique. [20], [36],[22]

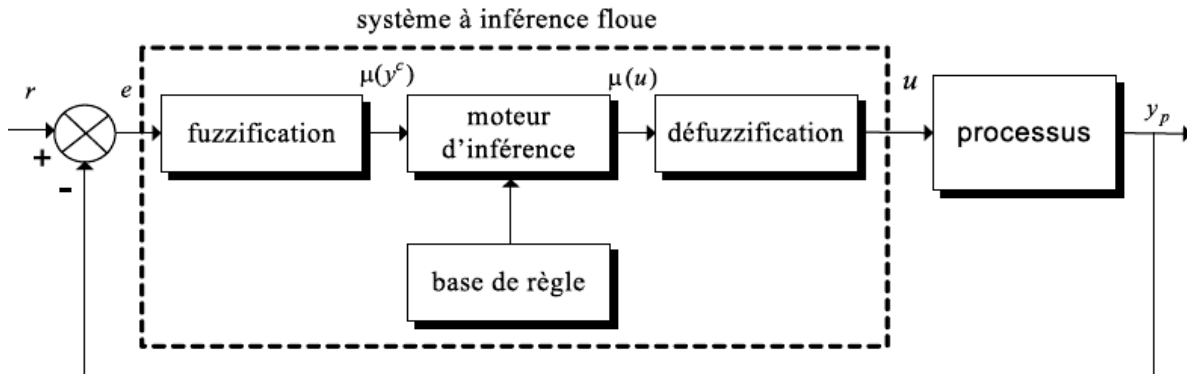


Figure II.14 : Configuration de base d'un régulateur par logique floue (RLF).

II.3.4. Avantages et inconvénients de la commande par logique floue

La commande par logique floue réunit un certain nombre d'avantages et de désavantages. [20]

Les avantages essentiels sont :

- La non-nécessité d'une modélisation (cependant, il peut être utile de disposer d'un modèle convenable) ;

- La possibilité d'implanter des connaissances (linguistiques) de l'opérateur de processus ;
- La maîtrise du procédé avec un comportement complexe (fortement non linéaire et difficile à modéliser) ;
- L'obtention fréquente de meilleures prestations dynamiques (régulateur non linéaire) ;

Les inconvénients de la commande par logique floue sont :

- Le manque de directives précises pour la conception d'un réglage (choix des grandeurs à mesurer, détermination de fuzzification, des inférences et de la défuzzification);
- L'approche non systématique (implantation des connaissances des opérateurs souvent difficile) ;
- L'impossibilité de la démonstration de la stabilité du circuit de réglage en toute généralité (en l'absence d'un modèle valable) ;
- La possibilité d'apparition de cycles limites à cause de fonctionnement non linéaire; la cohérence des inférences non garantie a priori (apparition possible de règles d'inférence contradictoires possible) ;

II.3.5. Conclusion

Dans ce chapitre, nous avons introduit les notions de base de la théorie des ensembles flous et de la logique floue. Alors nous pouvons dire que la logique floue ouvre des possibilités remarquables de codification des connaissances des experts .Cependant, les applications utilisant la logique floue ne sont pas fondamentalement plus performantes. Elle sont tout simplement plus facile à réaliser et à utiliser. L'utilisation faite par la logique floue d'expressions du langage courant permet aux systèmes flous de rester compréhensibles pour les personnes non expertes.

Sans grand risque de se tromper on peut affirmer que la logique floue va prendre de plus en plus de place dans plusieurs applications. Cette affirmation est motivée par le fait que ce principe ne nécessite pas de modèle du système à étudier et qu'il utilise les connaissances de l'opérateur (expert).

II.4.L'approche neuro-floue

II.4.1. Introduction

L'utilisation conjointe des réseaux de neurones et de la logique floue permet de tirer les avantages des deux méthodes ; les capacités d'apprentissage de la première et la lisibilité et la souplesse de la seconde. Diverses combinaisons de ces deux méthodes ont

été développées depuis 1988. Elles ont donné naissance aux systèmes neuro-flous, qui sont le plus souvent orientés vers la commande de systèmes complexes et les problèmes de classification. [22]

II.4.2. Motivations pour une approche hybride

Les dernières années ont vu se développer de nombreux travaux sur l'étude conjointe des réseaux de neurones et de la logique floue. Les raisons de cet engouement sont sans doute à rechercher dans les nombreuses similarités qui existent entre les deux approches. Tout d'abord les deux techniques présentent des propriétés d'approximateurs universels de fonction. En effet il a été montré qu'il est possible d'approcher n'importe quelle fonction continue à l'aide d'un système flou ou d'un réseau neuronal classique à couches. On peut aussi trouver des similitudes au niveau de la structure. Il existe en premier lieu une analogie entre les données des règles (IF-THEN) de la logique floue et les couples (entrée-sortie) des réseaux de neurones. De même on peut trouver un point commun entre les fonctions réalisées par les neurones formels et les fonctions d'appartenance caractérisant les différentes variables d'un système flou. Enfin on peut établir une corrélation entre les opérations de multiplication et d'addition induites par la structure neuronale et celles de maximisation et de minimisation correspondant aux conjonctions et aux disjonctions des règles. [24], [23]

Cependant la logique floue et les réseaux de neurones possèdent tous deux des points forts et des points faibles différents :

	<i>LA LOGIQUE FLOUE</i>	<i>LES RESEAUX DE NEURONE</i>
<i>AVANTAGES</i>	<ul style="list-style-type: none"> • Intégration de la connaissance à priori • Facilité de construction • Possibilité d'interpréter la connaissance 	<ul style="list-style-type: none"> • Capacités d'apprentissage • Parallélisme massif • Résistance aux données bruitées
<i>INCONVENIENTS</i>	<ul style="list-style-type: none"> • Construction manuelle des règles • Difficultés d'optimisation des différents paramètres 	<ul style="list-style-type: none"> • Aucune interprétation possible des résultats de l'apprentissage • Difficulté d'estimer les paramètres l'apprentissage

Tableau II.3 : Comparaison entre la logique floue et les réseaux de neurones.

De nombreux auteurs ont cherché à tirer partie de cette complémentarité en présentant des modèles hybrides empruntant des caractéristiques aux deux approches.

II.4.3. Systèmes hybrides neuro-flous

Les réseaux de neurones artificiels et la logique floue peuvent être complémentaires sur plusieurs points. La logique floue permet une spécification rapide des tâches à accomplir à partir de la connaissance symbolique disponible. Le réglage précis du système obtenu et l'optimisation de ses différents paramètres reste néanmoins beaucoup plus difficile dans de nombreux cas. Les modèles les plus courants de RNA, au contraire, n'autorisent pas l'incorporation de connaissance a priori mais permettent de régler par apprentissage le comportement précis du système.

De nombreux auteurs ont donc tout naturellement chercher à combiner ces deux paradigmes depuis le début des années 90 et ceci de plusieurs manières.

Nous portons ici notre attention sur les approches permettant de représenter sous forme de RNA les règles d'un système d'inférence flou. Cette technique consiste à utiliser des fonctions d'activation particulières pour les unités du réseau, ainsi qu'une organisation spécifique de ces dernières et ce afin de reproduire les différents éléments constitutifs d'un système d'inférence flou.

La structure du réseau est ainsi choisie en fonction de la forme de la fonction que l'on cherche à approcher. Les dépendances entre les variables d'entrée et de sortie sont en effet spécifiées par le choix des règles floues. Les différents paramètres de ces règles (c à d forme et position des sous-ensembles flous, sortie et poids des règles), peuvent ensuite être modifiés par un algorithme d'apprentissage supervisé. Nous pouvons noter que cela ne restreint en rien l'ensemble des fonctions approchables puisqu'il a été démontré que la logique floue présente comme les RNA des propriétés d'approximation universelle de fonction.

Si l'on adopte le point de vue des systèmes flous, cette approche permet de régler de manière précise, par apprentissage, le comportement du système réalisé. Si l'on se place au contraire du côté des RNA, la connaissance sous forme de règles floues permet de choisir l'architecture du réseau en fonction de la tâche à accomplir. [23], [24]

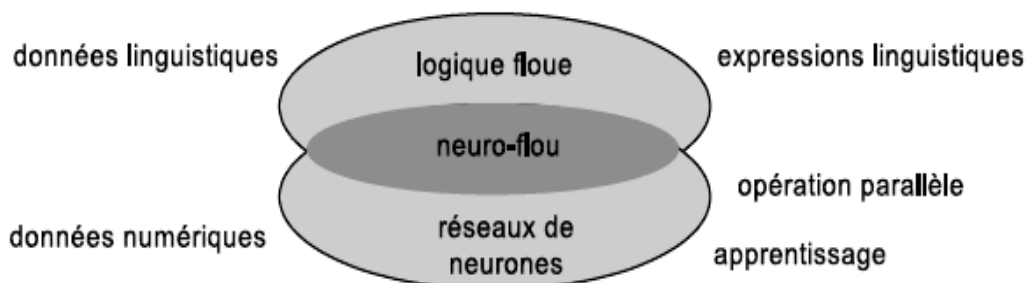


Figure II.15: Principe du système neuro-flou.

La figure II.15 résume le principe du système neuro-flou qui représente l'intersection entre la logique floue et les réseaux de neurones.

Un système neuro-flou est donc défini comme étant un réseau neuronal multicouche avec des paramètres flous, ou comme un système flou mis en application sous une forme distribuée parallèle. Il a été notamment répertorié trois familles différentes : [22]

1. Le modèle *FALCON* et le modèle *GARIC*:

Ces deux structures à 5 couches utilisent la fuzzification en entrée et la défuzzification en sortie, correspondant à l'interprétation juste de la technique de Mamdani. Ce sont des structures très précises mais très lentes à l'exécution. Ces modèles sont généralement utilisés pour la commande.

2. Le modèle *NEFCLASS* :

Modèle utilisé principalement en classification, il est constitué de 3 couches; une couche d'entrée avec des fonctions d'appartenance, une couche cachée représentée par des règles et une couche de sortie définissant les classes. Modèle facile à mettre en application car il évite l'étape de défuzzification tout en étant précis.

3. Le modèle *ANFIS* :

Modèle le plus utilisé en pratique. C'est une structure à 6 couches qui affine les règles floues déjà établies par des experts humains et réajuste le chevauchement entre les différents sous-ensembles flous. Des applications dans le traitement du signal et le filtrage adaptatif ont été réalisés avec cette architecture.

Plusieurs ouvrages et articles montrent un bon rendement du modèle ANFIS lorsqu'il est utilisé dans la commande de moteurs.

II.4.4. Système d'inférence flou basé sur les réseaux de neurones adaptatifs

Les réseaux de neurones (*RN*) multicouches constituent des approximateurs universels. L'atout principal de ces réseaux réside dans leur capacité d'apprentissage. Par contre, leur structure et leurs paramètres n'ont pas toujours des justifications physiques. De plus, la connaissance humaine ne peut pas être exploitée pour les construire. Les systèmes d'inférence flous sont également des approximateurs universels. Ces systèmes possèdent deux points forts par rapport aux *RN*. D'une part, ils sont généralement construits à partir de la connaissance humaine, d'autre part, ils ont une capacité descriptive élevée due à l'utilisation de variables linguistiques. Il est donc apparu naturel de construire des systèmes hybrides qui combinent les concepts des systèmes d'inférence flous et des *RN*. Ainsi l'approche *Neuro-floue* est apparue. [37]

Plusieurs auteurs ont utilisé cette approche. Nous présentons ici un type spécial de réseaux neuro-flous. Il s'agit de l'approche ANFIS (Système d'inférence flou organisé en réseau adaptatif) proposée par Jang. La modélisation découverte par Takagi & Sugeno a trouvé plusieurs applications dans les domaines de contrôle, de prédiction et d'inférence. Cependant, des aspects de base qui renforcent l'appréhension manquent, à savoir :

- La non-existence de méthodes standard pour transformer l'expérience humaine ou la connaissance humaine en une base de règles et une base de données d'un système d'inférence flou (S.I.F).
- Le besoin de méthodes effectives pour ajuster les fonctions d'appartenance en minimisant l'erreur mesurée en sortie ou en maximisant l'indice de performance.

Dans cette perspective, J.S.R. JANG proposa une nouvelle architecture appelée système d'inférence flou basée sur les réseaux de neurones adaptatifs *S.I.F.R.N.A* (A.N.F.I.S : Adaptative Network Fuzzy Inference System) qui peuvent servir comme une base pour construire l'ensemble des règles "Si-Alors" avec des fonctions d'appartenance appropriées en partant d'un ensemble de paires de données (entrées/sorties) du système.

II.4.4.1. Vue d'ensemble d'ANFIS

ANFIS met en application un SIF du type Takagi Sugeno et a une architecture composée de cinq couches comme représentée sur la figure II.16:

Pour simplifier la compréhension et sans perte de généralité, nous considérons un système à deux entrées x_1 et x_2 et une sortie y . Considérons aussi un modèle flou de type TSK de ce système, composé des deux règles suivantes:

$$\begin{aligned} \text{Si } x_1 \text{ est } A_1 \text{ et } x_2 \text{ est } B_1 \text{ alors } y_1 &= f_1(x_1, x_2) = a_1 x_1 + b_1 x_2 + c_1 \\ \text{Si } x_1 \text{ est } A_2 \text{ et } x_2 \text{ est } B_2 \text{ alors } y_2 &= f_2(x_1, x_2) = a_2 x_1 + b_2 x_2 + c_2 \end{aligned} \quad (\text{II.3})$$

Jang a proposé de représenter cette base de règles par le réseau adaptatif de la figure II.16.

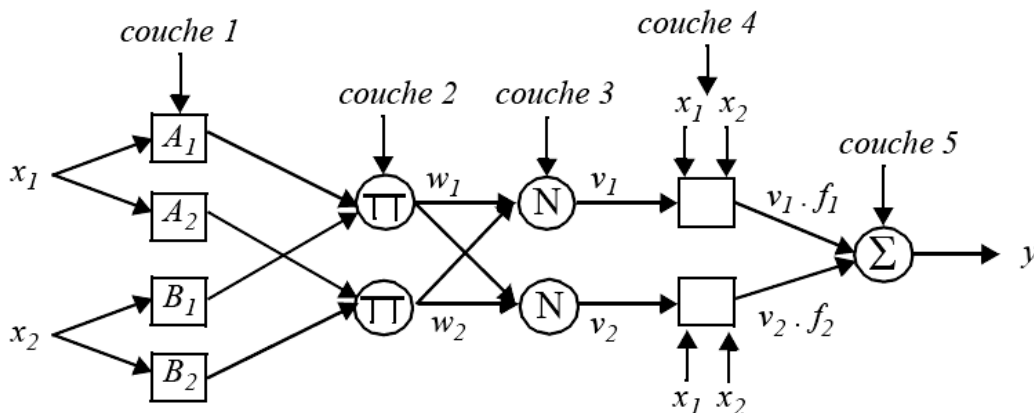


Figure II.16 : Architecture équivalente d'ANFIS pour deux règles.

Le réseau adaptatif ANFIS est un réseau multi-couches dont les connexions ne sont pas pondérées, ou ont toutes un poids de 1. Les noeuds sont de deux types différents selon leur fonctionnalité: les noeuds carrés (adaptatifs) contiennent des paramètres, et les noeuds circulaires (fixes) n'ont pas de paramètres. Toutefois, chaque noeud (carré ou circulaire) applique une fonction sur ses signaux d'entrées. La sortie O_i^k du noeud i de la couche k (appelée noeud (i, k)) dépend des signaux provenant de la couche $k-1$ et des paramètres du noeud (i, k) , c'est-à-dire,

$$O_i^k = f(O_1^{k-1} \dots O_{n_{k-1}}^{k-1}, a, b, c \dots) \quad (\text{II.4})$$

Où n_{k-1} est le nombre de noeuds dans la couche $k-1$, et $a, b, c \dots$ sont les paramètres du noeud (i, k) . Pour un noeud circulaire ces paramètres n'existent pas.

Dans le réseau de la figure (II.16), les noeuds d'une même couche ont des fonctions issues d'une même famille que nous explicitons ci-dessous.

Couche 1 :

La première couche représente les fonctions d'appartenances floues. Chaque noeud de cette couche est un noeud carré avec une fonction:

$$O_i^1 = \mu_{A_i}(x) \quad (\text{II.5})$$

Où x est l'entrée du noeud i , et A_i le terme linguistique associé à sa fonction. En d'autres termes, O_i^1 est le degré d'appartenance de x à A_i . Les paramètres d'un noeud de cette couche sont ceux de la fonction d'appartenance correspondante.

Couche 2 :

Chaque noeud i de cette couche est un noeud circulaire appelé Π qui engendre en sortie le produit de ses entrées. Ce produit représente le degré d'activation d'une règle:

$$w_i = \mu_{A_i}(x_1) \cdot \mu_{B_i}(x_2) \quad , i = 1, 2 \quad (\text{II.6})$$

Couche 3 :

Chaque noeud de cette couche est un noeud circulaire appelé N .

La sortie du noeud i est le degré d'activation normalisé de la règle i :

$$v_i = \frac{w_i}{w_1 + w_2} \quad (\text{II.7})$$

- La seconde et la troisième couche contiennent les noeuds qui forment la partie antécédente dans chaque règle

Couche 4 :

La quatrième couche calcule les coefficients de l'équation du premier ordre d'une règle de type Takagi-Sugeno et cela pour chaque règle floue.

Chaque noeud de cette couche est un noeud carré avec une fonction réalisant le calcul:

$$O_i^4 = v_i \cdot f_i = v_i (a_i x_1 + b_i x_2 + c_i) \quad , i = 1..2 \quad (\text{II.8})$$

Où v_i est la sortie de la couche 3, et $\{a_i, b_i, c_i\}$ est l'ensemble des paramètres de sortie de la règle i .

Couche 5 :

La cinquième couche qui représente la couche de sortie, calcule la sortie globale pesée (pondérée) du système. Le seul noeud de cette couche est un noeud circulaire qui effectue la somme des signaux provenant de la couche 4, c'est-à-dire,

$$O_1^5 = y = \sum_i v_i \cdot f_i \quad (\text{II.9})$$

La généralisation du réseau à un système à r entrées ne pose aucun problème particulier. Le nombre de noeuds de la couche 1 est toujours égal au nombre total de termes linguistiques définis.

L'apprentissage à partir d'un ensemble de données concerne l'identification des paramètres des prémisses et des conséquences, la structure du réseau étant fixée. L'algorithme d'apprentissage commence par construire un réseau initial, ensuite on applique une méthode d'apprentissage par rétro-propagation de l'erreur. Jang a proposé d'utiliser une règle hybride d'apprentissage qui combine un algorithme de descente de gradient avec une estimation par moindres carrées.

En ce qui concerne les paramètres des prémisses (couche 1), Jang a suggéré de n'appliquer une technique d'identification à partir des données que si l'ensemble de données est suffisamment grand. Dans ce cas, l'identification permet d'affiner les fonctions d'appartenance proposées par l'expert humain. Dans le cas contraire, il vaut mieux conserver les fonctions d'appartenance proposées par l'expert car elles reflètent les connaissances de l'expert.

Jang a montré l'équivalence entre les réseaux ANFIS et des réseaux neuronaux classiques avec des connexions pondérées. Il a également montré que sous certaines conditions, le comportement fonctionnel d'un ANFIS est équivalent à celui d'un RBFN (Radial Basis Function Network). Cette équivalence nécessite de nombreuses conditions restrictives sur les deux types de systèmes (flou ou RBF), entraînant donc leur perte d'identité propre. [37]

II.5.Conclusion

Dans cette section, nous avons présenté la méthode hybride neuro-floue. Nous avons vu que cette méthode :

- Combine les traitement parallèle et les capacités d'apprentissage des réseaux de neurones avec les raisonnements anthropomorphiques et les capacités d'explication des systèmes flous : Les *RNA* deviennent plus transparents, les systèmes flou acquièrent la capacité d'apprendre.
- *RNA* fonctionnellement équivalent à un modèle d'inférence floue ; on peut l'entraîner à développer des règles floues *SI-ALORS* et à trouver les fonctions d'appartenance de variables d'entrées/sorties en partant d'un ensemble de données représentatives.
- On peut aussi y inclure les connaissances d'un expert.
- Structure similaire à RNA multicouches : 1 couche d'entrée, 1 couche de sortie et 3 couches cachées pour les fonctions d'appartenance et les règles.

Le chapitre suivant sera consacré à l'application des réseaux de neurones pour la poursuite de point de puissance maximale.

Chapitre III

Conception d'un contrôleur MPPT neuronal

III.1.Introduction

Les caractéristiques de sortie d'un générateur photovoltaïque sont non linéaires et directement dépendant des conditions de travail tel que : le niveau d'irradiation solaire et la température ambiante. Aussi, pour une condition de travail donnée, il existe un seul point de fonctionnement au niveau duquel la puissance fournie par le générateur photovoltaïque est maximale appelé point de puissance maximale (PPM). Ce dernier se trouve à son tour fortement dépendant principalement du niveau d'irradiation solaire et de la température ambiante.

Les techniques MPPT (maximum power point tracking) sont utilisées dans les systèmes photovoltaïque pour maximiser la puissance délivrée par le panneau photovoltaïques en poursuivant continuellement le point de puissance maximale, ce qui n'est pas évident à atteindre ; En effet, ce problème de poursuite fait jusqu'à nos jours l'objet de plusieurs recherches. Aussi, plusieurs méthodes ont être élaborées et utilisées.

Dans ce chapitre, nous proposons une technique de poursuite de point de puissance maximale basée sur le réseau de neurone multicouche. Notre contribution consiste en l'identification de notre contrôleur et la commande de l'hacheur MPPT. Pour cela, et comme première étape, nous donnons une description détaillée de notre système neuronal développé, ainsi que les résultats de simulation obtenus.

