

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITE DE M'SILA
FACULTE DE TECHNOLOGIE
DEPARTEMENT DE ELECTRONIQUE



MEMOIRE DE FIN D'ETUDE EN VUE DE L'OBTENTION DU DIPLOME
D'INGENIEUR D'ETAT EN GENIE ELECTRONIQUE

OPTION

Contrôle

THEME

Optimisation des systèmes non-linéaires utilisant Particle Swarm Optimization (PSO)

Proposé et dirigé par:

Mr. AOUICHE Abdelaziz

Présenté par :

-MAHROUG Abdelkarim

-MENTADJI Moussa

-DEGHNOUCHE Houssam Eddine

N° D'ordre: 2010 092149339

PROMOTION: JUIN 2010



Remerciement

À l'issue de cette fin d'étude, nous adressons nos sincères remerciements premièrement à DIEU tout puissant qui nous a donné la santé, la volonté et la patience.

Ensuite, on tient à adresser nos plus vifs remerciements à nos promoteur Mr. AOUICHE Abdelaziz. pour nous avoir encadré, suivi et encouragé.

On remercie également, tous les membres du jury de bien vouloir juger ce travail. Ainsi que tous nos enseignants du Département d'Electronique.

On n'exclue pas de ces remerciements toutes les personnes qui ont aidé de près ou de loin dans la réalisation de ce mémoire, et notamment:
Mr. BARKAT Said, Youssef. B, Farés. K , Adel. B, Amine. H

On veut également remercier nos familles et nos amis pour leur soutien moral.

Table des matières

Remerciements.....	i
Table des matières	ii
Introduction générale.....	1

Chapitre 1

1L'optimisation par essaim particulaire	5
1.1 Introduction	5
1.2 Optimisation par essaim particulaire	5
1.2.1 Les différents types de PSO	6
1.2.1.1 PSO du optimum global	6
1.2.1.1.1 Principe de déplacement d'une particule dans PSO du optimum global	6
1.2.1.2 PSO du optimum local	9
1.2.1.2.1 Principe de déplacement d'une particule dans PSO du optimum local	10
1.2.2 Les Composants de vitesse	13
1.2.3 Les paramètres du PSO.....	14
1.2.4 La comparaison entre les deux version du PSO	15
1.3 Test de l'efficacité du PSO.....	16
1.4 Les champs d'application du PSO	17
1.5 Conclusion	17

Chapitre 2

2 Les Réseaux de Neurones Artificiels	18
2.1 Introduction	18
2.2 Le modèle neurophysiologique	18
2.3 Le modèle mathématique (Neurone Artificiel)	19
2.4 Propriété fondamentale des réseaux de neurones : L'approximation Universelle	21
2.5 Structure d'interconnexion	22
2.6 Différentes topologies neuronales	22
2.6.1 Réseau multicouche (MLP)	22

2.6.2 Réseau à connexions locales	24
2.6.3 Réseau à connexions récurrentes	24
2.6.4 Réseau à connexion complète	25
2.7 Apprentissage des réseaux de neurones	25
2.7.1 Apprentissage supervisé	25
2.7.2 Apprentissage non supervisé	26
2.8 L'algorithme de rétropropagation	26
2.9 Algorithme de Levenberg-Marquardt	29
2.9.1 Avantage de La méthode Levenberg-Marquardt	32
2.10 Le réseau à fonction de base radiale	32
2.11 Comparaison entre le réseau MLP et le réseau RBF	33
2.12 Mise en œuvre d'algorithme d'apprentissage du RNA	34
2.13 Généralisation d'un réseau MLP (Validation croisée)	36
2.14 Les champs d'application des RNA	38
2.15 Les limitations d'un réseau de neurones.....	38
2.15.1 Avantages	38
2.15.2 Inconvénients	38
2.16 Conclusion	39

Chapitre 3

3 Modélisation de l'onduleur triphasé cascadié à sept niveaux	40
3.1 Introduction.....	40
3.2 Structure de l'onduleur triphasé cascadié à sept niveaux	41
3.3 Modélisation aux valeurs instantanées de l'onduleur triphasé cascadié à sept niveaux	41
3.3.1 Hypothèses.....	42
3.3.2 Commande complémentaire	42
3.3.3 Différentes configurations d'un bras de l'onduleur triphasé cascadié à sept niveaux	43
3.3.4 Mise en équations	46
3.4 Stratégie de commande de l'onduleur	48
3.4.1 Commande de l'onduleur à sept niveaux par la stratégie d'élimination d'harmoniques	48
3.4.1.1 Modulation par paliers.....	48
3.4.1.2 Modulation par élimination sélective des harmoniques	50
3.5 Résultats de simulation	51
3.6 Conclusion	52