III.2. Description et architecture du contrôleur MPPT neuronal proposé

La figure III.1 représente le schéma synoptique d'un système photovoltaïque doté d'une commande MPPT à base des réseaux de neurones artificiels.

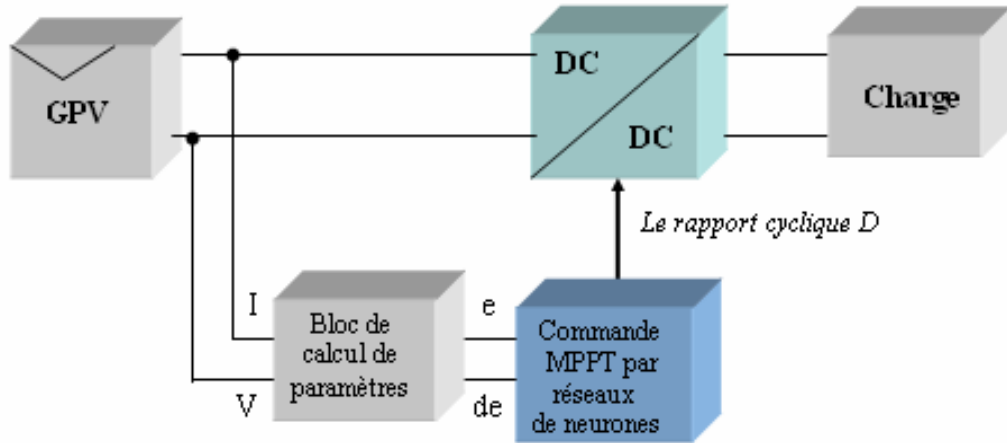


Figure III.1 : Schéma synoptique d'un système photovoltaïque avec une commande MPPT par réseaux de neurones.

III.2.1.Choix de la structure du réseau

Le réseau de neurones proposé est un réseau multicouche dont l'architecture adoptée est illustrée à la figure (III.2).

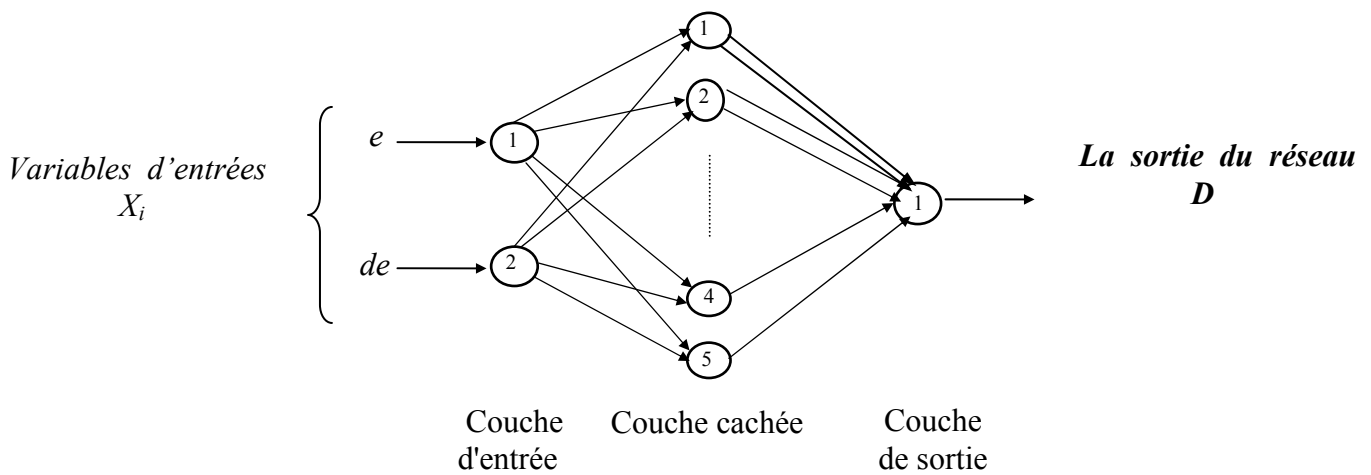


Figure III.2 : L'architecture du réseau neuronal proposé.

La figure III.2 montre bien que notre réseau est constitué de trois couches à savoir :

- Une couche d'entrée composée de deux neurones, dont le rôle est de transmettre les valeurs des entrées qui correspondent aux variables (e , de) vers la couche suivante appelée *couche cachée*.
- Une couche cachée dotée de cinq neurones avec des fonctions d'activations choisies de type sigmoïde tangentielle (Fig III.a).
- Une couche de sortie avec un seul neurone représentant la commande "D" dont la fonction d'activation employée est de type identité ou 'linéaire' (Fig III.b).

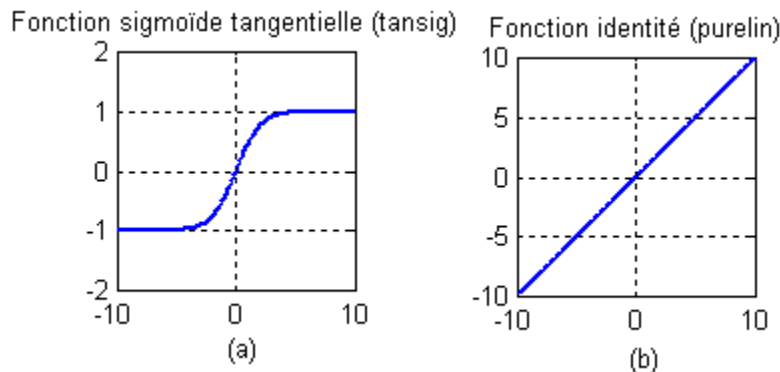


Figure III.3 : Simulation des fonctions de transfert du réseau de neurone.

Le nombre de neurones de la couche cachée a été optimisé empiriquement durant la phase d'apprentissage. En effet, les essais ont montré que la structure la plus stable est celle composée de cinq neurones.

Il est également à noter que le choix de la fonction d'activation de la couche cachée pour laquelle nous avons opté n'a pas été adopté de manière arbitraire, mais a été choisie après plusieurs tests qui ont montré que la fonction de type sigmoïde tangentielle converge plus rapidement par rapport à la fonction sigmoïde exponentielle durant la phase d'apprentissage.

En plus, deux autres paramètres doivent être également optimisés à savoir : le nombre d'époques qui représente la durée d'apprentissage, ainsi que la performance de l'apprentissage qui est l'écart minimum entre les valeurs présentées au réseau ou 'Patrons entrées-sorties' et les valeurs obtenues. Cet entraînement a été fait en utilisant l'algorithme de rétro-propagation qui sera détaillé ultérieurement.

Par ailleurs, la deuxième étape, après avoir proposé l'architecture optimale obtenue qui consiste en le choix du nombre de couches cachées, nombre de neurones par couches et les fonctions d'activations de notre réseau de neurones, consiste à procéder à son entraînement hors ligne en utilisant deux algorithmes différents. Pour ce faire, la structure générale du réseau de neurone sous simulink est illustrée à la figure III.4.

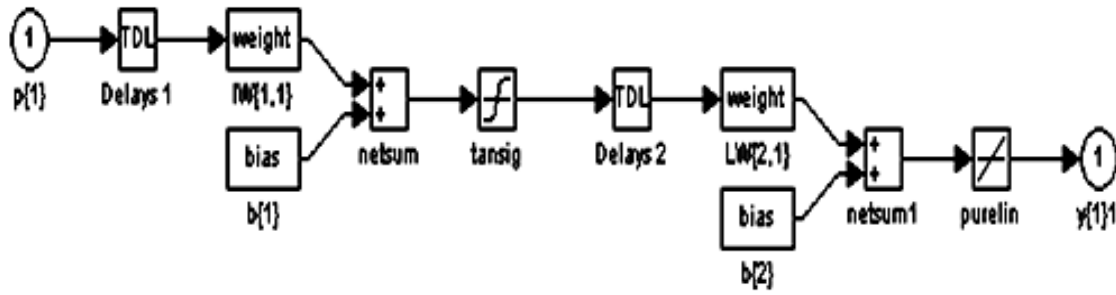


Figure III.4 : Structure générale du réseau de neurone sous Simulink.

La figure III.5 représente la structure interne de la première et deuxième couche de notre RNA sous Simulink.

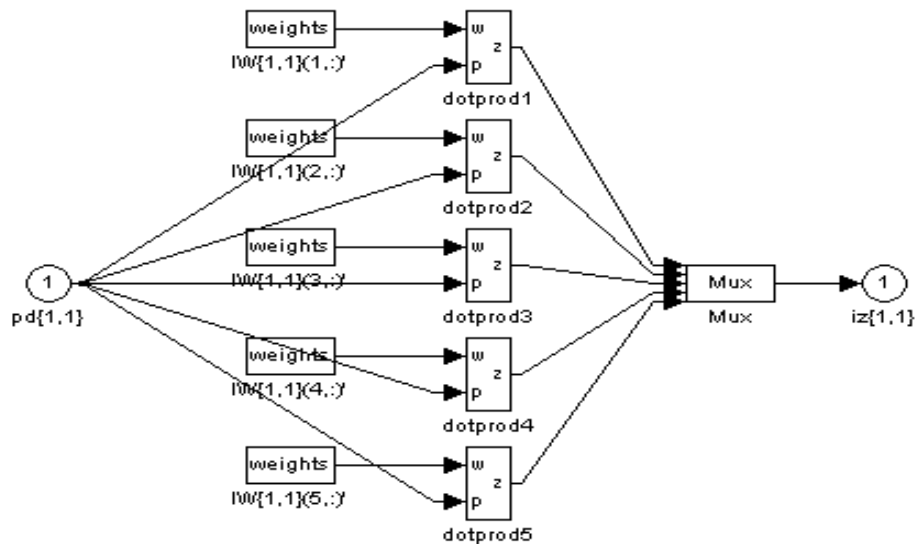


Figure III.5 : Structure interne de la première couche avec couche cachée du réseau de neurone sous Simulink.

Comme troisième étape, nous avons décidé de voir l'impact du choix de l'algorithme d'entraînement sur les performances de l'entraînement du point de vue rapidité de convergence (minimisation de l'erreur quadratique), et cela, en utilisant deux algorithmes différents à savoir : *rétro-propagation* et *Levenberg Marquardt*.

III.2.2. Présentation des deux algorithmes utilisés pour l'entraînement du réseau de neurones proposé

Comme il a été précisé précédemment, une fois l'architecture choisie, Le RNA doit subir une phase d'apprentissage. Cette dernière consiste à calculer les coefficients synaptiques de telle manière que les sorties du réseau de neurones soient, pour les exemples utilisés lors de l'apprentissage, aussi proches que possibles des sorties "*désirées*". Celles-ci correspondent aux valeurs des données (data) de la fonction que l'on veut approcher ou à la sortie du processus que l'on veut modéliser.

L'apprentissage de notre réseau a été effectué avec deux algorithmes différents :

- ❖ Algorithme de rétro propagation de gradient.
- ❖ Algorithme de Levenberg Marquardt.

III.2.2.1. Apprentissage par rétro propagation de gradient

Le développement du réseau se base sur l'apprentissage supervisé à l'aide de l'algorithme de rétro propagation. Notons que cet algorithme a été utilisé avec beaucoup de succès dans plusieurs applications complexes.

Pendant cette phase, l'algorithme de rétro propagation du gradient d'erreur cherche les poids W_i qui minimise l'erreur quadratique moyenne (*EQM*) entre la valeur imposée et la valeur estimée par le modèle régit par l'équation suivante :

$$EQM = \frac{1}{2} \sum (D_{désirée} - D_{estimée})^2$$

En d'autres termes, les poids W_i sont continuellement modifiés tant que l'erreur RQM est supérieure à une certaine valeur seuil correspondant à la précision acceptable de sortie.

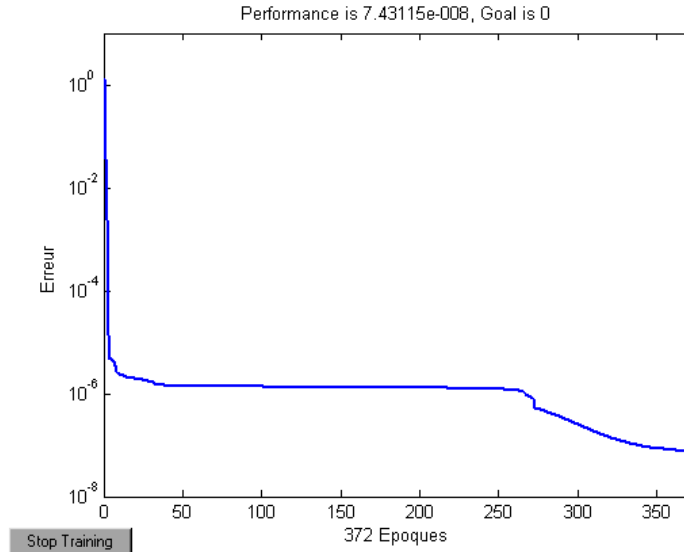


Figure III.6 : Evaluation de l'erreur quadratique en fonction du nombre d'itérations d'apprentissage (en utilisant la méthode de rétro propagation de gradient).

L'adaptation des poids des connexions, représentés par les matrices [W1] et [W2], est effectuée en fonction des stimuli présents à l'entrée du réseau. Le but est de trouver les poids des connexions appropriés aux données qui assurent la convergence vers la sortie désirée.

Une fois l'apprentissage effectué fini, les poids ne sont plus modifiés, ce qui correspond à la structure optimale de RNA.

III.2.2.2. Apprentissage par Levenberg Marquardt

D'un autre côté, et en vue d'une comparaison avec les performances obtenues avec la méthode précédente de rétro propagation, l'entraînement du système sera maintenant effectué par la méthode de Levenberg Marquardt.

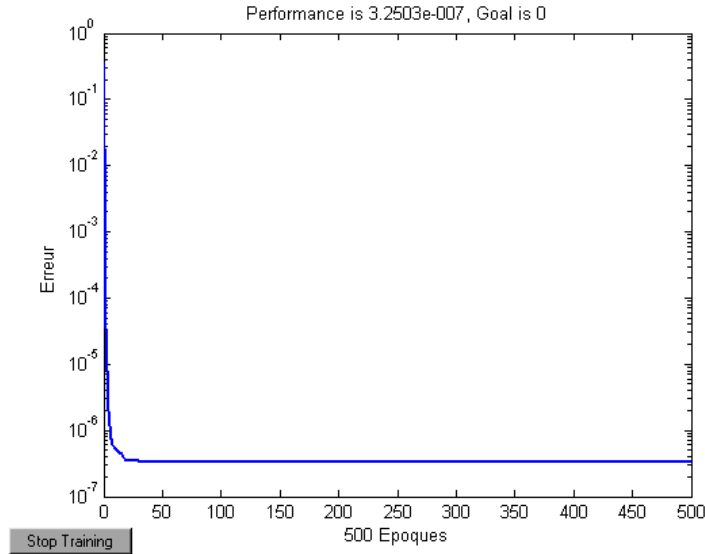


Figure III.7 : L'erreur d'apprentissage pendant l'entraînement du réseau de neurone (en utilisant la méthode de Levenberg Marquardt).

Les résultats obtenus montrent clairement que l'erreur visée a été atteinte par cette méthode avec une grande vitesse de convergence et à une grande capacité d'approximations, la raison qui nous a conduit à adopter cet algorithme pour la suite du travail.

III.2.3. Validation "Capacité de généralisation"

L'évaluation de la capacité de généralisation du réseau est réalisée sur la base de validation, en utilisant des couples entrée-sortie n'appartenant pas à la base d'apprentissage ; c'est-à-dire après avoir effectué l'apprentissage et avoir trouvé les poids du réseau nécessaires au calcul des propriétés, il faut estimer l'erreur sur la base de test (données qui n'ont pas été utilisés dans la base d'apprentissage pour évaluer la capacité de généralisation du réseau).

Le système photovoltaïque utilisé est composé des blocs suivants : Un générateur photovoltaïque, un hacheur, une batterie et un contrôleur MPPT neuronal (Fig. III.8).

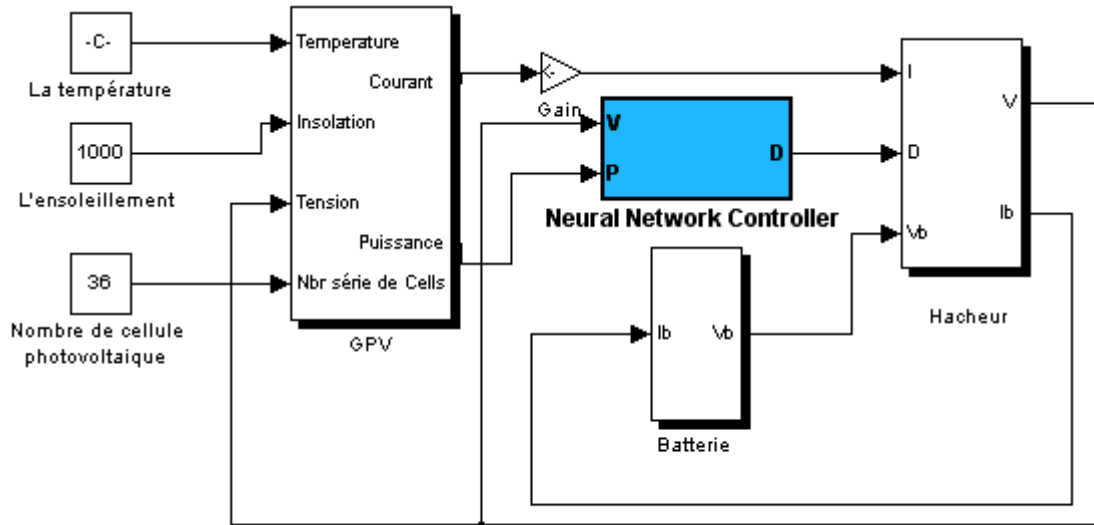


Figure III.8 : Schéma synoptique d'un système photovoltaïque doté d'un contrôleur neuronal MPPT sous Simulink.

Le système photovoltaïque utilisé est composé des blocs suivants : Un générateur photovoltaïque, un hacheur, une batterie et un contrôleur MPPT neuronal (Fig. III.8).

III.3. Résultats de simulation de l'application de l'approche neuronale à la poursuite du point de puissance maximale

Une fois la capacité de généralisation du réseau prouvée nous pouvons maintenant insérer ce dernier dans le schéma de contrôle comme le montre la figure (III.8). Une simulation numérique a été effectuée pour les différentes conditions à savoir :

- 1) Simulation du fonctionnement du système sous des conditions constantes (standards) ;
- 2) Simulations sous des conditions variables d'ensoleillement où deux cas sont envisagés lesquels consistent en la diminution et l'augmentation respectives de l'ensoleillement (lente et rapide) ;
- 3) Simulation sous des conditions variables de la température où également deux cas sont considérés : une augmentation lente de la température ou rapide et une diminution lente de la température ou rapide ;
- 4) Simulation sous des conditions aléatoires de l'ensoleillement et de la température ;

III.3.1. Simulation du fonctionnement sous des conditions standard (constantes)

Dans le but de visualiser la position du point de puissance maximale par la méthode de commande par réseaux de neurones artificiels, une simulation est effectuée pour une température et un ensoleillement constants, c'est-à-dire une température $=25^{\circ}\text{C}$ et un ensoleillement $=1000\text{W}/\text{m}^2$ (Fig. III.9.).

La figure III.9 montre la variation de la puissance du module et de la puissance de la batterie ainsi que la commande D (le rapport cyclique d) en fonction du temps.

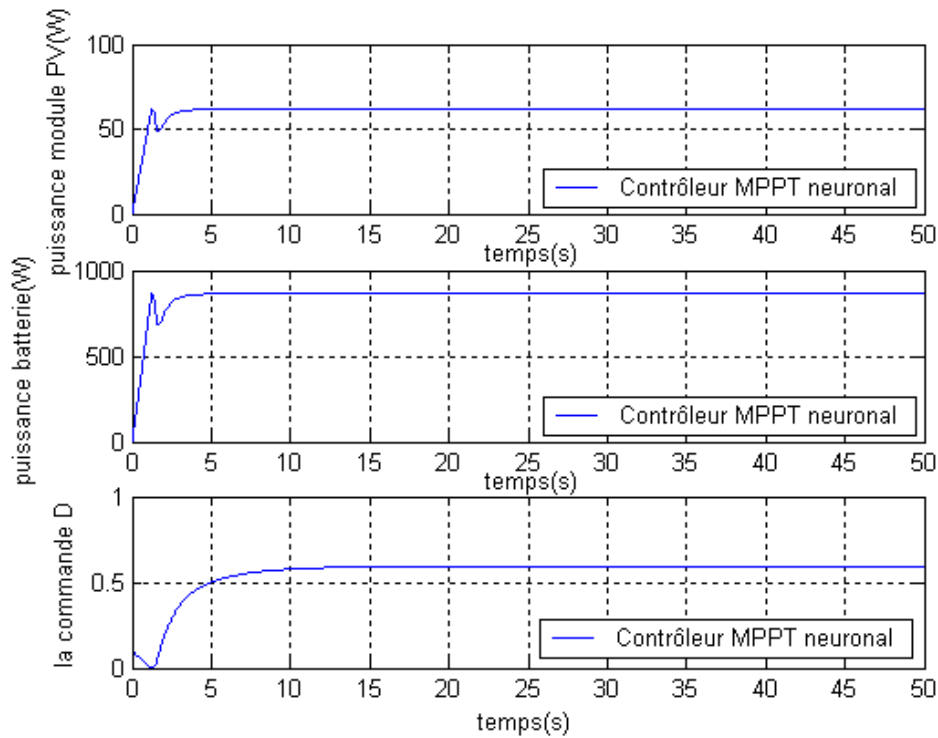


Figure III.9 : Variation de la puissance du module, la puissance de la batterie et de la commande D pour un ensoleillement de $1000\text{W}/\text{m}^2$ et une température de 25°C .

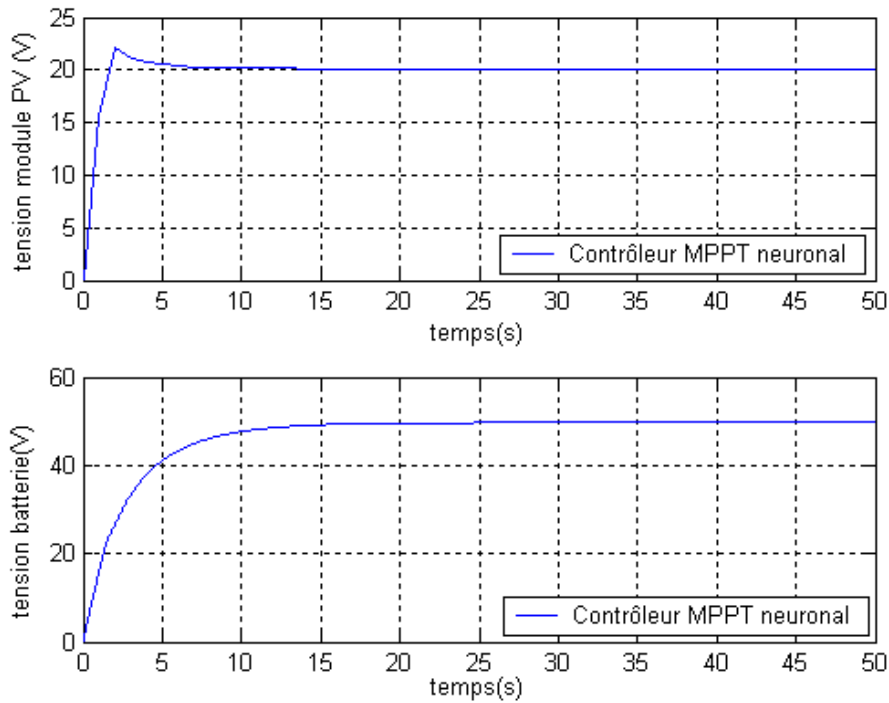


Figure III.10 : Courbe de variation des tensions de batterie et de module avec le contrôleur MPPT neuronal.

III.3.2.Simulation sous des conditions variables d'ensoleillement (changement de l'éclairement)

Afin d'évaluer les performances du mécanisme de poursuite par l'approche neuronale, ce dernier est soumis à des variations des conditions de l'ensoleillement qui consiste à effectuer une diminution et augmentation rapides suivies d'autres variations lentes de l'ensoleillement à une température constante.

❖ Augmentation rapide de l'ensoleillement (500 à 1000 W/m²)

Dans ce test, nous procédons à une augmentation de l'ensoleillement de 500 à 1000 W/m² pour une durée très courte de 5 secondes durant laquelle la température est maintenue constante égale à 25°C (figure III.11).

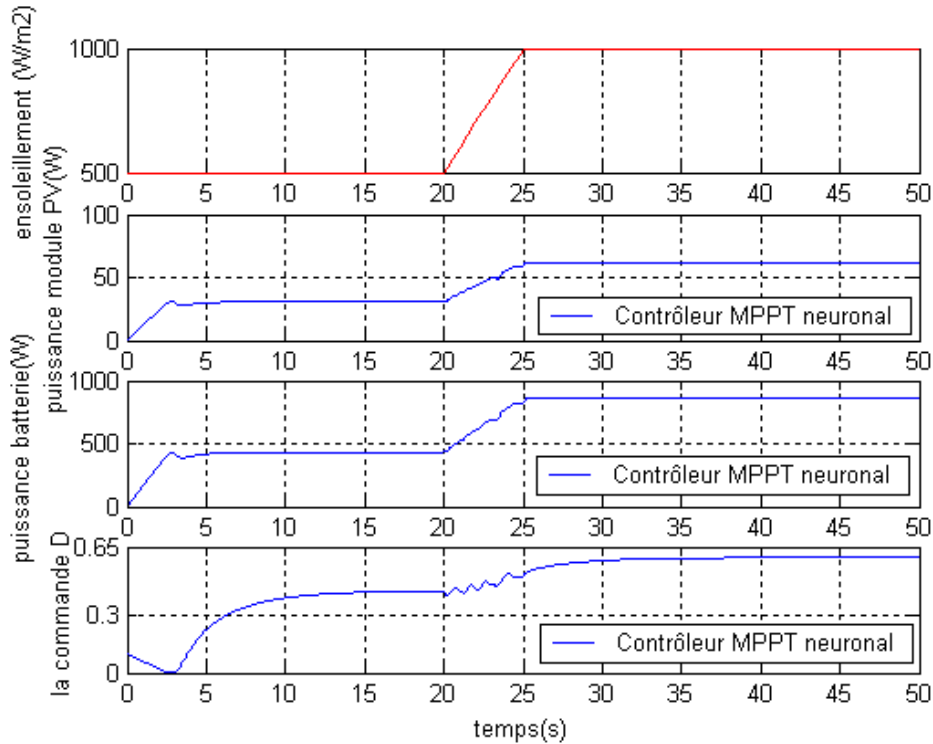


Figure III.11 : Variation de la puissance du module, la puissance de la batterie et de la commande D pour une augmentation rapide de l'ensoleillement de 500 à 1000 W/m^2 en 5 secondes avec une température constante de 25°C.

Nous remarquons que l'augmentation de l'insolation entraîne systématiquement celle de la puissance comme le montre la figure (III.11). Ce changement subit est instantanément détecté par le mécanisme de poursuite lequel essaye de s'adapter avec cette variation pour atteindre la puissance désirée (Réponse en temps réel).

❖ Diminution rapide de l'ensoleillement (1000 à 500 W/m^2)

Dans ce test la température est toujours maintenue constante à 25°C alors qu'on effectue une diminution de l'ensoleillement de 1000 à 500 W/m^2 dans un temps très court de 5 seconde (Fig. III.12).

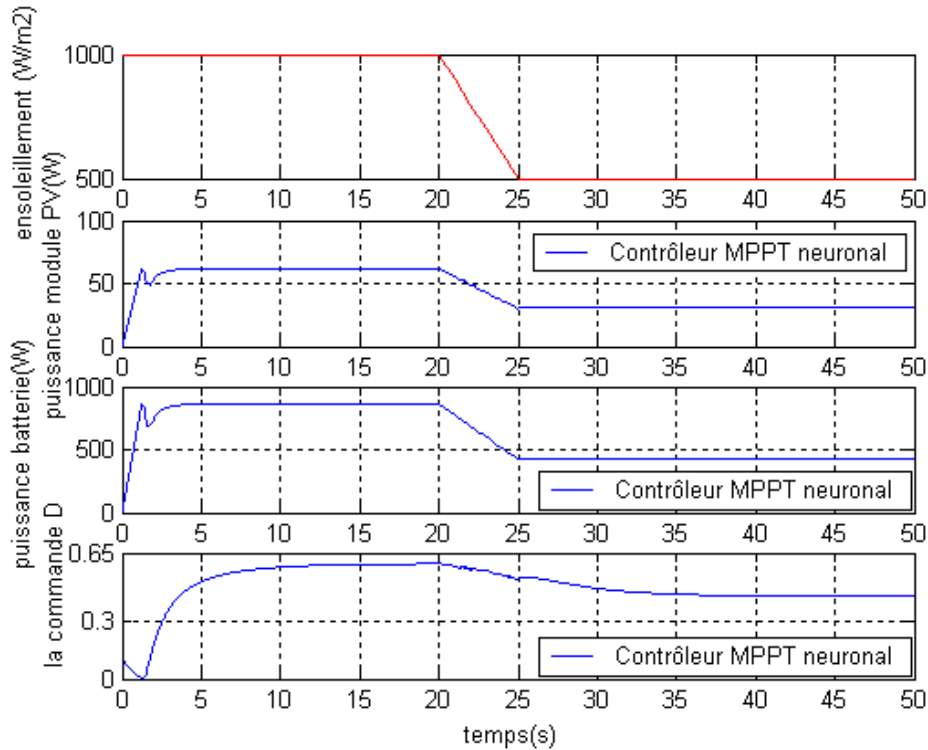


Figure III.12 : Variation de la puissance du module, la puissance de la batterie et de la commande D pour une diminution rapide de l'ensoleillement de 1000 à 500 W/m² en 5 secondes avec une température constante de 25°C.

La diminution rapide de l'insolation entraîne également une diminution rapide de la puissance. Le mécanisme de poursuite neuronal détecte cette diminution, et va réagir afin d'ajuster le rapport cyclique et cela par une légère diminution de sa valeur. Une fois l'insolation ne varie plus, le rapport cyclique se stabilise.

❖ Diminution lente de l'ensoleillement (1000 à 500 W/m²)

Une diminution lente de 1000W/m² à 500 W/m² de l'ensoleillement dans un intervalle de 120s est effectuée avec une température constante de 25°C (Fig.III.13)

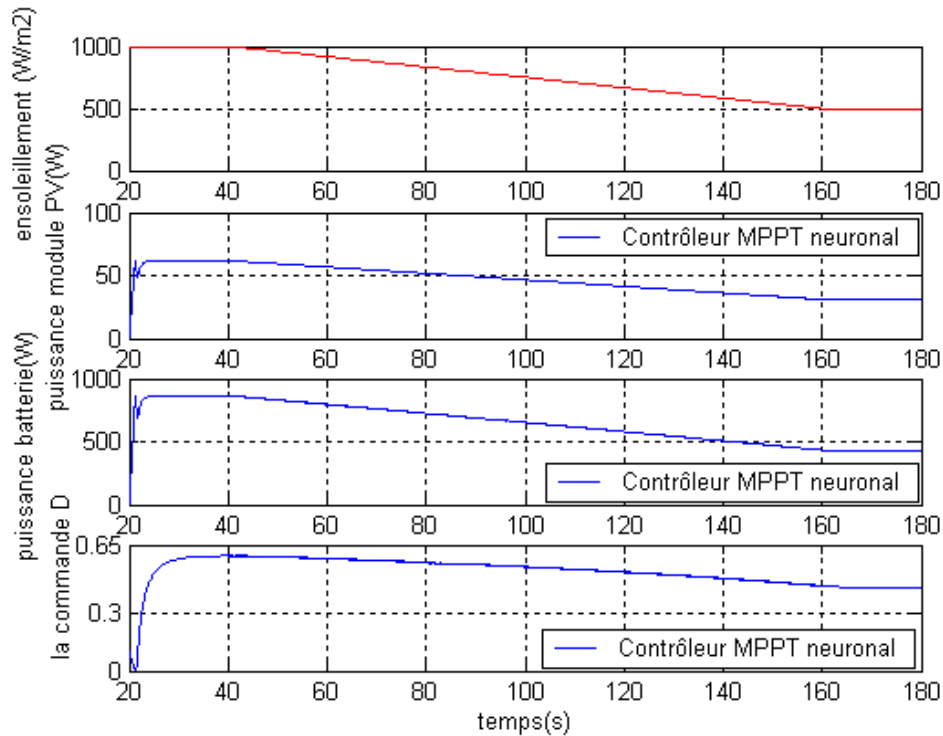


Figure III.13 : Variation de la puissance du module, la puissance de la batterie et de la commande D pour une diminution lente de l'ensoleillement de 1000 à 500 W/m² en 120 secondes avec une température constante de 25°C.

D'après le graphe de la figure III.13 il est évident que la diminution lente de l'ensoleillement a entraîné une diminution lente de la puissance et par conséquent un déplacement du point de puissance maximale, pendant ce temps la commande neuronale répond par une décrémentation jusqu'à la stabilisation une fois l'ensoleillement se stabilise

❖ Augmentation lente de l'ensoleillement (1000 à 500 W/m²)

Dans ce test l'ensoleillement augmente de 500W/m² à 1000W/m² dans un temps de 120 secondes alors que température est prise constante à 25°C. Ce cas de figure est illustré par les courbes suivantes (Fig III.14).

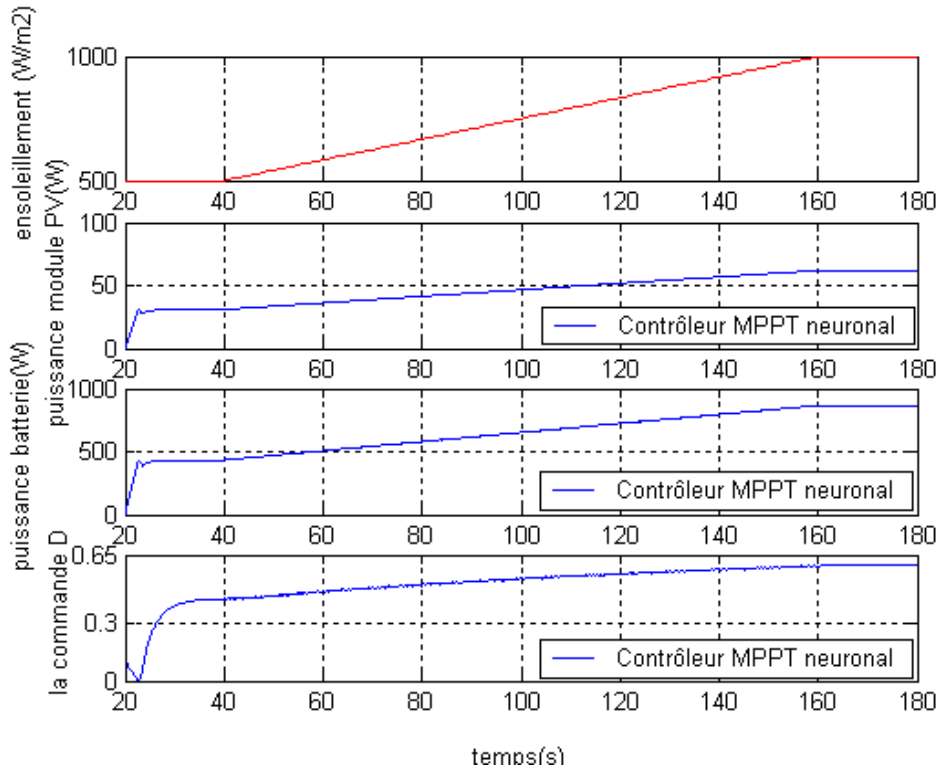


Figure III.14: Variation de la puissance du module, la puissance de la batterie et de la commande D pour une augmentation lente de l'ensoleillement de 500 à 1000 W/m² en 120 secondes avec une température constante de 25°C.

A son tour l'augmentation lente de l'insolation provoque également une augmentation lente de la puissance, le rapport cyclique détecte ce changement par une incrémentation instantanée afin d'atteindre le point de puissance maximale une fois l'insolation se stabilise. Le contrôleur MPPT neuronal effectue une poursuite parfaite du PPM.

III.3.3. Simulation sous des conditions variables de température (changement de la température)

La température de la cellule solaire est un facteur important qui influe sur les caractéristiques du panneau solaire et par conséquent sur la puissance. Nous allons effectuer les mêmes tests que précédemment.

❖ Augmentation lente de température de 20°C à 45 C°

L'insolation est maintenue constante (1000W/m²) alors qu'on augmente la température de 20°C à 45 °C durant un temps de 120 secondes. Les résultats de la simulation sont illustrés par les courbes de la figure (Fig III.15).

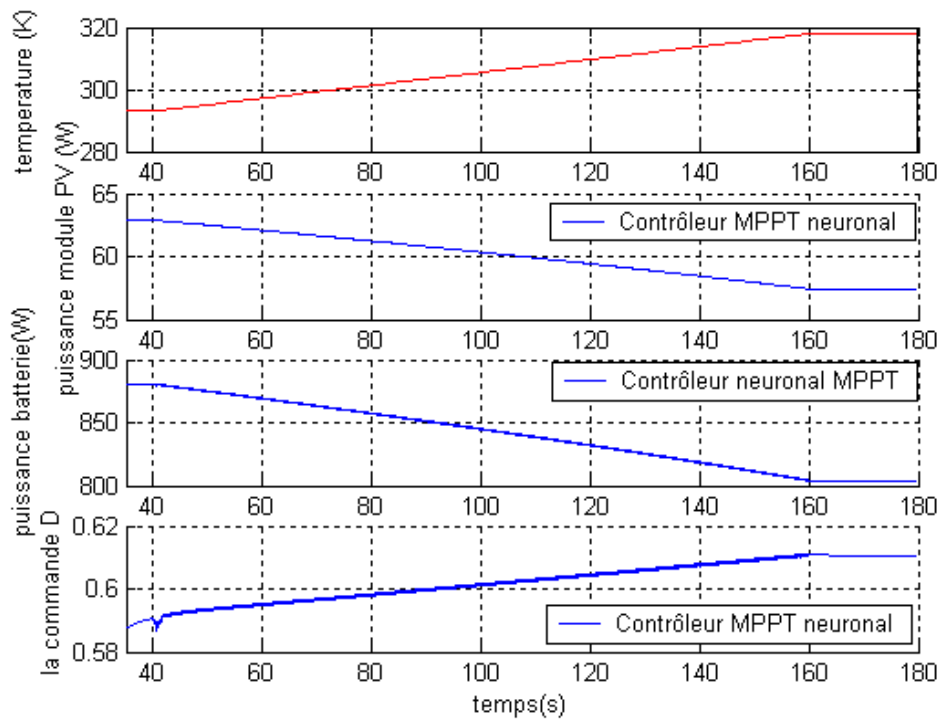


Figure III.15: Variation de la puissance du module, la puissance de la batterie et de la commande D pour une augmentation lente de la température de 20°C à 45°C en 120 secondes avec un ensoleillement de $1000\text{W}/\text{m}^2$.

L'augmentation lente de la température à son tour influence l'efficacité du système et par conséquent son rendement, cette augmentation va provoquer comme illustré dans la figure III.15 une diminution lente de la puissance, ce changement est aussi détecté par le contrôleur neuronal qui tente d'ajuster sa valeur par une légère augmentation pour atteindre le point de puissance maximale. Dans ce cas le contrôleur suit parfaitement cette diminution et n'effectue aucun retard.

❖ Diminution lente de la température (45°C à 20°C)

La figure III.16 montre les signaux de la commande D (rapport cyclique) et les puissances pour une diminution lente de la température de 45°C à 20°C avec un ensoleillement de $1000\text{W}/\text{m}^2$ produite dans un intervalle de temps de 120 secondes.

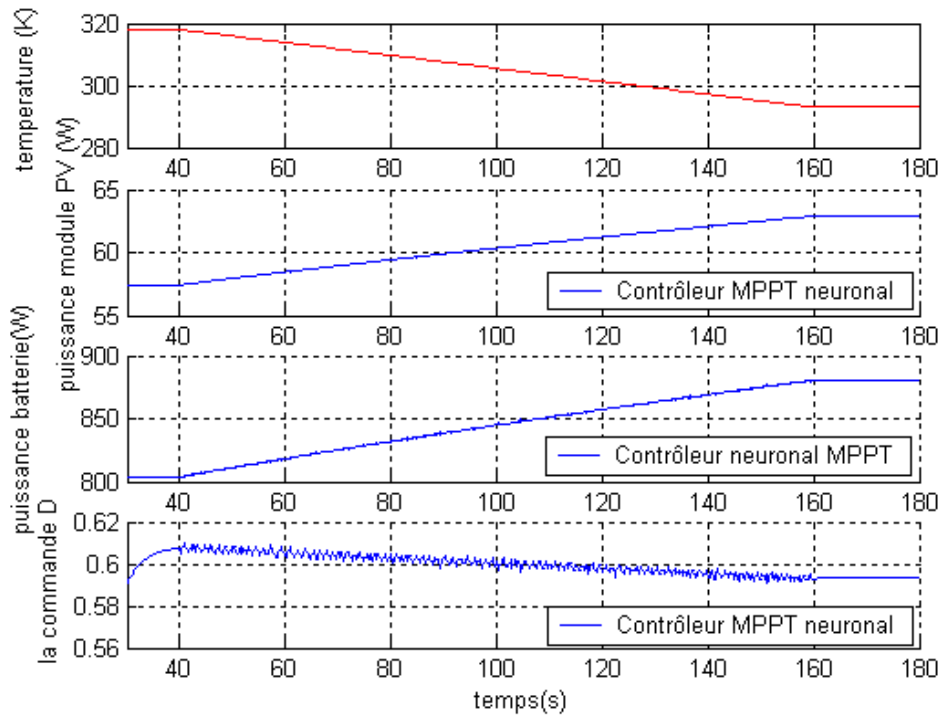


Figure III.16 : Variation de la puissance du module, la puissance de la batterie et de la commande D pour une diminution lente de la température de 45°C à 20°C et un ensoleillement de $1000\text{W}/\text{m}^2$.

❖ **Augmentation rapide de la température (20°C à 45°C)**

On fait subir à notre système une augmentation de la température de 20°C à 45°C durant un intervalle de temps de 5 secondes avec un ensoleillement de $1000\text{W}/\text{m}^2$.