Chapitre 4

Résultats de simulation:

Application du PSO et RNA sur de l'onduleur triphasé cascadié à sept niveaux	53
4.1 Introduction	53
4.2 Elimination d'harmoniques	54
4.2.1 Formulation du problème	54
4.2.2 Elimination d'harmoniques par PSO	55
4.2.3 Résultats de simulation	57
4.3 Commande de l'onduleur triphasé à sept niveaux par les réseaux de neurones	62
4.3.1 Architecture du réseau élaboré	63
4.3.2 Résultats de simulation	63
4.4 Conclusion	67
Conclusion générale	69
Annexe A	71
Bibliographie.....	72

Introduction générale

L'optimisation est un paradigme actuel dans chaque aspect de la vie. Elle propose constamment des manières améliorées pour résoudre les anciens et nouveaux problèmes. Dans le contexte du développement technologique et de l'innovation, l'optimisation décrit la recherche des techniques qui font une meilleure utilisation des ressources disponibles pour résoudre les problèmes. Les applications scientifiques et technologiques exigent régulièrement des algorithmes efficaces pour chercher et localiser les solutions optimales.

Les méthodes numériques d'optimisation non linéaire font l'objet de nombreuses études depuis plusieurs décennies. La méthode de descente de gradient, les méthodes de Newton et la méthode des gradients conjugués [13] font partie de ces algorithmes d'optimisation.

Un des handicaps associés à ces méthodes est leur totale dépendance à l'évaluation des dérivées partielles des fonctions à optimiser. En outre, pour réaliser une optimisation globale, à l'aide de l'une des méthodes énumérées précédemment, il faut choisir un point de départ correspondant à une solution de départ acceptable [4]. De cette manière, la convergence vers un minimum environnant est grandement favorisée.

L'optimisation intelligente se rapporte à une large catégorie d'algorithmes stochastiques basés sur la notion de population, tels que l'évolution différentielle, les algorithmes génétiques, l'optimisation par essaims de particules, etc. Les algorithmes intelligents d'optimisation sont considérés avantageux comparé aux méthodes classiques d'optimisation si le problème d'optimisation est complexe, stochastique, ou fortement non-linéaire avec des optimums locaux multiples en utilisant un minimum de connaissance sur le domaine de recherche [4].

Ces algorithmes intelligents d'optimisation sont intrinsèquement parallèles. La plupart des algorithmes classiques sont périodiques et peuvent seulement explorer l'espace de recherche dans une seule direction à la fois. Les algorithmes intelligents d'optimisation peuvent explorer l'espace de solution dans des directions multiples simultanément. Si un chemin ne fonctionne pas, ils peuvent facilement éliminer ce chemin et continuer le travail sur le plus prometteur. Ceci fournit une plus grande chance de trouver la solution optimale [4].

Depuis de nombreuses années nous assistons à une évolution sans cesse croissante des convertisseurs de l'électronique de puissance, grâce notamment aux énormes progrès effectués dans le domaine des semi-conducteurs. Ce développement rapide a considérablement amélioré la qualité de la vie moderne en offrant aux applications domestiques et industrielles des équipements à rendement optimum et des commandes sophistiquées.

Dans ce contexte et afin de répondre aux besoins grandissant de l'industrie, les puissances convertibles par les convertisseurs statiques sont de plus en plus élevées. Pour satisfaire cette exigence, sans affecté le rendement du convertisseur, l'augmentation de la tension est souvent privilégiée. Cependant, l'augmentation de la tenue en tension d'un composant semi-conducteur entraîne une dégradation de ses performances statiques et dynamiques. Cette contrainte a conduit à l'émergence de nouvelles structures, dites multiniveaux, basées sur l'idée de répartir la contrainte en tension sur plusieurs composants semi-conducteurs basses tensions connectés en série.

De telles structures apportent non seulement une augmentation de la puissance commutée mais également une meilleure qualité spectrale des ondes délivrées. De plus, la valeur de la tension commutée est réduite à la tension de blocage des composants de puissance. Cela, il réduit d'autant les pertes par commutation et les contraintes imposées à la charge qui subit dans les convertisseurs conventionnels une tension qui commute entre ses deux valeurs extrêmes.

Plusieurs topologies de convertisseur multiniveaux ont été proposées dans la littérature. Elles comprennent principalement dans un ordre chronologique, la structure à diodes flottantes, la structure cascadée ainsi que la structure à cellules imbriquées et superposées [4].