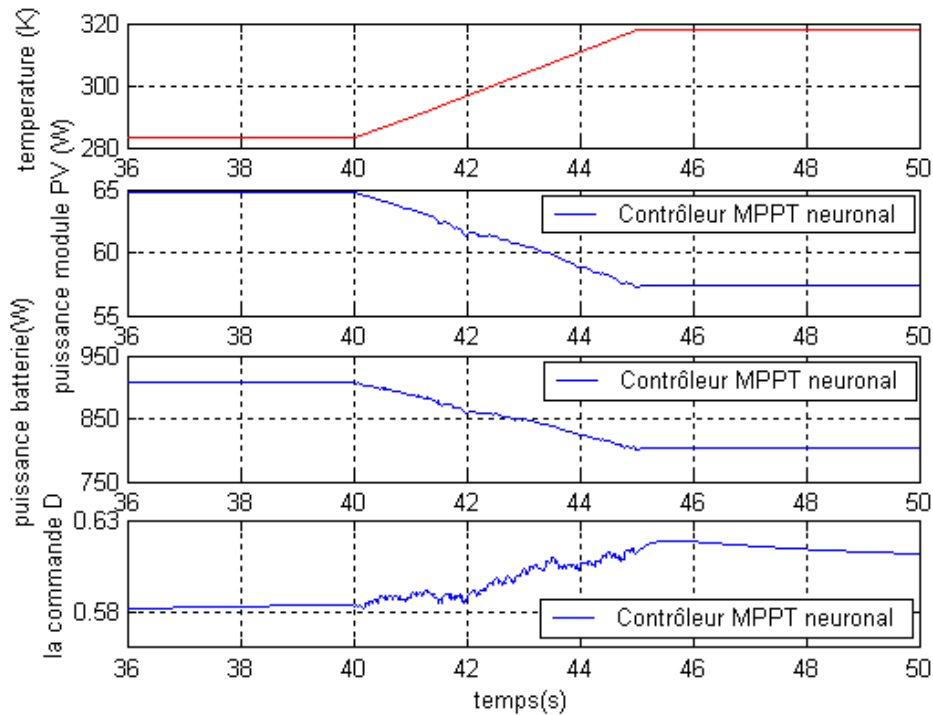


Figure III.17 : Variation de la puissance du module, la puissance de la batterie et de la commande D pour une augmentation rapide de la température de 20°C à 45°C en 5 secondes avec un ensoleillement de 1000W/m².

❖ **Diminution rapide de la température (45°C à 20°C)**

La figure III.18 montre les signaux du rapport cyclique les puissances pour une diminution rapide de la température de 45°C à 20°C dans un intervalle de 5 secondes, avec un ensoleillement de 1000W/m².

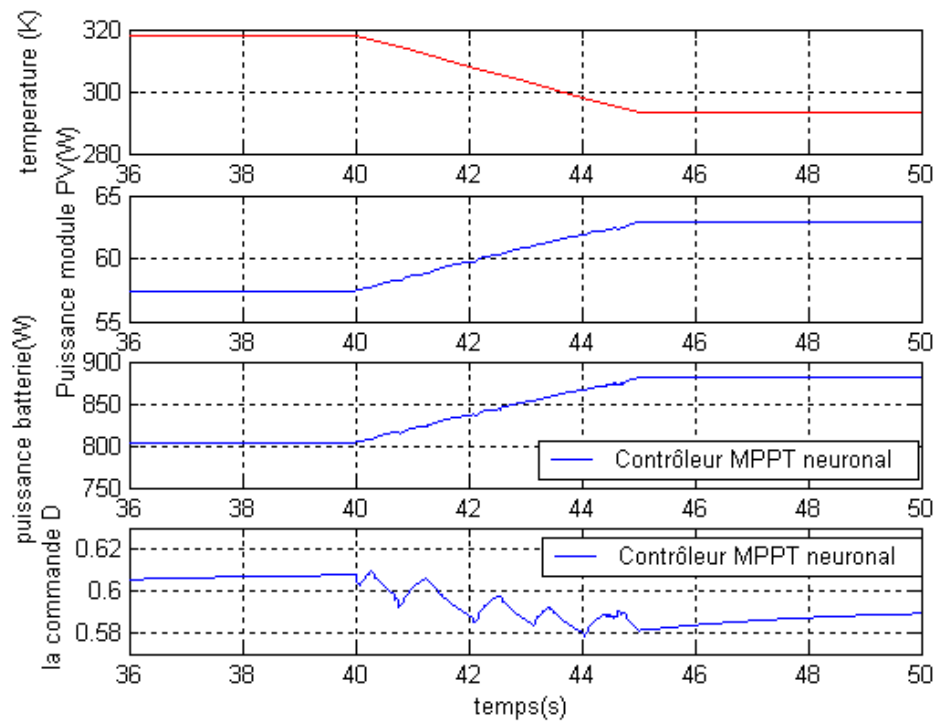


Figure III.18 : Variation de la puissance du module, la puissance de la batterie et de la commande D pour une diminution rapide de la température de 45°C à 20°C et un ensoleillement de 1000W/m².

III.3.4.Simulation sous des conditions aléatoires d'ensoleillement et de température

Dans la simulation suivante cette technique de poursuite neuronale est soumise à des conditions d'ensoleillement et de température bien réelles (aléatoires), la variation de l'ensoleillement est prise entre 400 et 1000W/m² tandis que la variation de la température est entre 20°C et 45°C. Les résultats sont illustrés à la figure III.19

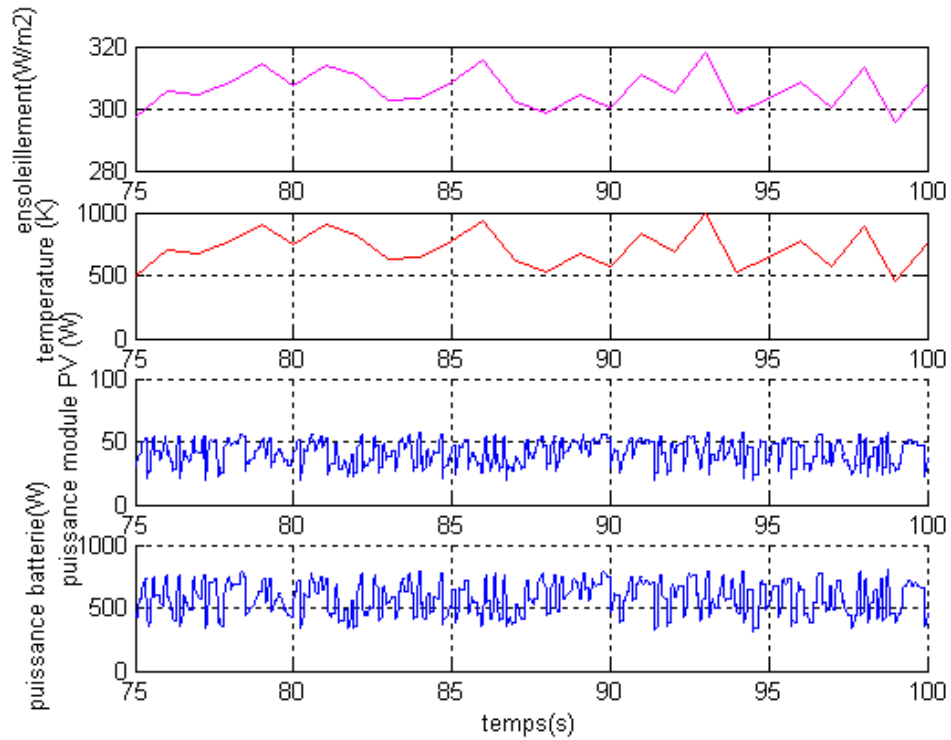


Figure III.20 : Variation de la puissance du module, la puissance de la batterie pour un changement aléatoire à une température de 20°C à 45°C et un ensoleillement de 400 à1000 W/m².

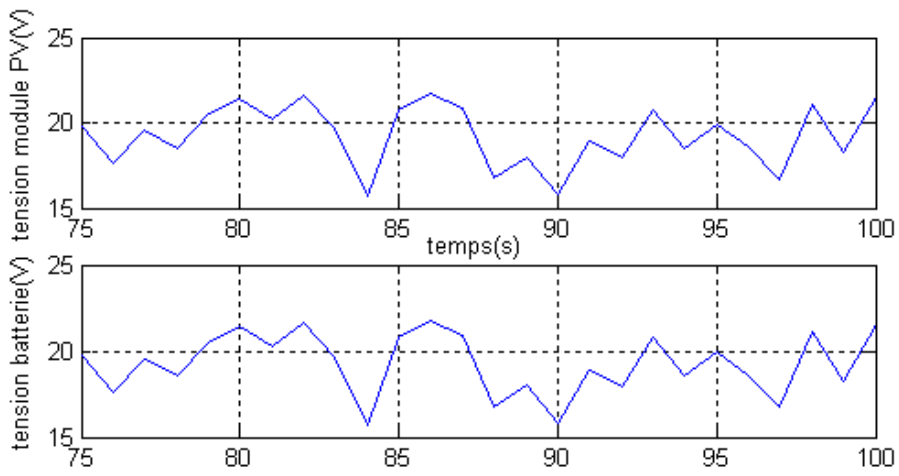


Figure III.21 : Variation des tensions de batterie et de module pour un changement aléatoire (une température de 20°C à 45°C et un ensoleillement de 400 à1000 W/m²).

III.4.Conclusion

L'utilisation des réseaux de neurones pour poursuivre le PPM a deux avantages :

- ❖ Premièrement, ils permettent l'apprentissage automatique à partir des situations et des résultats précédents.
- ❖ Deuxièmement, ils peuvent modéliser les relations complexes existantes entre la variable dépendante et indépendante (paramètres d'entrées et sortie)

Cependant, l'approche neuronique a des limitations qui l'empêchent d'être acceptée comme une pratique usuelle dans la commande :

- L'approche neuronique est considérée comme étant une approche 'boîte noire'. Par conséquent, il est difficile d'expliquer et d'interpréter son processus.
- Il n'existe aucune démarche standard pour le choix des différents paramètres de la topologie d'un réseau de neurones (nombre de couches, nombre d'unités de traitement par couche, valeurs initiales des poids de connexions, etc.).

Chapitre IV

Conception d'un contrôleur MPPT neuro-flou

IV. 1.Introduction

Dans ce travail, nous développons une méthode intelligente et innovante, combinant les deux techniques précédentes (floue et neuronale) ayant pour objet d'améliorer les réponses obtenues avec ces techniques du point de vue poursuite du point de puissance maximale quelle que soit la variation des conditions climatiques (ensoleillement, température,.....etc.) même dans les cas les plus défavorables. Pour cela nous allons décrire dans un premier temps le contrôleur MPPT neuro-flou proposé, la présentation de l'architecture neuro-floue (NF) choisie, la procédure d'apprentissage adoptée et les règles d'inférence générées à cet effet. Des simulations et leurs résultats vont être mis en place afin de discuter les performances de ce contrôleur.

L'utilisation de la méthode hybride "*ANFIS*" (adaptive network based fuzzy inference system) a les avantages suivants :

- Combinaison des approches logique floue et réseaux de neurones ;
- L'objectif est d'optimiser un ensemble de règles floues par apprentissage ;
- L'idée principale est de construire un *FIS* (fuzzy inference system) comme un réseau de neurone spécialisé et d'appliquer des règles classique d'apprentissage

IV.2. Conception d'un contrôleur MPPT à base de réseau neuro-flou

La figure IV.1 représente le schéma synoptique d'un système photovoltaïque doté de commande MPPT à base de réseaux neuro-flou.

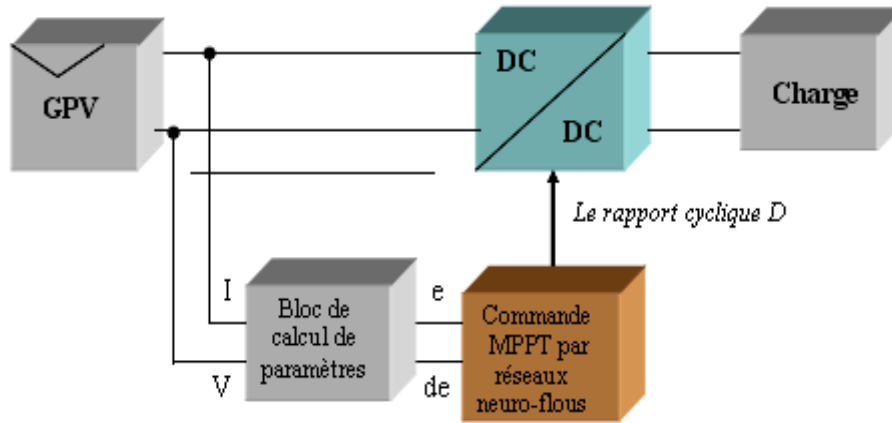


Figure IV.1. Schéma synoptique d'un système photovoltaïque avec une commande MPPT par réseau neuro-flou (ANFIS).

IV.2.1 : Description et structure du contrôleur MPPT neuro-flou proposé

Le contrôleur neuro-flou développé dans cette partie comprend deux entrées ' e ' et ' de ' et une seule sortie ' D ' qui représentent respectivement l'erreur, la variation de l'erreur, et la commande. Les deux variables d'entrées génèrent l'action de contrôle ' D ' qui sera appliquée au hacheur, afin d'ajuster le rapport cyclique de ce dernier de telle manière à assurer l'adaptation de la puissance fournie par le GPV.

Le contrôleur neuro-flou est un contrôleur *ANFIS* qui met en application un système d'inférence flou (*SIF*) du type Takagi Sugeno et à une architecture composée de cinq couches comme représentée sur la figure (IV.2). Ce contrôleur permet une génération automatique de règles floues basées sur le modèle d'inférence de Sugeno :

$$\begin{aligned}
 & \text{Si } e \text{ est } A_1 \text{ et } de \text{ est } B_1 \quad \text{Alors } d_1 = f(e, de) \\
 & \text{Si } e \text{ est } A_2 \text{ et } de \text{ est } B_1 \quad \text{Alors } d_2 = f(e, de) \\
 & \quad \quad \quad \vdots \\
 & \text{Si } e \text{ est } A_5 \text{ et } de \text{ est } B_5 \quad \text{Alors } d_{25} = f(e, de)
 \end{aligned}$$

Où e , de , sont des variables d'entrée et A_1, A_2, \dots, A_5 et B_1, B_2, \dots, B_5 sont des ensembles flous.

Lorsque $d=$ constante, on obtient un modèle de Sugeno d'ordre zéro. Le conséquent d'une règle est un singleton.

Lorsque d est une combinaison linéaire des entrées : $d= a e + b de + c$.

On obtient un modèle de Sugeno de premier ordre.

La figure IV.2 montre un exemple d'application de deux règles en utilisant la méthode de défuzzification "moyennes pondérées" ou encore "weighted average".

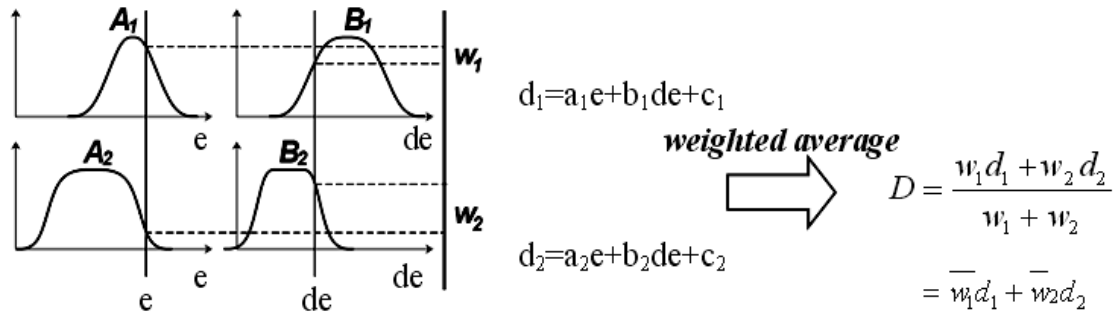


Figure IV. 2 : Exemple de défuzzification par la méthode weighted average.

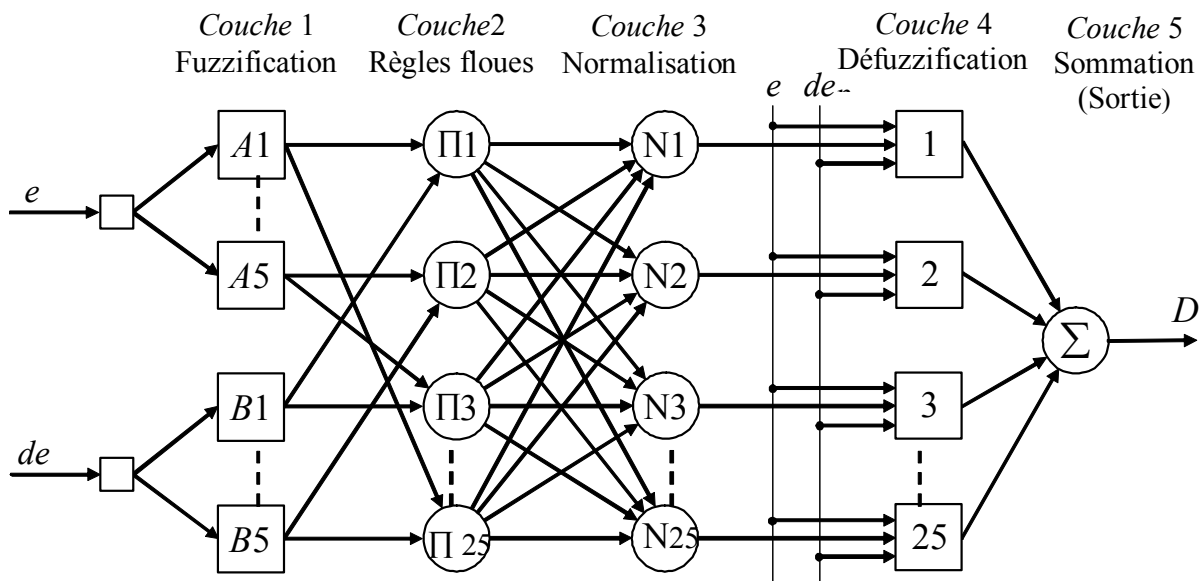


Figure IV.3 : Architecture du modèle ANFIS proposé.

Les nœuds d'entrées e et de transmettent simplement les données d'entrée à la couche de fuzzification.

Les cinq couches du réseau neuro-flou ont pour rôle :

Couche 1 :

Les neurones de cette couche réalisent les ensembles flous qui serviront dans les antécédents des règles. Dans le modèle de Jang, les fonctions d'appartenance sont des gaussiennes.

Couche 2 :

Chaque neurone dans cette couche correspond à une règle floue Sugeno. Il reçoit les sorties des neurones de fuzzification et calcule son activation. La conjonction des antécédents est réalisée avec l'opérateur produit.

Couche 3 :

Chaque neurone calcule le degré de vérité normalisé d'une règle floue donnée. La valeur obtenue représente la contribution de la règle floue au résultat final.

Couche 4 :

Chaque neurone i de cette couche est relié à un neurone de normalisation correspondant et aux entrées initiales du réseau. Il calcule le conséquent.

Couche 5 :

Comprend un seul neurone qui fournit la sortie de ANFIS en calculant la somme des sorties de tous les neurones de défuzzification

La structure neuronale équivalente proposée sous Matlab est représentée à la figure IV.4.

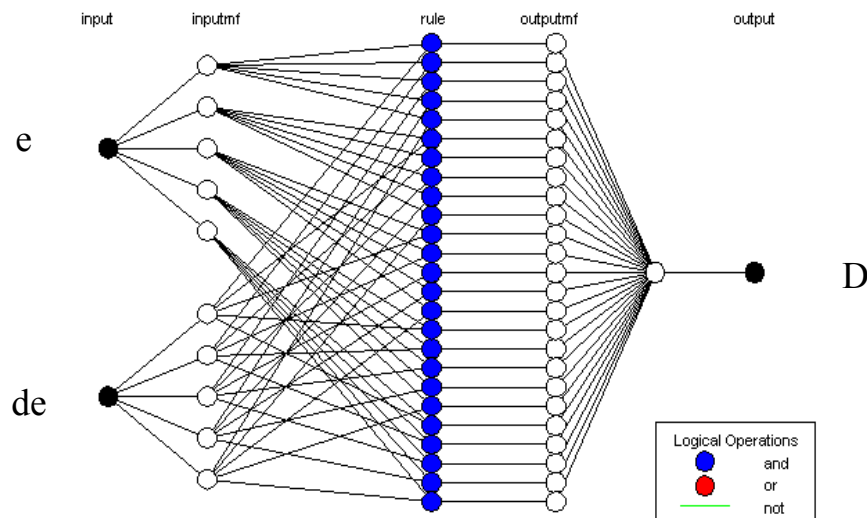


Figure IV.4 : Structure neuronale du modèle proposé sous Matlab.

IV. 2.2. Apprentissage du contrôleur "Entraînement d'un réseau ANFIS "

L'apprentissage du notre contrôleur est effectué par le biais de l'algorithme de rétro propagation, afin de déterminer les paramètres des prémisses (ajustement des paramètres liés aux fonctions d'appartenance) et l'estimation des paramètres conséquents par la méthode des moindres carrés. Ce qui a pour appellation "*apprentissage hybride*".

L'entraînement du réseau ANFIS est donc effectué par un algorithme à deux temps où on estime d'abord les paramètres des conséquents par une technique de moindres carrés et ensuite les poids du réseau par une descente de gradient.

Chaque époque d'entraînement comprend une passe avant et une passe arrière. Durant la passe avant, les patrons d'entrée servent à déterminer les sorties des neurones couche par couche, permettant de déterminer les valeurs des paramètres de conséquents en bout de compte.

Durant la passe arrière, l'algorithme de rétropropagation d'erreur est appliqué pour régler les poids des différentes couches.

Lors de la passe arrière, l'algorithme de rétro propagation d'erreur est appliqué pour mettre à jour les poids des antécédents des règles.

Dans l'algorithme ANFIS de Jang, on optimise aussi bien les paramètres de antécédents que ceux des conséquents. Durant la passe avant, les paramètres des conséquents sont adaptés alors que les paramètres des antécédents sont maintenus constants ; durant la passe arrière, les rôles sont échangés.

Le tableau suivant illustre le rôle des deux passes pour le réglage des paramètres antécédents et conséquents pendant la phase d'apprentissage :

	Passe avant	Passe arrière
Les paramètres prémisses	Fixés	Descente de gradient
Les paramètres conséquents	Moindre carré	Fixes
Signal	de nœud sortie	Taux d'erreur

Tableau IV.1 : Illustration des deux étapes d'apprentissage.