Les efforts de la recherche contemporaine dans ce secteur ont largement contribué dans le développement des différentes stratégies de modulation. Selon que la fréquence de commutation soit élevée ou faible, il est possible de distinguer principalement deux classes de techniques de modulation. La première famille comporte les techniques de modulation de largeurs d'impulsions tels que la modulation sinusoïdale et ses variantes et la modulation vectorielle . Alors que, la commande vectorielle et la technique par élimination d'harmoniques constituent la seconde famille[4].

Avec la technique de modulation de largeurs d'impulsions, les harmoniques indésirables se produisent à des fréquences beaucoup plus élevées. Ainsi le filtrage est beaucoup plus facile et moins coûteux. En outre, les fréquences des harmoniques pourrait être au-dessus de la largeur de la bande passante de la charge, ce qui signifie qu'il n'y'a aucune dissipation de puissance due à ces

harmoniques. Cependant, les pertes par commutation augmentent au fur et à mesure que la fréquence de commutation augmente.

De ce fait, dans les applications de la grande puissance, la technique de commande par élimination d'harmoniques semble la plus adaptée pour commander les onduleurs multiniveaux. Cette stratégie peut être interprétée comme un processus de quantification dans lequel la tension sinusoïdale désirée est rapprochée par les niveaux discrets de la tension côté continu. Typiquement, cette solution présente l'avantage de faibles pertes par commutation dans les composants de puissance ce qui augmente ainsi leur longévité du fait qu'ils commutent en faible fréquence. Dans cette technique, le problème de la commande se ramène à la résolution d'un système d'équations algébriques non linéaires exprimant l'amplitude des harmoniques en fonction des angles de commutation des composants de puissance. Toutefois, la difficulté principale de cette méthode est dans le calcul des angles de commutation. De nombreuses approches sont proposées pour les calculer. La plupart de ces travaux utilisent surtout la méthode traditionnelle de Newton-Raphson [2],[7],[8],[14]. Cependant, la convergence de cette méthode dépend fortement des valeurs initiales adoptées. Une deuxième approche à nature analytique basée sur la théorie résultante et les polynômes symétriques a été employée pour transformer les équations harmoniques transcendantales en équations polynômes. Cette méthode a montré son efficacité, notamment, lorsqu'il s'agit de trouver des solutions multiples [17]. Néanmoins, elle atteint ses limites pratiques dès que le nombre d'angles de commutation augmente. Récemment, plusieurs chercheurs ont penché sur l'utilisation des algorithmes évolutionnaires pour l'optimisation de la commande par élimination d'harmoniques des onduleurs multiniveaux. Ils sont focalisés principalement sur les algorithmes génétiques [4].

L'implémentation pratique de la technique de commande par élimination d'harmoniques exige le stockage de toutes les valeurs de ces angles. Le besoin d'une grande capacité mémoire rend la réalisation complexe et coûteuse. Aussi, une approche mathématique exigeant moins de calculs et d'implémentation facile est souhaitée. Cette approche peut être réalisée par les réseaux de neurones artificiels (RNA). Ces derniers sont connus par leur propriété d'approximation universelle parcimonieuse [7],[14].

Les travaux présentés dans ce mémoire. Ils ont comme objectifs :

- L'application d'un algorithme métaheuristique dans l'optimisation de la commande de l'onduleur à sept niveaux. Il s'agit de rendre meilleur le comportement harmonique de la tension de sortie de l'onduleur multiniveaux à l'aide des essais particulières,

- développement de la commande par élimination d'harmoniques des onduleurs multiniveaux en exploitant les réseaux de neurones artificiels (RNA).

Dans le but d'approfondir chacun des points mentionnés ci-dessus, ce mémoire est organisée en quatre chapitres. Dans les paragraphes suivants, on explique brièvement le contenu de chacun.

Le premier chapitre, présente une introduction sur l'optimisation par essaim particulaire. Il est consacré à la présentation des différents types de l'algorithme d'optimisation par essaim particulaire, et l'efficacité du cet algorithme dans la résolution des problèmes non linéaires.

Le deuxième chapitre, présente une introduction sur les réseaux de neurones artificiels. Il est consacré à la présentation des différentes topologies neuronales, l'approximation universelle, et l'apprentissage des réseaux de neurones artificiels.

Le troisième chapitre de ce travail sera consacré à l'établissement d'un modèle mathématique capable de refléter le comportement de l'onduleur triphasé cascadié à sept niveaux vis-à-vis les commandes développées.

La stratégie d'élimination d'harmoniques sera l'objet du dernier chapitre. Dans la première partie de ce chapitre, une méthode évolutionnaire est appliquée pour le calcul des angles de commutation des composants de puissance. Il s'agira de l'optimisation du commande de l'onduleur par les essaims particuliers. par la suite nous avons choisi le réseau de neurones artificiels (RNA) adéquat pour assurer l'approximation des caractéristiques de commande des interrupteurs des onduleurs obtenues par essaims particuliers.