- **Utilisation du ANFIS comme modèle de base pour la poursuite du point de puissance maximale**

Une fois que le vecteur (d'entrée-sortie) des données mesurées est obtenu, et comme *ANFIS* fait partie de la boîte à outil "logique floue" du logiciel Matlab, il a été utilisé efficacement pour construire l'ensemble des règles floues Si Alors avec leur

fonctions d'appartenance appropriées. La manière dont ANFIS fonctionne est décrite dans l'organigramme de la figure IV.5.

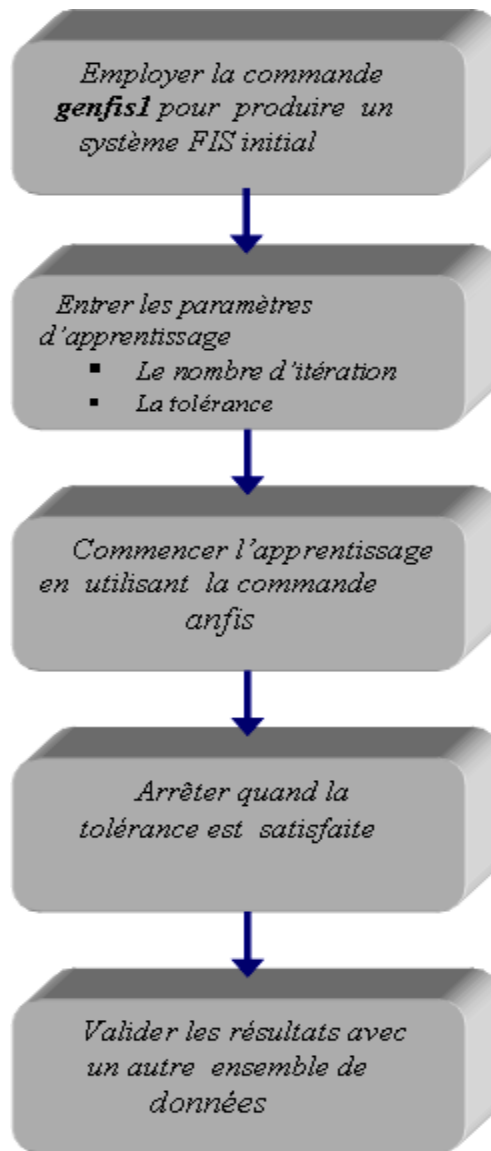


Figure IV.5: Organigramme du processus de calcul d'ANFIS dans Matlab.

Il est important de noter qu'il n'existe pas une loi pour déterminer la table des règles et le nombre de fonctions d'appartenance assigné à chaque variable d'entrée. Généralement, ce nombre est choisi empiriquement en examinant son influence sur la réaction du système. Le résultat du processus d'apprentissage est montré sur la figure (IV.6).

- La base de donnée utilisées pour l'apprentissage du réseau est la même que celle utilisée pour l'entraînement RN et cela dans le but de mieux évaluer les capacités

et les performances de celui-ci. Le résultat du processus d'apprentissage est montré sur la figure (IV.6).

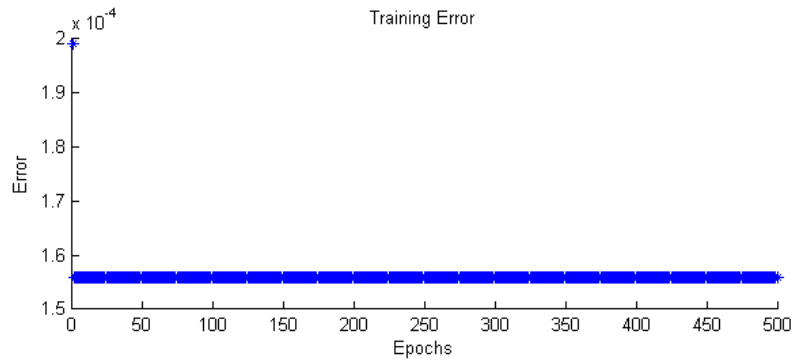


Figure IV.5 : Entraînement du réseau neuro-flou (avec un nombre d'itération de 500).

IV. 2. 3. Test de la capacité de généralisation après apprentissage

Après apprentissage du système neuro-flou adaptatif (ANFIS), nous avons testé sa généralisation, en présentant des données de validation (Checking Data).

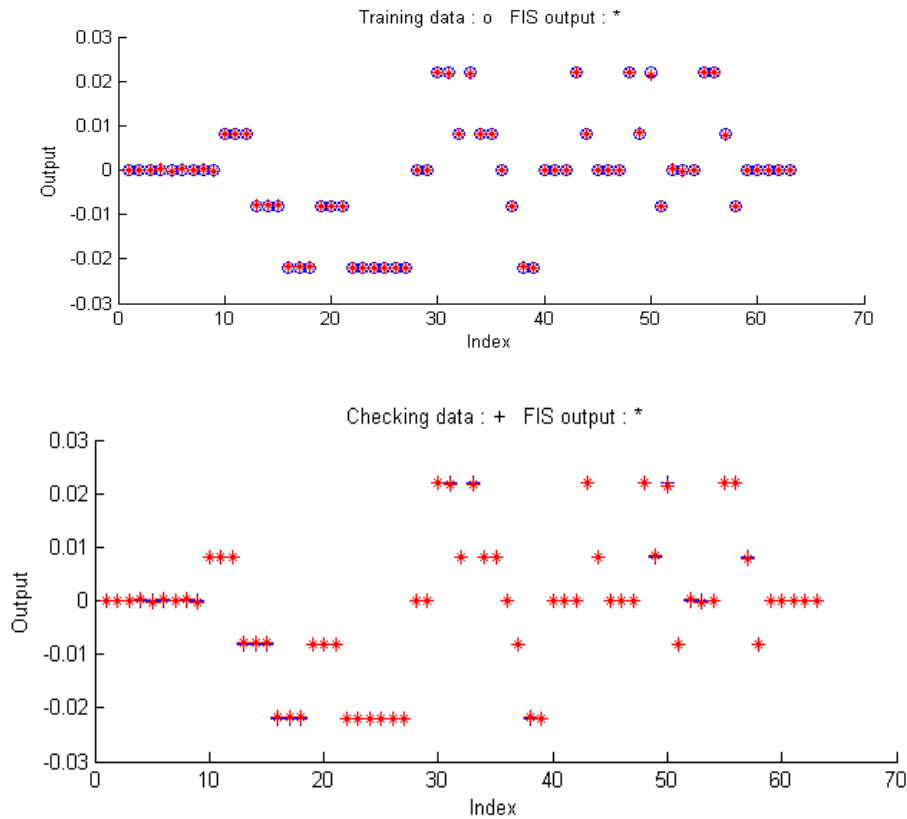


Figure IV. 6 : Test de la capacité de généralisation après apprentissage.

Pour déterminer le pas d'apprentissage et les fonctions d'apprentissage que nous avons élaboré un programme susceptible de les déterminer.

Les résultats obtenus sont illustrés à la figure IV.7, IV.8 et IV.9.

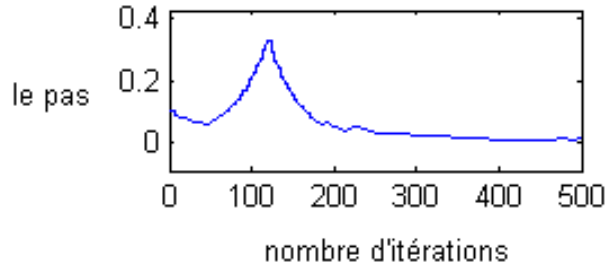


Figure IV.7 : Le pas d'apprentissage.

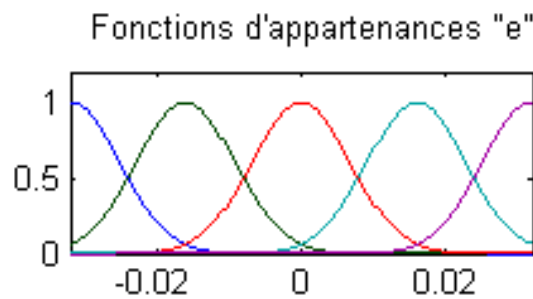


Figure IV.8 : Les fonctions d'appartenance de l'erreur "e" après apprentissage.

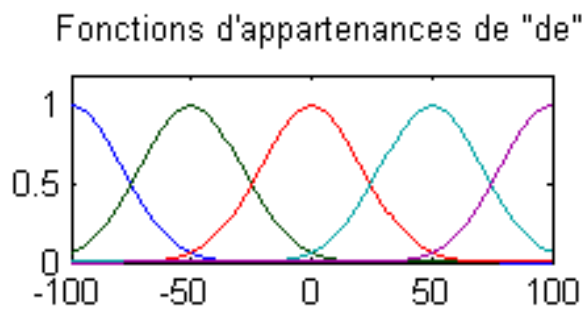


Figure IV.9 : Les fonctions d'appartenance du changement de l'erreur "de" après apprentissage

Le tableau suivant montre les règles générées par le ANFIS :

e \ de	A ₁	A ₂	A ₃	A ₄	A ₅
B ₁	d ₁	d ₂	d ₃	d ₄	d ₅
B ₂	d ₆	d ₇	d ₈	d ₉	d ₁₀
B ₃	d ₁₁	d ₁₂	d ₁₃	d ₁₄	d ₁₅
B ₄	d ₁₆	d ₁₇	d ₁₈	d ₁₉	d ₂₀
B ₅	d ₂₁	d ₂₂	d ₂₃	d ₂₄	d ₂₅

Tableau IV.2. Table des règles générées.

Une fois le contrôleur neuro-flou conçu et testé, il est inséré dans le système photovoltaïque. Une simulation numérique est par la suite effectuée.

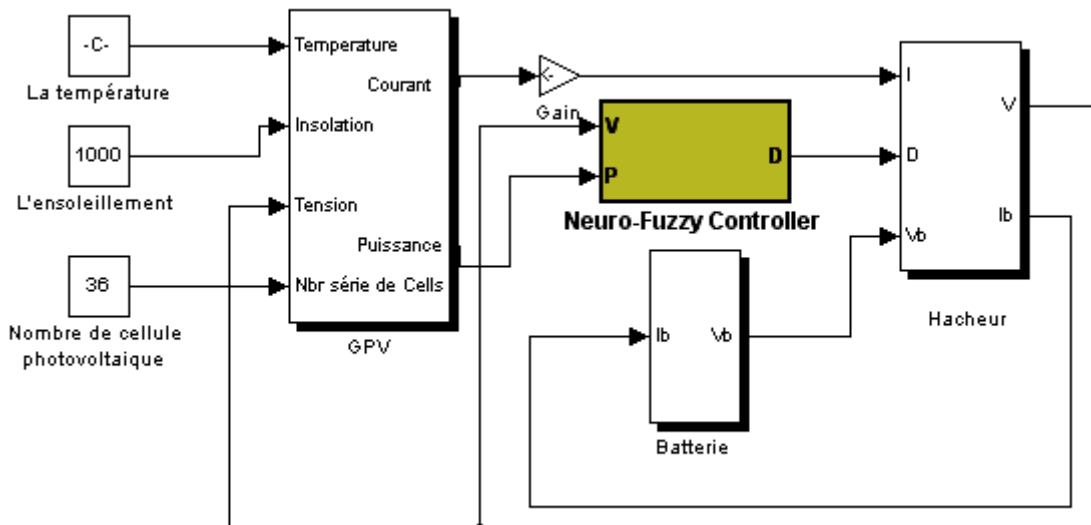


Figure IV.11. : Schéma synoptique d'un système photovoltaïque avec le contrôleur neuro-flou sous Simulink.

Le système photovoltaïque utilisé pour la simulation est principalement composé des blocs suivants : un générateur photovoltaïque, un hacheur, une batterie et le neuro-flou (Fig. IV.11).

IV.3. Résultats de simulation de l'application de l'approche neuro-floue à la poursuite du point de puissance maximale

Dans cette section, nous allons examiner les effets de cette hybridation, illustrés par les graphiques suivants qui mettent ainsi en évidence l'apport de la recherche globale dans l'amélioration des résultats d'apprentissage classique des réseaux de neurones.

Une fois le contrôleur neuro-flou conçu et testé, il est inséré dans le système photovoltaïque. Une simulation numérique est par la suite effectuée. Dans cette section nous allons faire subir notre contrôleur aux mêmes conditions de test que précédemment.

IV.3.1. Simulation du fonctionnement sous des conditions standard (constantes)

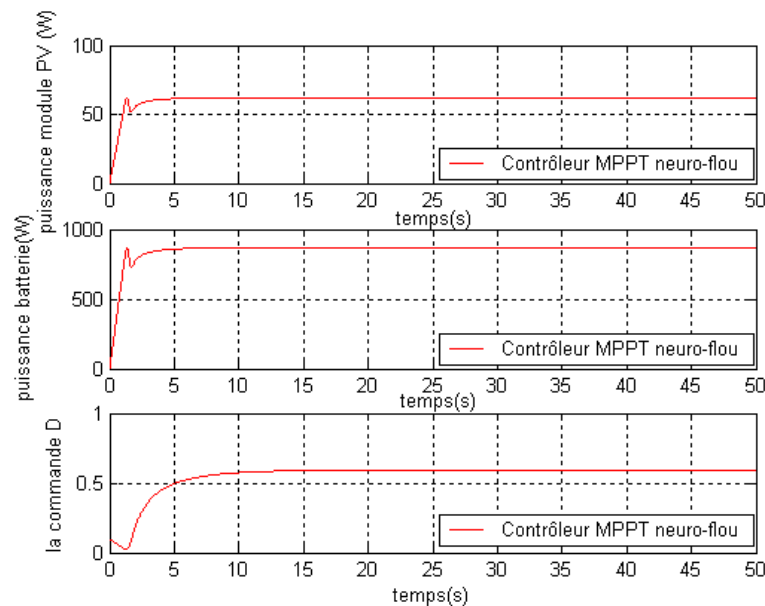


Figure IV. 12 : Variation de la puissance du module, la puissance de la batterie et de la commande D pour une température de 25°C et un ensoleillement de 1000w/m².

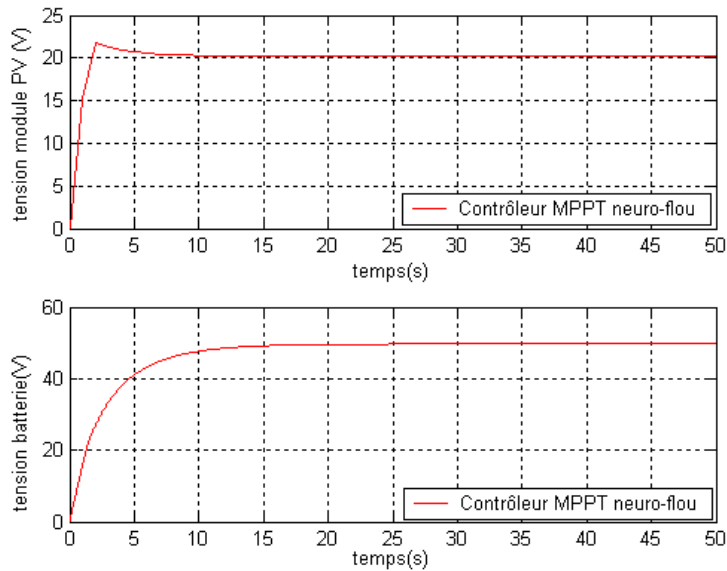


Figure IV.13 : Courbe de variation des tensions de batterie et de module avec le contrôleur neuro-flou.

IV.3.2. Simulation des conditions variables d'ensoleillement

❖ Augmentation rapide de l'ensoleillement (500 à 1000 W/m²)

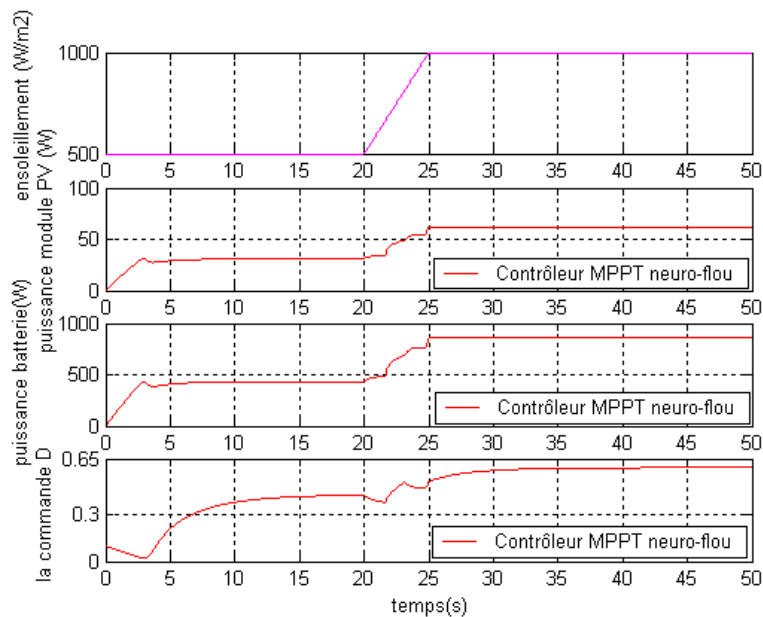


Figure IV.14 : Variation de la puissance du module, la puissance de la batterie et de la commande D pour une augmentation rapide de l'ensoleillement de 500 à 1000 W/m² en 5 secondes avec une température constante de 25°C.

❖ Diminution rapide de l'ensoleillement (1000 à 500 W/m²)

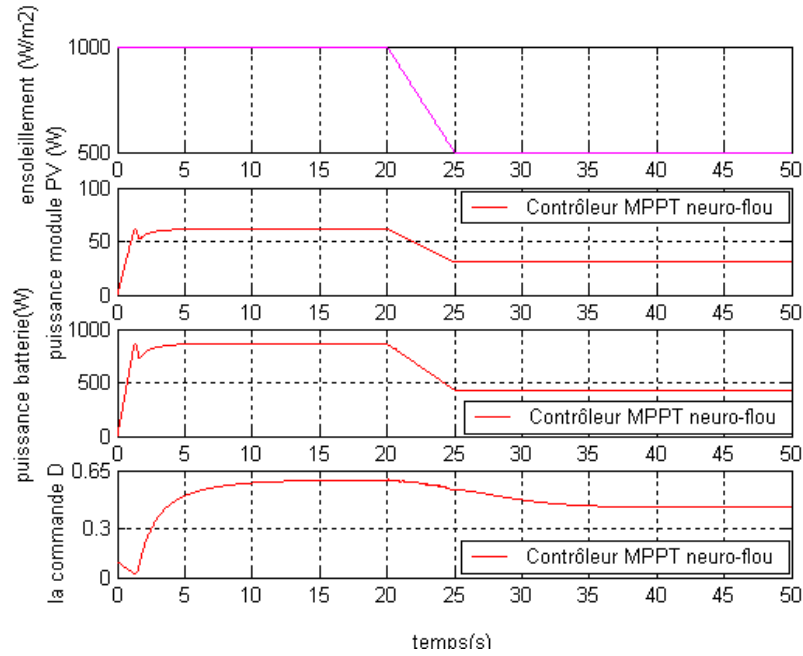


Figure IV.15 : Variation de la puissance du module, la puissance de la batterie et de la commande D pour une diminution rapide de l'ensoleillement de 1000 à 500 W/m² en 5 secondes avec une température constante de 25°C.

❖ Diminution lente de l'ensoleillement (1000 à 500 W/m²)

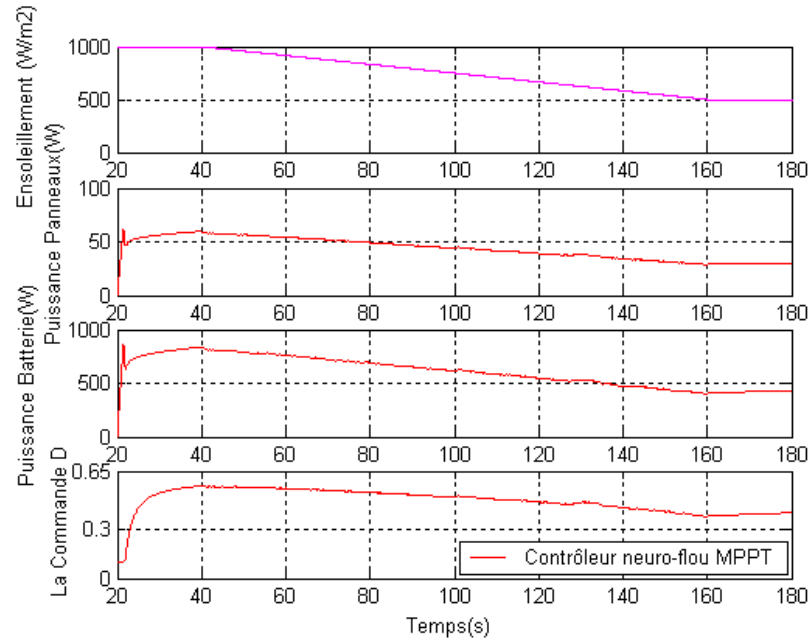


Figure IV.16 : Variation de la puissance du module, la puissance de la batterie et de la commande D pour une diminution lente de l'ensoleillement de 1000 à 500 W/m² en 120 secondes avec une température constante de 25°C.

❖ Augmentation lente de l'ensoleillement (500 à 1000 W/m²)

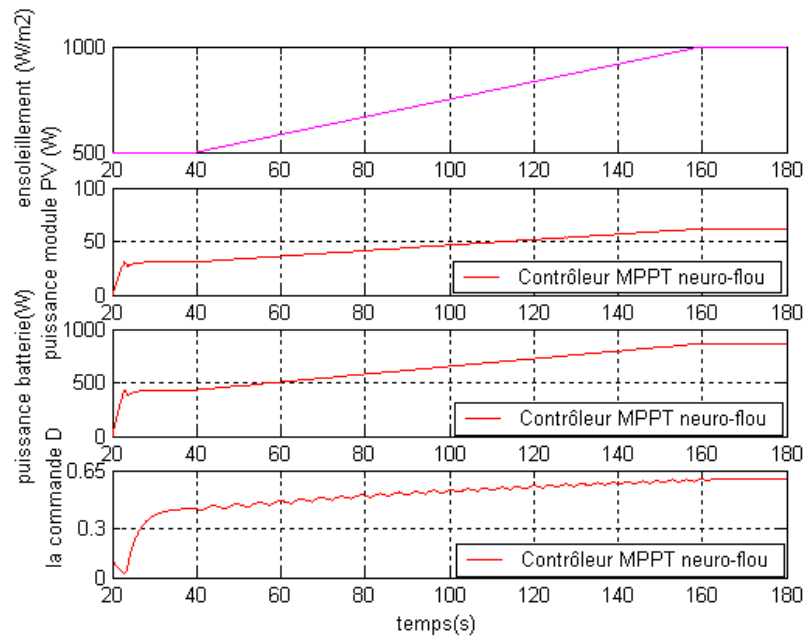


Figure IV.17 : Variation de la puissance du module, la puissance de la batterie et de la commande D pour une augmentation lente de l'ensoleillement de 1000 à 500 W/m² en 120 secondes avec une température constante de 25°C.

IV.3.3. Simulation sous des conditions variables de température (changement de la température)

❖ Augmentation rapide de la température (20°C à 45°C)

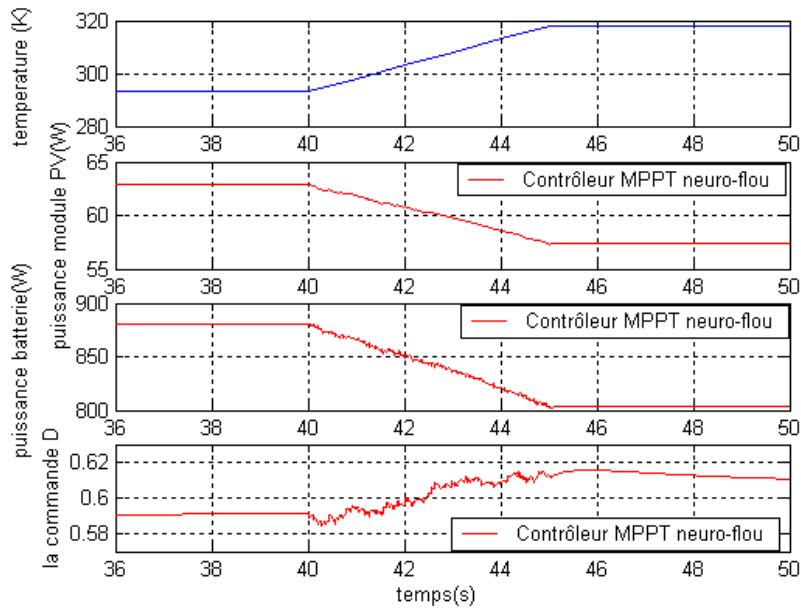


Figure IV.18 : Variation de la puissance du module, la puissance de la batterie et de la commande D pour une augmentation rapide de la température de 20°C à 45°C en 5 secondes avec un ensoleillement de 1000W/m².

❖ Diminution rapide de température (45°C à 20°C)

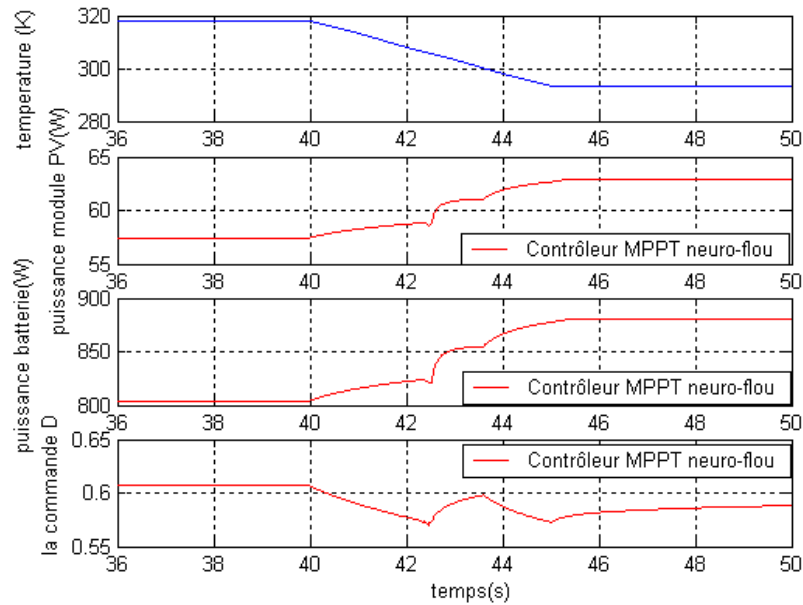


Figure IV.19: Variation de la puissance du module, la puissance de la batterie et de la commande D pour une diminution rapide de la température de 45°C à 20°C en 5 secondes avec un ensoleillement de 1000W/m².

❖ Augmentation lente de la température (20°C à 45°C)

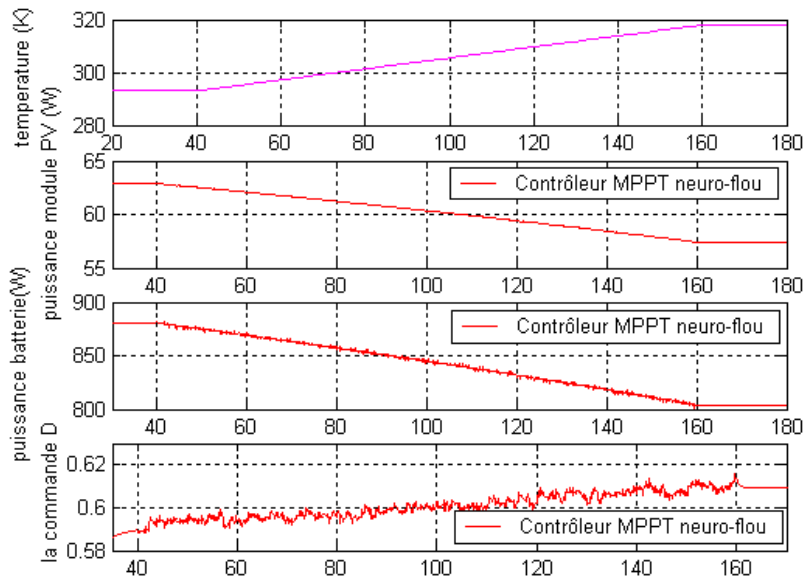


Figure IV.20: Variation de la puissance du module, la puissance de la batterie et de la commande D pour une augmentation lente de la température de 20°C à 45°C et un ensoleillement de 1000W/m².

❖ Diminution lente de la température (45°C à 20°C)

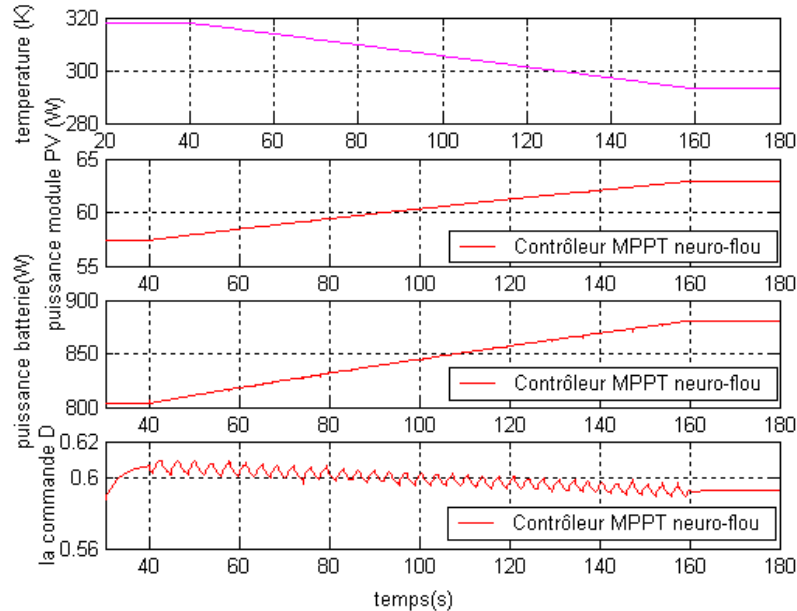


Figure IV.21: Variation de la puissance du module, la puissance de la batterie et de la commande D pour une diminution lente de la température de 45°C à 20°C en 120 secondes avec un ensoleillement de 1000W/m².

IV.3.4.Simulation sous des conditions aléatoires d'ensoleillement et de température

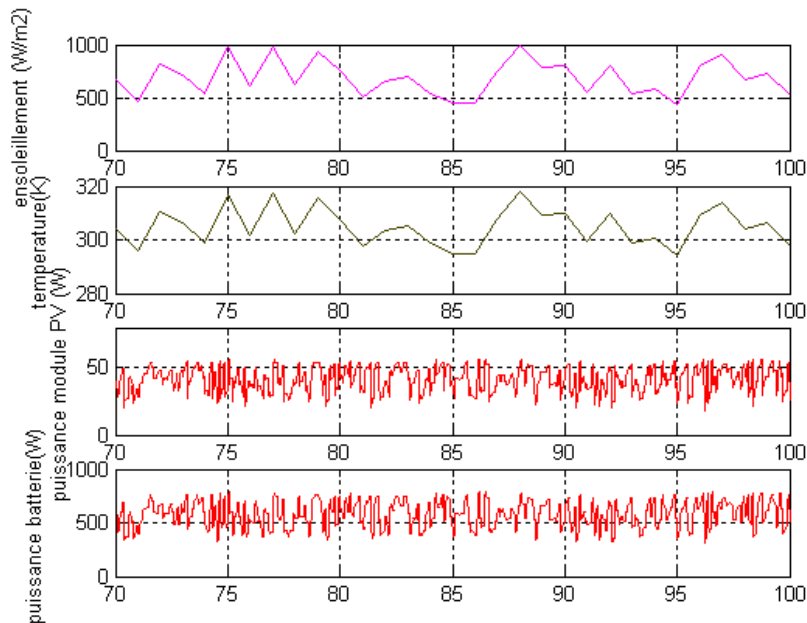


Figure IV.22 : Variation de la puissance du module, la puissance de la batterie et de la commande D pour un changement aléatoire à une température de 20°C à 45°C et un ensoleillement de 400 à 1000 W/m².

IV.4 .Synthèse et comparaison des résultats des trois techniques (floue, neuronale, neuro-floue)

Les sections III.3 et IV.3 étaient consacrés à la présentation des deux contrôleurs développés (neuronal, neuro-flou). Dans cette partie, nous allons effectuer une étude comparative des trois approches pour la poursuite de PPM (floue, neuronale, et neuro-floue) mettant ainsi en évidence l'efficacité de chaque technique pour permettre à la fin de montrer la meilleure solution à adopter.

III.4.1.Simulation du fonctionnement sous des conditions standard (constantes)

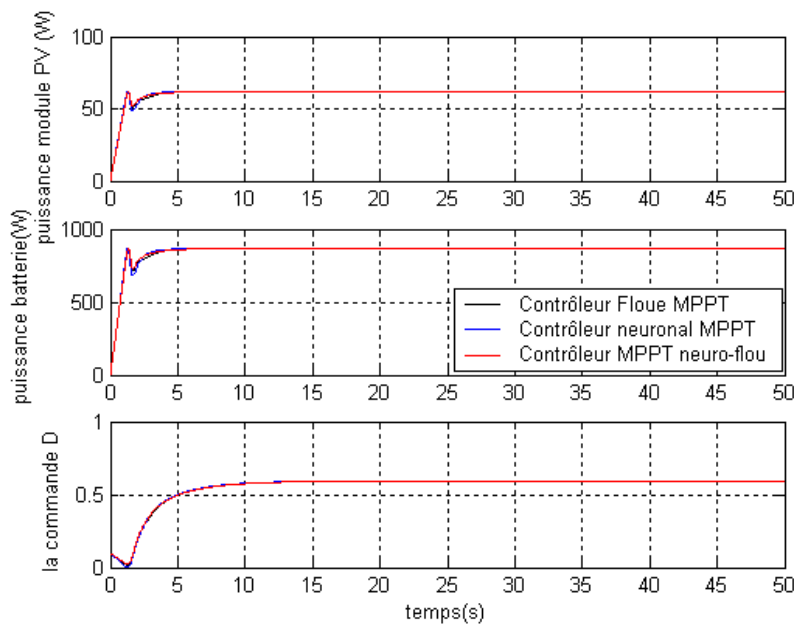


Figure IV.23 : Variation de la puissance du module, la puissance de la batterie et de la commande D pour un ensoleillement de $1000\text{W}/\text{m}^2$ avec une température de 25°C .

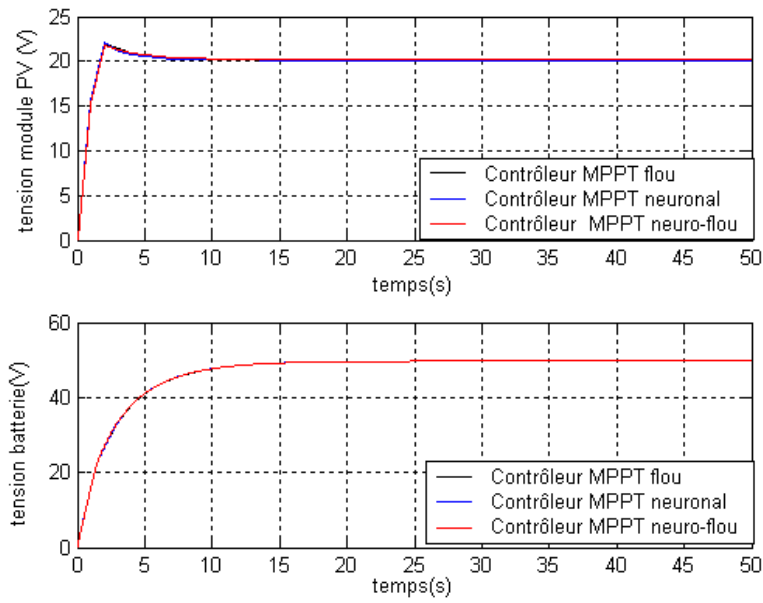


Figure IV.24 : Courbe de variation des tensions de batterie et de module.

II.4.2.Simulation sous des conditions variables d'ensoleillement

❖ Diminution rapide de l'ensoleillement (1000 à 500W/m²)

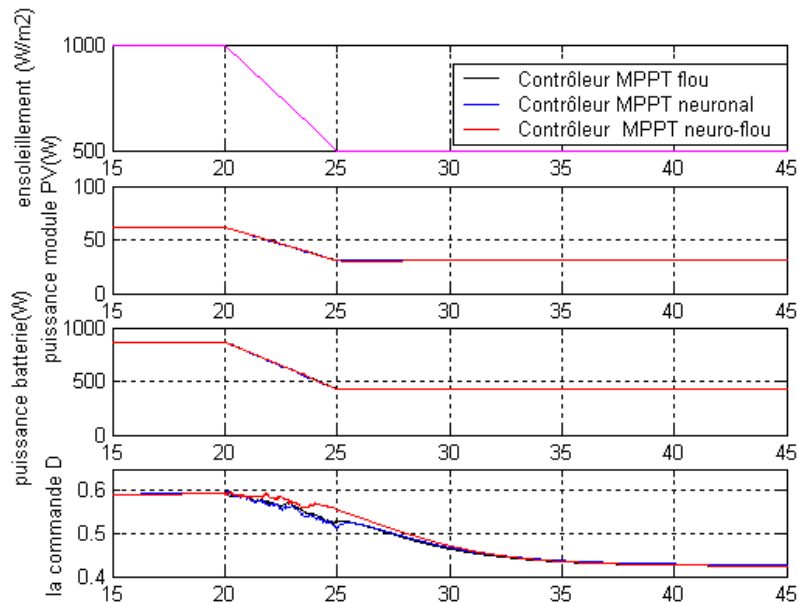


Figure IV.25 : Variation de la puissance du module, la puissance de la batterie et de la commande D pour une diminution rapide de l'ensoleillement de 1000 à 500 W/m² en 5 secondes avec une température constante de 25°C.

❖ Augmentation rapide de l'ensoleillement (500 à 1000W/m²)

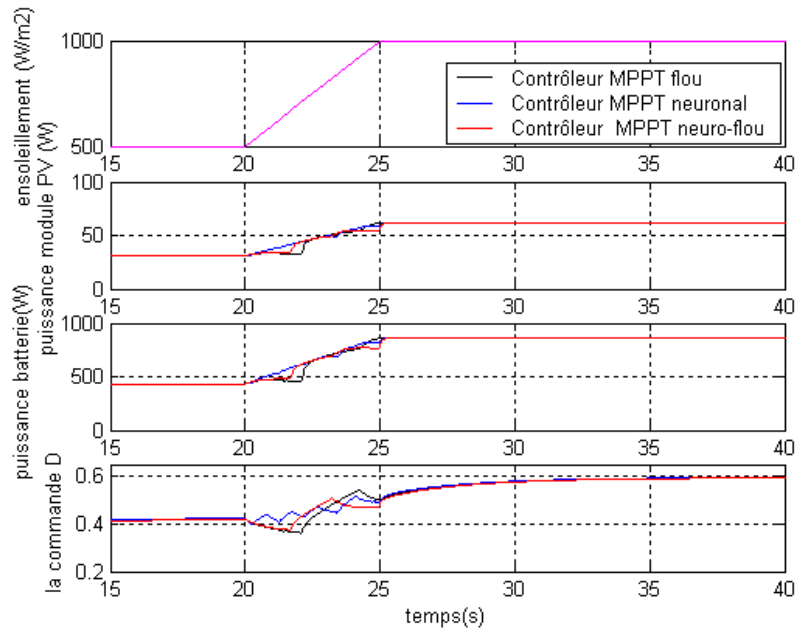


Figure IV.26 : Variation de la puissance du module, la puissance de la batterie et de la commande D pour une diminution rapide de l'ensoleillement de 500 à 1000 W/m² en 5 secondes avec une température constante de 25°C.

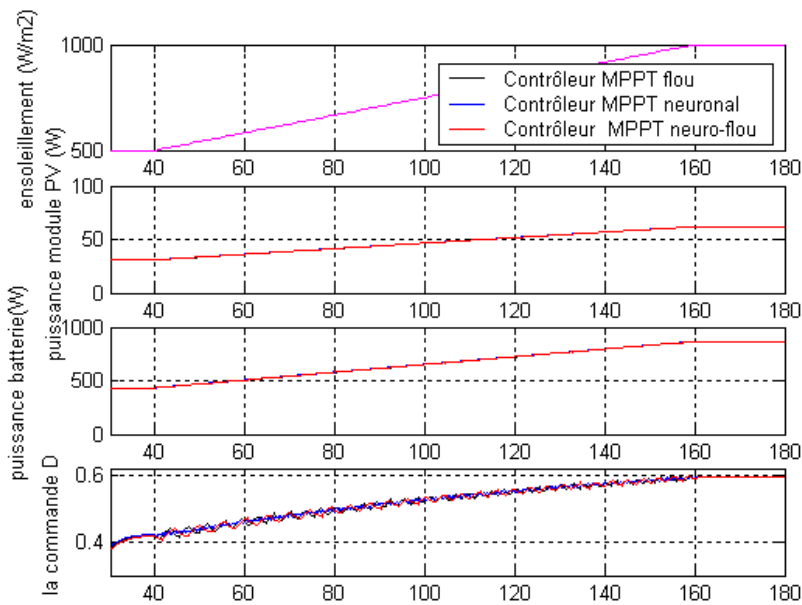


Figure IV.27 : Variation de la puissance du module, la puissance de la batterie et de la commande D pour une augmentation lente de l'ensoleillement de 500 à 1000 W/m² en 120 secondes avec une température constante de 25°C.

IV.4.3. Simulation sous des conditions variables de température

❖ Augmentation rapide de température (20°C à 45°C)

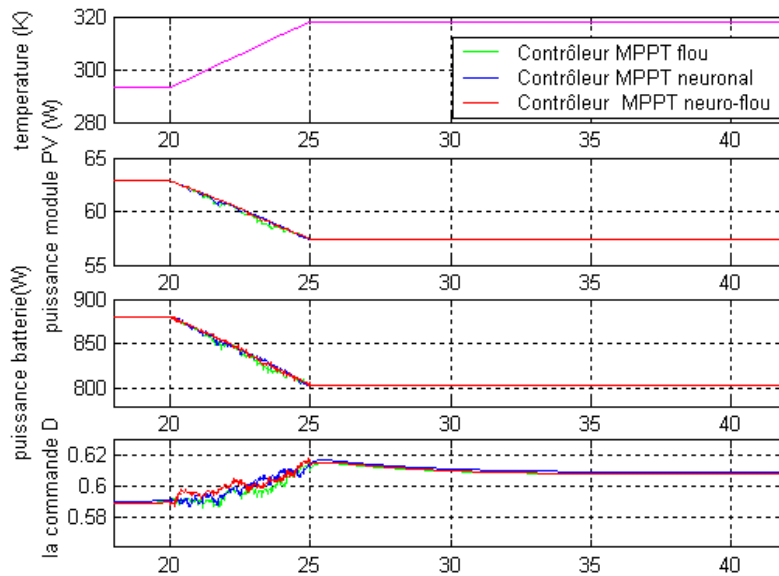


Figure IV.28 : Variation de la puissance du module, la puissance de la batterie et de la commande D pour une augmentation rapide de la température de 20°C à 45°C en 5 secondes avec un ensoleillement de 1000W/m².

❖ Diminution rapide de la température (45°C à 20°C)

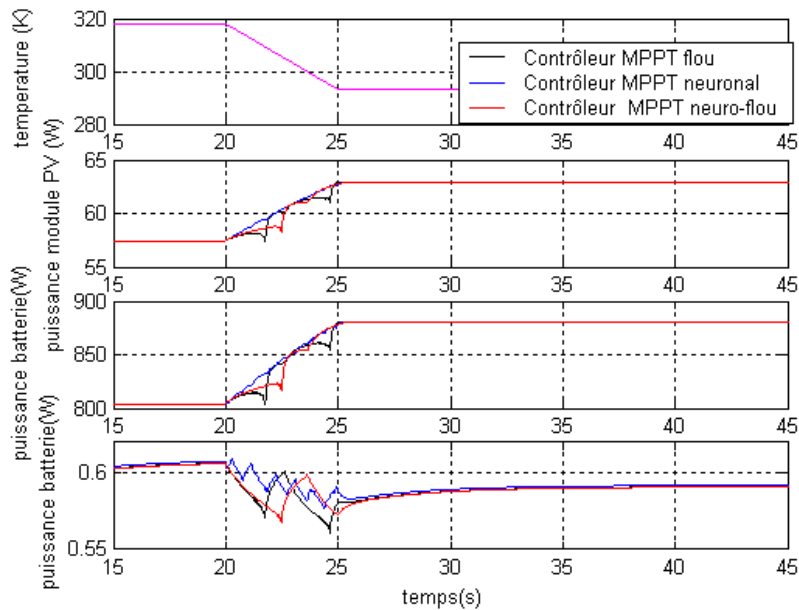


Figure IV.29 : Variation de la puissance du module, la puissance de la batterie et de la commande D pour une diminution rapide de la température de 45°C à 20°C et un ensoleillement de 1000W/m².

❖ Diminution lente de la température (45°C à 20°C)

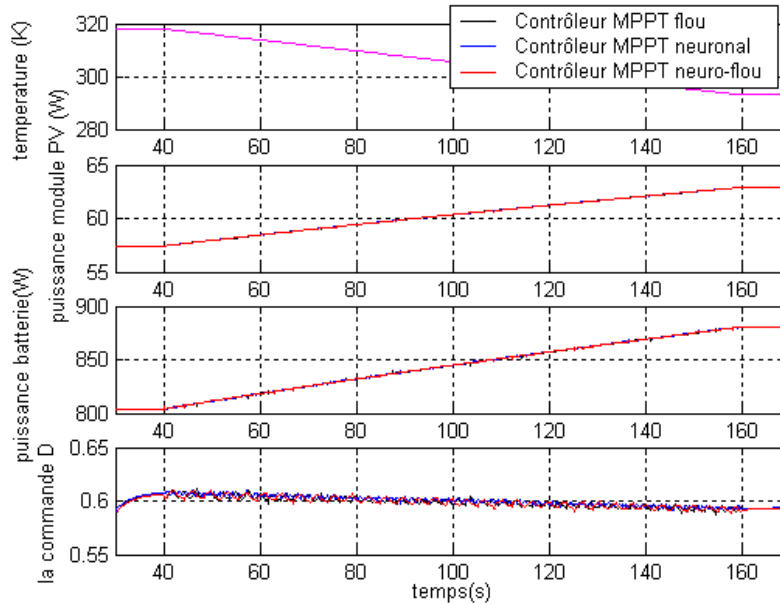


Figure IV.30 : Variation de la puissance du module, la puissance de la batterie et de la commande D pour une diminution lente de la température de 45°C à 20°C en 120 secondes avec un ensoleillement de 1000W/m².

IV.4. Comparaison des trois approches

Les résultats de simulation obtenus, mettent clairement en évidence les capacités d'approximations et de généralisation aux cas non appris. En effet, d'après les réponses obtenues par l'approche neuro-floue, nous pouvons constater que les performances restent similaires que ceux des deux techniques floue et neuronales et cela dans les conditions normales de fonctionnement ; non seulement pour la poursuite du point de puissance maximum du système photovoltaïque mais aussi du côté temps de réponse. Par ailleurs, d'après les tests des figures (25,26,28,29), nous remarquons que la commande établie par les trois techniques effectue une poursuite parfaite du point de puissance max par contre le contrôleur neuro-flou adopté s'avère plus performant du point de vue stabilité ce qui était prévisible.

IV.5 .Conclusion

Dans ce chapitre, nous avons présenté une stratégie d'hybridation entre deux techniques : les réseaux de neurones, et la logique floue en vue d'élaborer un système de commande et de poursuite du point de puissance maximale afin d'extraire le maximum de puissance.

La logique floue, avec ses capacités de modélisation de connaissances vagues et incertaines, a permis de surmonter cette difficulté liée à la non-interprétabilité des RN

grâce à la modélisation des décisions prises par l'expert à l'aide de règles linguistiques. Néanmoins, l'élaboration d'un système flou demande beaucoup d'efforts dans l'ajustement des paramètres. La logique floue a été utilisée également avec les RN, les deux paradigmes ayant été combinés dans un système dit Réseau Neuro-Flou (RNF). Le RNF implémente le système flou par une architecture parallèle, et permet par rapport à un RN, de modéliser les règles linguistiques de décision et d'évaluer l'inférence floue par l'utilisation des neurones spécialisés (neurones flous). Il permet de réduire l'effort fourni lors de l'élaboration du système en utilisant l'apprentissage neuronal pour évaluer les paramètres du système flou.

La poursuite du point de puissance maximale par ANFIS montre l'efficacité de cette dernière. L'utilisation de l'approche neuro-flou (ANFIS) permet une génération automatique des règles floues.

Conclusion Générale

La conversion photovoltaïque de l'énergie solaire est une des alternatives qui, depuis l'avènement des grands programmes spatiaux, a montré sa grande souplesse et son aptitude à fonctionner en milieu hostile. Il s'agit d'une solution intéressante aux moyens de productions conventionnels. Cependant, l'énergie délivrée par ces générateurs est directement dépendante des conditions atmosphériques et est relativement coûteuse, en raison du prix toujours élevé de ces générateurs. Ses principaux atouts restent sa grande autonomie de fonctionnement possible, une durée de vie des générateurs garantie supérieure à vingt ans et une absence de pollution lors de la production d'énergie.

Pour assurer le fonctionnement d'un générateur photovoltaïque à son point de puissance maximale, des contrôleurs MPPT sont souvent utilisés. Ces contrôleurs sont destinés à la poursuite PPM et à minimiser ainsi l'erreur entre la puissance de fonctionnement et la puissance maximale de référence qui est variable en fonction de la charge et des conditions climatiques.

Dans le même but, plusieurs techniques de contrôle MPPT ont été introduites dernièrement à savoir : la logique floue, les réseaux de neurones...etc.

Dans ce mémoire nous avons présenté une étude sur l'introduction des réseaux de neurones à couches non récurrentes pour la poursuite du point de puissance maximale.

Après avoir introduit de manière générale les réseaux de neurones, nous avons présenté les divers types de réseaux, leur application dans différents domaines, ainsi que leur architecture et leur modélisation. Puis, à l'aide du logiciel Matlab, nous avons procédé à des simulations pour montrer l'efficacité des réseaux de neurones particulièrement dans le domaine du contrôle.

Afin de bien utiliser les réseaux de neurones, on doit aussi connaître leurs limitations. Ils se comportent comme de véritables boîtes noires et il n'existe pas une forme explicite expliquant et analysant la relation entre les entrées et les sorties.

Cela cause des difficultés d'interprétation des résultats obtenus par les réseaux. En plus il n'y a pas de méthode définie pour déterminer la meilleure structure d'un réseau de neurone. La méthode la plus utilisée est de réaliser des essais, puis de les comparer avec l'erreur obtenue. De cette manière on choisit la structure de réseau la plus performante.

Puis et afin d'améliorer les réponses et voir l'effet d'hybridation nous avons ainsi conçu un contrôleur ANFIS qui a permis une génération automatique des règles floues.

Ces deux techniques de commande MPPT (neuronal et neuro-floue) proposées ont été testées pour montrer leurs efficacités. Les résultats de simulation de l'ensemble du système photovoltaïque montrent que l'approche neuro-floue a donné les meilleures performances du point de vue stabilité mais les approches neuronale et floue ont montré aussi qu'ils ont de bonnes performances.

Selon les résultats, on peut conclure que le système neuro-floue fait apparaître un bon compromis entre la caractérisation et l'efficacité des calculs. Sa robustesse, sa rapidité et la précision de ses sorties lui permettent de donner des décisions correctes et d'éviter les cas d'indécisions, les réseaux de neurones avec leurs capacités à s'adapter à des situations inconnues par le biais de l'apprentissage, et la logique floue avec sa capacité de modélisation de connaissances imprécises et gestion de l'incertitude ont été combinés dans un système appelé réseau neuro-floue.

Dans la conception des contrôleurs neuronal et neuro-flous, la base de données servant à la conception des contrôleurs est primordiale à la réussite de la conception.

Ce travail ne recherche nullement à condamner les méthodes classiques. L'approche présentée vient avant tout enrichir la grande famille des méthodes de poursuite du point de puissance maximale.

Les résultats obtenus lors de notre travail encouragent la poursuite des recherches dans ce sens. Nous pouvons également envisager de les améliorer. Il serait également intéressant de voir l'effet de l'utilisation d'autres architectures hybrides ainsi que d'autres algorithmes afin de raffiner les réglages de paramètres et les biais.

On proposera de faire une poursuite de point de puissance maximale en incluant d'autres paramètres dans notre modèle de réseaux de neurones, il existe aussi une possibilité de poursuivre le MPPT à l'aide d'autres types de réseaux de neurone : les réseaux récurrents.

Il serait également très intéressant d'implémenter ses approches (la réalisation pratique).

Annexe

Paramètres d'apprentissage

Tous les modèles de réseaux de neurones requièrent un apprentissage. Plusieurs types d'apprentissages peuvent être adaptés à un même type de réseau de neurones. Les critères de choix sont souvent la rapidité de convergence ou les performances de généralisation. Le critère d'arrêt de l'apprentissage est souvent calculé à partir d'une fonction de coût, caractérisant l'écart entre les valeurs de sortie obtenues et les valeurs de références (réponses souhaitées pour chaque exemple présenté).

Coefficient d'apprentissage η :

Ce paramètre détermine la vitesse de convergence. Si la valeur de démarrage de η est grande, alors on aura un apprentissage très rapide mais au prix de la création d'oscillations dans l'erreur totale moyenne qui empêcheront l'algorithme de converger vers le minimum désiré. Le réseau devient instable. Dans la plupart de cas, si la fonction d'erreur possède plusieurs minimums locaux, le réseau subira un blocage dans l'un d'eux. Toutes ces conditions nous obligent à commencer l'apprentissage avec une petite valeur de η , si on veut atteindre un minimum global, même si l'apprentissage est long.

Momentum α :

Ce paramètre permet l'introduction de l'ancien poids dans le calcul de sa nouvelle valeur. Le momentum empêche l'algorithme de rester bloqué dans un minimum local et il augmente encore la vitesse de convergence. La valeur de α est toujours inférieure à 1 (valeur typique $\alpha=0.9$).

Seuil de tolérance:

Ce paramètre critique détermine la précision dans la réponse du réseau de neurones.

Nombre de neurones cachés:

Le nombre de neurones dans la couche cachée détermine la structure de notre réseau. Un grand nombre de neurones est nécessaire pour modéliser une relation complexe (relation entrée-sortie). Mais dans certains cas, trop de neurones entraîne un sur apprentissage du réseau, et ce réseau tentera de mémoriser des informations au lieu de généraliser.

Nombre de couches cachées:

Il est démontré théoriquement qu'un réseau neuronal multicouche avec au moins une seule couche cachée peut modéliser arbitrairement une relation non linéaire complexe.

Algorithmes d'apprentissage utilisés**❖ Algorithme de rétro propagation du gradient**

1. Initialisation des poids à des valeurs aléatoires de faible grandeur,
2. Sélection aléatoire d'un exemple dans la base d'apprentissage $\langle x, d \rangle$,
3. Propager l'entrée à travers le réseau,
4. Appliquer l'entrée x sur le réseau et calculer la sortie o de chaque couche,
5. Pour chaque sortie du réseau calculer le terme d'erreur:

$$\delta_k = (d_k - o_k) \sigma'(x_i)$$

6. Pour chaque couche cachée, calculer le terme d'erreur:

$$\delta_k = \sum_{k=1}^N w_{ki} \cdot \sigma'(x_i)$$

7. Mettre à jour chaque poids synaptique du réseau:

$$w_{ij} = w_{ij} + \Delta w_{ij}$$

Où :

$$\Delta w_{ij} = \eta \cdot \delta_j \cdot x_{ij}$$

8. Retourner en 2. Tant que l'erreur est trop grande.

❖ Méthode de Levenberg Marquardt (LM)

C'est la méthode la plus utilisée, cette méthode considère la surface d'erreur quadratique, en se basant sur la fonction d'énergie qui a la forme :

$$E(k+1) = E(k) + \dot{E}(k) \cdot \Delta W(k) + \frac{1}{2} (\Delta W(k)) \cdot \ddot{E}(k) \cdot \Delta W(k) \quad (1)$$

En posant pour chaque neurone i d'une couche L le vecteur de poids $(W)_i^l$ lui parvenant des neurones de la couche qui le précède. Minimiser la fonction d'énergie revient à résoudre l'équation :

$$\frac{\partial E(k+1)}{\partial \Delta W(k)} = 0 \quad (2)$$

Avec $\Delta W(k)$ est le vecteur global des taux de tous les vecteurs de poids $(W)_i^l$.

Soit J_i^l les vecteurs des dérivées de l'erreur à la sortie du réseau par rapport à ces vecteurs poids $(W)_i^l$ pour $l=1, \dots, L$, et $i=1, \dots, n_l$ où n_l est le nombre de neurones dans la dernière couche et L le nombre de couches cachées.

$$J_i^l = \sum_{p=1}^k \sum_{i=1}^{n_l} (d_i^p - y_i^{(L)}) \left[-\frac{\partial y_i^p}{\partial w_i^l} \right] \quad (3)$$

k représente le nombre d'exemples d'entraînement. On définit J le vecteur rassemblant les vecteurs J_i^l qui est le jacobien de l'erreur par rapport aux vecteurs poids. Afin de trouver l'optimum de (2), on obtient :

$$H \Delta W = -J \quad (4)$$

H représente la matrice Hessienne (ou le Hessien)

L'équation (4) représente un système d'équations linéaires dont la résolution peut être fait par ma méthode de Gauss Newton.

La méthode de Levenberg-Marquardt remplace le calcul de Hessien par une approximation numérique donnée par :

$$H = FF^T + \lambda \Omega \quad (5)$$

Où λ Est un coefficient positif

Ω : Une matrice choisie (le choix d'une matrice identité diagonale peut bien être fait).

Tout comme les méthodes quasi-Newtoniennes, l'algorithme d'apprentissage de Levenberg Marquardt était conçu pour approximer le calcul des deuxièmes dérivées sans devoir calculer la matrice Hessienne, en utilisant l'approximation suivante

$$H = J_C^T J_C$$

et le gradient est calculé par :

$$G = J_c^T \cdot \mathcal{E}$$

Où J_c est la matrice Jacobinne qui contient les premières dérivées de l'erreur par rapport aux poids. Le Jacobinien peut être calculé par rétro propagation (méthode du gradient) qui est beaucoup moins complexe que le calcul de la matrice Hessienne

L'algorithme de Levenberg Marquardt emploie cette approximation de la matrice Hessienne, pour calculer les nouveaux poids du réseau de neurones de la couche L, en modifiant l'algorithme de Newton comme suit :

$$W_{ij}^l(n+1) = W_{ij}^l(n) - [H(n) + \eta_u I]^{-1} G(n)$$

I: est une matrice d'identité

η_u : est le facteur d'apprentissage

H (n) : est la matrice Hessienne

Quand η_u tend vers zéro, on trouve la méthode de Newton, en utilisant l'approximation de la matrice Hessienne et quand η_u est suffisamment grand, on obtient l'algorithme de la méthode de descente de gradient avec un pas d'essai relativement faible.

Le choix de η_u est cependant était délicat ; on peut aboutir à des effets inverses, des oscillations ou ralentissement de la convergence .Les performances apportées par ce terme ne sont pas toujours convaincantes. Le mieux est de choisir un facteur d'apprentissage adaptatifs μ_n est augmentée après chaque itération qui provoque une réduction de l'erreur, et diminué seulement quand un pas d'essai augmente l'erreur ce qui fait que l'erreur est réduite à chaque itération de l'algorithme.

L'algorithme de Levenberg Marquardt paraît être la méthode la plus rapide, quand il est utilisé pour l'apprentissage des réseaux de neuronaux de taille modérée de type feedforward comprenant plusieurs centaines de poids.

1. Circuits équivalents et modèles mathématiques

1.1. Circuit équivalent et modèle mathématique d'une cellule photovoltaïque

Le schéma du circuit équivalent idéal simplifié d'une cellule photovoltaïque est représenté sur la figure 1

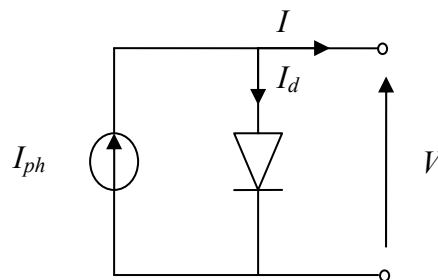


Figure 1 : Schéma du circuit équivalent idéal simple d'une cellule photovoltaïque.

L'application de la loi de Kirchhoff et de l'équation de diode exponentielle (1) mène au modèle mathématique simple pour une cellule photovoltaïque (2).

$$I_d = I_s \left[e^{\frac{qV}{2kT}} - 1 \right], \quad (1)$$

$$I = I_{ph} - I_s \left[e^{\frac{qV}{2kT}} - 1 \right]. \quad (2)$$

I et V sont respectivement le courant et la tension de sortie de la cellule. I_{ph} est le photo-courant produit et I_s est le courant de saturation inverse de la diode. En plus, le modèle dépend de la température T , de la constante de charge élémentaire q ($1,602 \cdot 10^{-19} \text{ C}$) et de la constante de Boltzmann k ($1,380 \cdot 10^{-23} \text{ J/K}$).

Les mesures réelles sur de vraies cellules dans des conditions de fonctionnement diverses montrent cependant le besoin de modèles plus sophistiqués. En particulier, la résistance interne du dispositif doit être prise en compte. Ceci mène au « modèle de deux diode », largement répandu, représenté sur la figure 3.

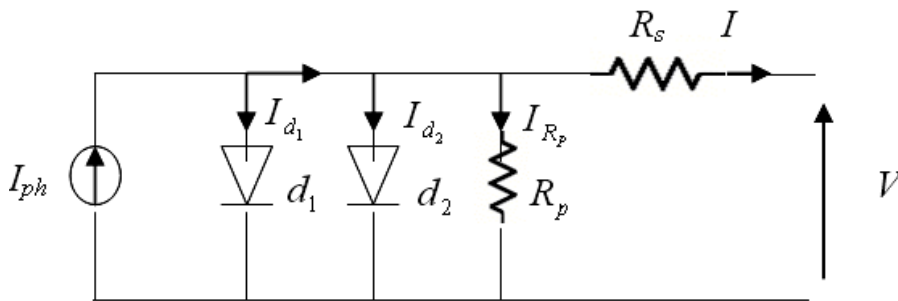


Figure 3 : Modèle équivalent à deux diodes d'une cellule photovoltaïque.

La figure 3 est une représentation du modèle mathématique pour la caractéristique courant-tension qui est donnée par :

$$I = I_{ph} - I_{s1} \left[e^{\frac{q(V+I.R_s)}{n_1 k T}} - 1 \right] - I_{s2} \left[e^{\frac{q(V+I.R_s)}{n_2 k T}} - 1 \right] - \frac{V + I R_s}{R_p}. \quad (3)$$

I_{s1} et I_{s2} sont les courants de saturation des diodes, n_1 et n_2 les facteurs de pureté de la diode, R_s et R_p sont respectivement la résistance série et la résistance parallèle, et T est la température absolue en Kelvin. L'équation contient également la charge élémentaire constante q ($1,602 \cdot 10^{-19} \text{ C}$) et la constante de Boltzmann k ($1,380 \cdot 10^{-23} \text{ J/K}$). Le photo-courant $I_{ph, \max}$ est atteint à une insolation maximum, souvent on a ($I_{ph} = S \cdot I_{ph, \max}$) avec S : pourcentage d'insolation.

Il est évident de l'équation (3), que la caractéristique courant-tension dépend fortement de l'insolation et de la température. La dépendance de la température est encore amplifiée

par les propriétés du photo-courant I_{ph} et les courants de saturation inverse des diodes qui sont donnés par :

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

$$I_{s1} = K_1 T^3 e^{-\frac{E_g}{kT}}, \quad (5)$$

$$I_{s2} = K_2 T^{\frac{5}{2}} e^{-\frac{E_g}{kT}}, \quad (6)$$

Où E_g est la bande d'énergie du semi-conducteur et

$$K_1 = 1,2 \text{ A/cm}^2 \cdot \text{K}^3 \quad (7)$$

$$K_2 = 2,9 \cdot 10^5 \text{ A/cm}^2 \cdot \text{K}^{5/2}. \quad (8)$$

1.2. Circuit équivalent et modèle mathématique d'un module solaire

Dans les systèmes d'énergie photovoltaïque, des cellules sont combinées dans des rangées de pile solaire reliant un certain nombre de cellules en série. La considération du modèle de circuit équivalent (la figure 3) mène à l'équation pour une rangée photovoltaïque de cellules, avec z cellules photovoltaïques raccordées en série.

$$I = I_{ph} - I_{s1} \left[e^{\frac{q(V+IzR_s)}{zn_1kT}} - 1 \right] - I_{s2} \left[e^{\frac{q(V+IzR_s)}{zn_2kT}} - 1 \right] - \frac{V + IzR_s}{zR_p}. \quad (9)$$

Ces modules alors peuvent être encore arrangés en série ou en parallèle pour réaliser la tension et les valeurs de courant désirés pour le système.

1.3 .Modèle mathématique des batteries au plomb

Dans la littérature des auteurs ont utilisé le modèle représenté sur la figure 4. Le modèle décrit le comportement primaire d'une batterie correctement, mais n'explique pas les propriétés lentement changeantes d'une batterie telles que l'augmentation et la diminution de la tension de fonctionnement provoquée par le processus de la charge et de la décharge.

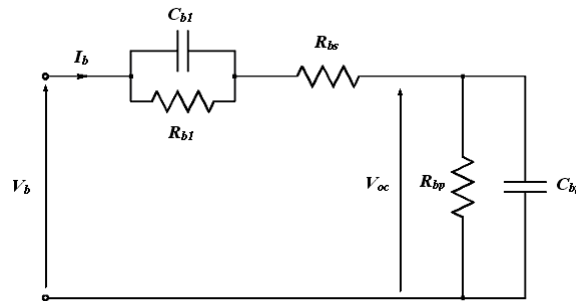


Figure 4 : Modèle équivalent de la batterie.

Ce modèle inclut les composants équivalents pour toutes les caractéristiques de fonctionnement principales d'une batterie plomb-acide:

La capacité électrochimique de la batterie est représentée par le condensateur C_{bp} et sa valeur est donnée par l'expression générale de l'énergie :

$$E_c = \frac{1}{2} C V_c^2, \quad (10)$$

Avec : C la capacité et V_c la tension du condensateur.

A la différence d'un condensateur, la tension d'une batterie ne sera pas égale à zéro à son plus bas état de charge. C'est équivalent à un condensateur ayant un niveau minimum de charge égal à l'énergie $E_{c_{\min}}$. La capacité entièrement chargée de la batterie est représentée par un niveau maximum de charge $E_{c_{\max}}$. Ceci est indiqué par l'équation suivante :

$$E_b = E_{c_{\max}} - E_{c_{\min}} = \frac{1}{2} C V_{\max}^2 - \frac{1}{2} C V_{\min}^2 = \frac{1}{2} C_{bp} (V_{\max}^2 - V_{\min}^2) \quad (11)$$

L'énergie E_b est donnée par le constructeur de la batterie directement en kilowatt par heure (kWh). Les tensions V_{\max} et V_{\min} sont la tension maximale et la tension minimale de la batterie en circuit ouvert respectivement. D'après (11) on a :

$$C_{bp} = \frac{2E_b}{V_{\max}^2 - V_{\min}^2} \quad (12)$$

C_{bp} est la capacité de la charge de la batterie.

La résistance interne de la batterie est représentée par les deux résistances R_{bs} et R_{b1} en série. La résistance en bloc d'électrolyte et de plaque est représentée par la résistance R_{bs} tandis que la résistance R_{b1} représente la diffusion d'électrolyte. Ceci représente la tension du circuit ouvert de la batterie dès qu'une charge sera reliée. De même on peut observer un saut soudain de tension avec l'application d'un courant de remplissage. En utilisant la notation indiquée sur la figure 4 on peut exprimer la tension de batterie V_b en fonction de la tension de batterie en circuit ouvert et les autres composants R_{b1} , R_{bs} et C_{b1} avec la constante de temps $\tau = R_{b1} C_{b1}$.

$$V_b = V_{oc} + R_{b1} \left(1 - e^{-t/\tau}\right) i_b + R_{bs} i_b \quad (13)$$

Une autre caractéristique très importante d'une batterie est la décharge spontanée, représentée par la résistance R_{bp} parallèle avec condensateur principal C_{bp} . Elle est provoquée par électrolyse de l'eau aux tensions élevées et par la fuite lente à travers les bornes de batterie aux basses tensions.

Le circuit sur la figure 4 décrit les caractéristiques d'une batterie plomb-acide d'une manière complète mais pourtant très simplifiée:

Ce circuit exprime l'impédance équivalente d'entrée d'une batterie plomb-acide par :

$$Z(s) = R_{bs} + (R_{b1} \parallel C_{b1}) + (R_{bp} \parallel C_{bp}) = R_{bs} + \frac{R_{b1}}{R_{b1} \cdot C_{b1} s + 1} + \frac{R_{bp}}{R_{bp} \cdot C_{bp} \cdot s + 1} \quad (14)$$

$$Z(s) = \frac{a_2 s^2 + a_1 s + a_0}{b_2 s^2 + b_1 s + b_0} \quad (15)$$

Les coefficients a_i et b_j sont employés pour représenter les différents composants:

$$\begin{aligned} a_2 &= R_{bs} R_{b1} R_{bp} C_{b1} C_{bp}, \\ a_1 &= R_{bs} R_{b1} C_{b1} + R_{bs} R_{bp} C_{bp} + R_{b1} R_{bp} C_{bp} + R_{bp} R_{b1} C_{b1}, \\ a_0 &= R_{bs} + R_{b1} + R_{bp}, \\ b_2 &= R_{b1} R_{bp} C_{b1} C_{bp}, \\ b_1 &= R_{b1} C_{b1} + R_{bp} C_{bp}, \\ b_0 &= 1. \end{aligned} \quad (16)$$

2. Convertisseur Continu-Continu (DC/DC) (Hacheurs)

Les hacheurs sont les convertisseurs statiques continu-continu permettant de générer une source de tension continue variable à partir d'une source de tension continue fixe. Le hacheur se compose de condensateurs, d'inductances et de commutateurs. Tous ces dispositifs dans le cas idéal ne consomment pas de puissance, c'est la raison pour laquelle les hacheurs ont de bons rendements. Généralement le commutateur est un transistor MOSFET qui est un dispositif semi-conducteur en mode (bloqué-saturé).

Si le dispositif semi-conducteur est bloqué, son courant est zéro d'où sa dissipation de puissance est nulle. Si le dispositif est dans l'état saturé, la chute de tension à ses bornes sera presque zéro et par conséquent la puissance perdue sera très petite.

Comme le montre la figure 5 pendant le fonctionnement du hacheur, on ferme le commutateur avec un temps de fermeture égale à $d.T_s$, et on l'ouvre dans un temps d'ouverture = $(1-d)T_s$, où:

- T_s est la période de commutation qui est égale à $1/f_s$.
- d le rapport cyclique du commutateur ($d \in [0,1]$).

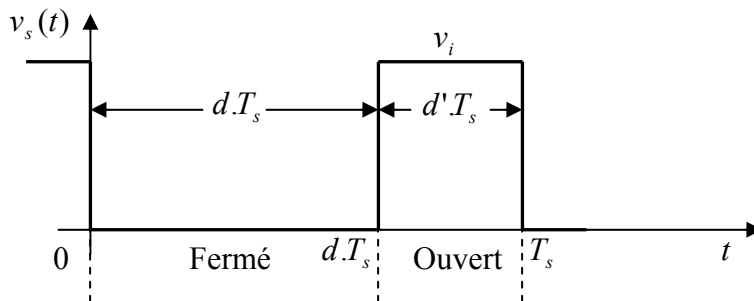


Figure 5 : Tension $v_s(t)$ idéale du commutateur, rapport cyclique d et période de commutation T_s .

2. 1. Hacheur survolteur (Boost converter)

Le convertisseur boost ou hacheur parallèle connu par le nom d'élévateur de tension, est donné par la figure (6). En premier temps dT_s , figure (7), le Transistor est fermer, le courant dans l'inductance croit progressivement, au fur et à mesure elle emmagasine de l'énergie, jusqu'à la fin de la première période, le transistor s'ouvre et l'inductance L s'oppose à la diminution du courant I_L , ainsi elle génère une tension qui s'ajoute à la tension de source, qui s'applique sur la charge à travers la diode.

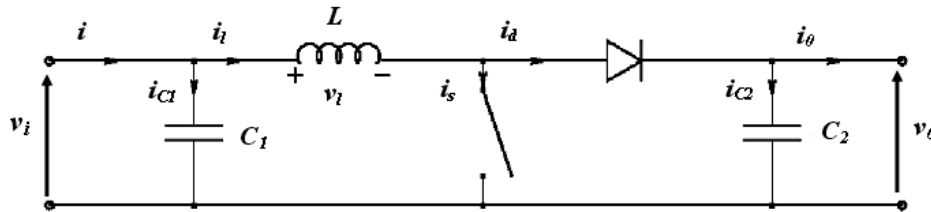


Figure 6 : Circuit idéal du hacheur survolteur.

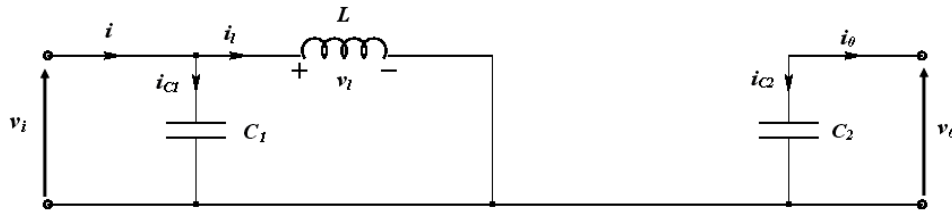


Figure 7 : Circuit équivalent d'un hacheur survolteur au temps de fermeture dT_s de commutateur.

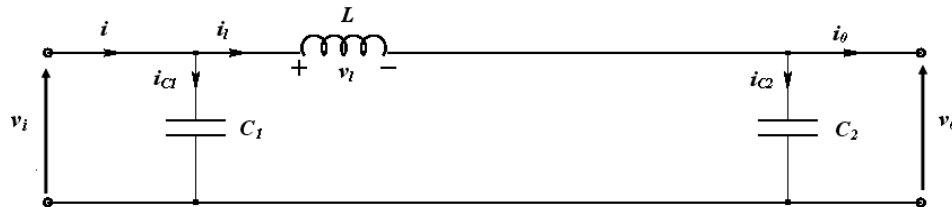


Figure 8: Circuit équivalent d'un hacheur survolteur au temps d'ouverture de commutateur dT_s .

I.2.2.2.1 Modèle mathématique du circuit équivalent

La figure 6 représente le modèle équivalent de l'hacheur survolteur.

1- Quand le commutateur est fermé (figure 7) (c.à.d $t \in [0, dT_s]$), on a les équations suivantes :

$$\begin{aligned}
 i_{C1}(t) &= C_1 \frac{dv_i(t)}{dt} = i(t) - i_l(t), \\
 i_{C2}(t) &= C_2 \frac{dV_0(t)}{dt} = -i_0(t), \\
 v_l(t) &= L \frac{di_l(t)}{dt} = v_i(t).
 \end{aligned} \tag{17}$$

2- Quand le commutateur est ouvert ($t \in [dT_s, T_s]$), (Figure 8). On a les équations suivantes :

$$\begin{aligned}
 i_{C1}(t) &= C_1 \frac{dv_i(t)}{dt} = i(t) - i_l(t), \\
 i_{C2}(t) &= C_2 \frac{dV_0(t)}{dt} = i_l(t) - i_0(t), \\
 v_l(t) &= L \frac{di_l(t)}{dt} = v_i(t) - v_0(t).
 \end{aligned} \tag{18}$$

Détermination des ondulations des courants et des tensions

La pente de courant I_l et des tensions v_{c1} et v_{c2} pendant la première période de fonctionnement est donnée par :

$$\left\{ \begin{aligned}
 \frac{dv_{c1}}{dt} &\approx \frac{i_{C1}}{C_1} = \frac{I - I_l}{C_1} \\
 \frac{dv_{c2}}{dt} &\approx \frac{i_{C2}}{C_2} = \frac{-I_0}{C_2} \\
 \frac{di_l}{dt} &\approx \frac{v_l}{L} = \frac{V_i}{L}
 \end{aligned} \right. \tag{19}$$

Les valeurs crête à crête des courants et des tensions sont :

$$\left\{ \begin{aligned}
 v_{icc} &= 2\Delta v_i = \frac{I - I_l}{C_1} dT_s \\
 v_{0cc} &= 2\Delta v_0 = \frac{-I_0}{C_2} dT_s \\
 i_{lcc} &= 2\Delta i_l = \frac{V_l}{L} dT_s
 \end{aligned} \right. \tag{20}$$

Les valeurs des composants à choisir pour des ondulations données sont :

$$\left\{ \begin{aligned}
 C_1 &= \frac{I - I_l}{2\Delta v_i} dT_s \\
 C_2 &= \frac{-I_0}{2\Delta v_0} dT_s \\
 L &= \frac{V_i}{2\Delta i_l} dT_s
 \end{aligned} \right. \tag{21}$$

Bibliographie

- [1] Y.Pankow, *Etude de l'intégration de la production décentralisée dans un réseau basse tension. Application au générateur photovoltaïque*, Thèse de doctorat, l'Ecole Nationale Supérieure d'Arts et Métiers, décembre 2004.
- [2] R.Andoulsi, *Etude d'une classe de systèmes photovoltaïque par une approche bond graph. modélisation, analyse et commande*, Thèse de doctorat, Ecole centrale de LILLE, et université des sciences et technologies de LILLE, Décembre 2001.
- [3] O.Gergaud, *Modélisation énergétique et optimisation économique d'un système de production éolien et photovoltaïque couplé au réseau et associé à un accumulateur*, Thèse de doctorat, École normale supérieure de Cachan, décembre 2002.
- [4] M.Angel Cid Pastor, *Conception et réalisation de modules photovoltaïques électroniques*, Thèse de doctorat, Institut National des Sciences Appliquées de Toulouse, septembre 2006.
- [5] A. Learreta, *Réalisation de commandes MPPT numériques*, Rapport de Stage Projet Fin d'Etudes, Septembre 2006.
- [6] C. Alonso, *Contribution à l'optimisation, la gestion et le traitement de l'énergie*, Mémoire de l'habilitation à diriger les recherches Université Paul Sabatier – ToulouseIII, Décembre 2003.
- [7] S.Hadji, F.Krim, *Développement et implémentation d'un algorithme de poursuite du point de puissance maximale d'un système photovoltaïque à base d'algorithmes génétiques*, colloque national sur l'énergie solaire Proceeding, Bejaia,Algérie, 2006, pp 144-150.
- [8] Abd El-Shafy A. Nafeha, Faten H. Fahmya, Essam M. Abou El-Zahab, *Evaluation of a proper controller performance for maximum-power point tracking of a stand-alone PV system*, Solar Energy Materials & Solar Cells 75 (2003) 723–728.
- [9] H. Tarik Duru, *A maximum power tracking algorithm based on $I_{mpp} = f(P_{max})$ function for matching passive and active loads to a photovoltaic generator*, Solar Energy 80 (2006) 812–822.
- [10] M.Veerachary, T.Senju, and K.Uezato, *Neural-Network-Based Maximum-Power-Point Tracking of Coupled-Inductor Interleaved-Boost-Converter-Supplied PV System Using Fuzzy Controller*, IEEE Transactions on industrial electronics, Vol. 50, N°. 4, August 2003.
- [11] Tishan Eram, Jonathan W. Kimball, Philip T. Krein, Fellow, Patrick L. Chapman, and Pallab Midya, *Dynamic Maximum Power Point Tracking of Photovoltaic Arrays Using Ripple Correlation Control*, IEEE Transactions on power electronics, Vol. 21, N°. 5, September 2006.
- [12] A.D. Karlis, T.L. Kottas, Y.S. Boutalis, *A novel maximum power point tracking method for PV systems using fuzzy cognitive networks (FCN)*, Electric Power Systems Research 77 (2007) 315–327.

- [13] R. Akkaya, A.A. Kulaksız, O. Aydog̃du, *DSP implementation of a PV system with GA-MLP-NN based MPPT controller supplying BLDC motor drive*, Energy Conversion and Management 48 (2007) 210–218.
- [14] Leonard G. Leslie, *Design and analysis of a grid connected photovoltaic generation system with active filtering function*, Master of Science in Electrical Engineering, March, 2003.
- [15] K. Benlarbi a, L. Mokrani b, M.S. Nait-Said, *A fuzzy global efficiency optimization of a photovoltaic water pumping system*, Solar Energy 77 (2004) 203–216.
- [16] N. Patcharaprakiti, S. Premrudeepreechacharn, Y. Sriuthaisiriwong, *Maximum power point tracking using adaptive fuzzy logic control for grid-connected photovoltaic system*, Renewable Energy 30 (2005) 1771–1788.
- [17] G.F. Tchoketch, *Commande des hacheurs MPPT par la logique floue*, Magister, ENP, Algérie 2006.
- [18] O. Thameur, *Application des algorithmes génétiques dans la commande des hacheurs MPPT*, Magister, ENP, Algérie 2006.
- [19] F. Belhachat, *Système de classification des chromosomes humains par réseaux de Hopfield*, PFE, USTHB, 2004.
- [20] M. Nezar, *Diagnostic des associations convertisseurs statiques machines asynchrones en utilisant les techniques de l'intelligence artificielle*, Thèse de doctorat, Université de Batna, Juin 2006.
- [21] A.B.G. Bahgat, N.H. Helwa, G.E. Ahmad, E.T. El Shenawy, *Maximum power point tracking controller for PV systems using neural networks*, Renewable Energy 30 (2005) 1257–1268.
- [22] D. Ould Abdeslam, *Techniques neuromimétiques pour la commande dans les systèmes électriques : application au filtrage actif parallèle dans les réseaux électriques basse tension*, Thèse de doctorat, Université de Haute-Alsace, décembre 2005.
- [23] E. Gauthier, *Gestion d'une flotte de véhicules autonomes à l'intérieur d'un parking haute densité, Rapport d'étude*, Institut National Polytechnique de Grenoble DEA Informatique Option Robotique Vision Image, juin 1995.
- [24] E. Gauthier, *Utilisation des réseaux de neurones artificiels pour la commande d'un véhicule autonome*, Thèse de doctorat, Institut National Polytechnique de Grenoble, Janvier 1999.
- [25] V. Andronova, *Utilisation de données météo et des réseaux de neurones pour la prédiction de vitesses de vent*, Projet de fin d'étude Master 2^{ème} année, Université technique de Sofia - Université de Corse – Pasquale Paoli, juillet 2006.
- [26] P. Lucidarme, *Apprentissage et adaptation pour des ensembles de robots réactifs coopérants*, Thèse de doctorat, Université Montpellier II, novembre 2003.
- [27] F. Santos Osório, *Un système hybride neuro-symbolique pour l'apprentissage automatique constructif*, Thèse de doctorat, Institut National Polytechnique de Grenoble - I.N.P.G, Février 1998.
- [28] L. Barazane, *Application des systèmes émergents à la commande d'un moteur asynchrone*, Thèse de doctorat, ENP, 2003.

- [29] H.Ait Abdesslam, *Réalisation d'un système d'aide à la détection des aberrations chromosomiques*, Magister, ENP, Février 2005.
- [30] O.Kisi I, *Suspended sediment estimation using neuro-fuzzy and neural network approaches*, Erciyes University, 2005. Hydrological Sciences–Journal–des Sciences Hydrologiques, 50(4) August 2005.
- [31] K. Benlarbi, L. Mokrani, M.S. Nait-Said, *A fuzzy global efficiency optimization of a photovoltaic water pumping system*, Solar Energy n°77,p 203–216,2004.
- [32] A.Aissa,*Commande sans capteur de vitesse par DTC d'une machine synchrone à aimants permanents dotée d'un observateur d'ordre complets à modes glissants*,Thèse de magister,Université de Batna, Octobre 2005.
- [33]Y.Bourek, *Détermination des conditions d'ionisation caractérisant le seuil de claquage de l'air par la logique floue*, Mémoire de magister, Université de Batna, 2005.
- [34] S. Boudraa, *Analyse des gaz dissous dans les huiles des transformateurs en utilisant les techniques de l'intelligence artificielle*, Mémoire de magister, Université de Batna, 2005.
- [35] N.Fadlallah, *Contribution à l'optimisation de la synthèse du lobe de rayonnement pour une antenne intelligente. Application à la conception de réseaux à déphasage*, Thèse de doctorat, Université de Limoges, Mai 2005.
- [36]M.Ta Cao, *Commande numérique de machines asynchrones par logique floue*, PHD, Université Laval, 1997.
- [37] Y.Nakoula, *Apprentissage des modèles linguistiques flous, par jeu de règles pondérées*, Thèse de doctorat, Université de Savoie, 1997.
- [38] Jyh-Shing Roger Jang, *ANFIS: Adaptive-Network-Based Fuzzy Inference System*, IEEE Transactions on systems, Man, and Cybernetics, Vol 23, N°3, Mayijune 1993.
- [39] D.Vlachos, Y. A. Toliás, *Neuro-Fuzzy modeling in bankruptcy prediction*, Yugoslav Journal of Operations Research 13 (2003), Number 2, 165-174.
- [40] S.Chopra, R. Mitra, and V.Kumar, *a neuro fuzzy learning and its application to control system*, International journal of computational intelligence volume 3 Number 1 2006 .
- [41]R.Babuška, H.Verbruggen, *Neuro-fuzzy methods for nonlinear system identification*, Annual Reviews in Control 27 (2003) 73–85.
- [42] J. S. Jang, C. T. Sun, and E. Mizutani, “*Neuro-fuzzy and soft computing*”, Prentice Hall, 1997.
- [43] Ž.Čojbašić, V.Stephan, H.Gross, J.Wernstedt, *Intelligent control of complex combustion processes*, Facta Universitatis, Series: Mechanical Engineering Vol. 1, No 10, 2003, pp. 1393 – 1406.
- [44] S.Tao ,Y. Si-jia, C.Guang-yi, Z. Xin-jian ,*Modelling and control PEMFC using fuzzy neural networks* , Journal of Zhejiang University SCIENCE , 2005 6A(10):1084-1089.
- [45] A.Khosravi, J.Lu, *Fault Modeling for Nonlinear Systems Using ANFIS*, Proceedings of the International Multiconference on Computer Science and Information Technology, pp. 75 – 83, 2006.