Pour finir, la dernière partie de ce mémoire sera dédiée à la conclusion générale. On évaluera l'ensemble de résultats obtenus et on proposera des possibles extensions des travaux réalisés dans ce mémoire.

Chapitre 1

L'optimisation par essaim particulaire

1.1 Introduction

L'optimisation est un paradigme actuel dans chaque aspect de la vie. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation. Bien que les problèmes d'optimisation soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes complexe, stochastique, ou fortement non-linéaire et ne possèdent donc pas à ce jour de solution algorithmique efficace valable pour toutes les données.

Dans ce chapitre, l'accent sera mis sur une méthode (algorithme) d'optimisation intelligente classée comme métaheuristique inspirée par analogie de l'éthologie. Il s'agit l'algorithme d'optimisation par essaim particulaire (OEP). Nous nous intéressons, par la suite, notamment à la présentation des différents types de l'algorithme d'optimisation par essaim particulaire, et l'efficacité de cet algorithme dans la résolution des problèmes non linéaires.

1.2 Optimisation par essaim particulaire

La bio-inspiration est un changement de paradigme qui amène les ingénieurs à s'inspirer de la nature pour développer de nouveaux systèmes artificiels. Depuis son apparition, l'optimisation par essaim particulaire a gagné la popularité croissante parmi des chercheurs et des praticiens comme une technique robuste et efficace pour résoudre des problèmes d'optimisation difficiles [5].

L'optimisation par essaim particulaire (OEP) est classée dans les techniques d'optimisation stochastiques à population introduite en 1995 par *James Kennedy* (psychologue social et statisticien) et *Russel Eberhart* (ingénieur) sous le nom Particle Swarm Optimization (PSO) [1],[4],[5],[20]. PSO est inspiré par le comportement social des animaux, comme les essaims d'abeilles, les vols d'oiseaux et les bancs de poissons [3],[4],[5]. En effet, tout comme ces animaux se déplacent en groupe pour trouver de la nourriture, ou émigrent, Si un membre voit un chemin souhaitable pour l'exploration, le reste de cet essaim suivra rapidement. Raison pour laquelle, cette technique est fondée sur la notion de coopération et l'échange d'information entre les particules. Les individus de l'algorithme sont appelés particules et la population est appelée essaim.

1.2.1 Les différents types du PSO

1.2.1.1 PSO du optimum global

Dans cette version le voisinage pour chaque particule est l'essaim entier. Le réseau social utilisé par ce type d'algorithme est la topologie d'étoile, figure (1.1). Pour la topologie de voisinage d'étoile, le composant social du mise à jour de vitesse de particules reflète l'information obtenue à partir de toutes les particules dans l'essaim. Dans ce cas-ci, l'information sociale est la meilleure position trouvée par l'essaim, désigné sous le nom de g .

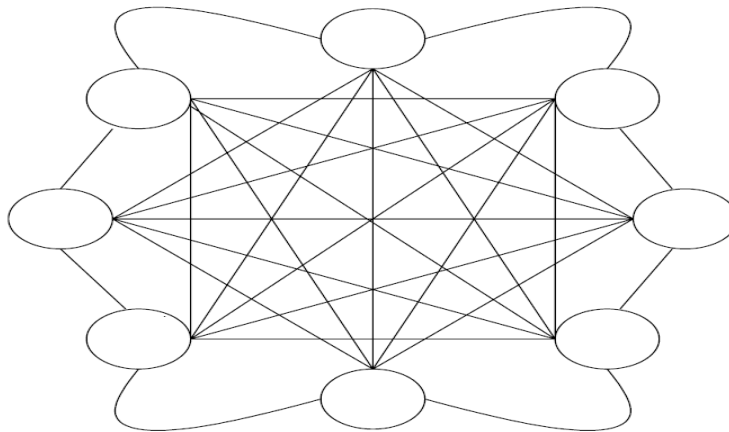


Figure (1.1) L'étoile topologie de voisinage

1.2.1.1.1 Principe de déplacement d'une particule dans PSO du optimum global

L'algorithme PSO dans sa version globale est initialisé par une population de solutions potentielles aléatoires interprétées comme des particules se déplaçant dans l'espace de recherche [3],[19]. Toutes les particules étant considérées voisines et chacune d'elle dispose d'une vitesse et une petite mémoire, lui permettant de se souvenir de sa meilleure performance, en position et en

valeur. La performance de chaque particule est mesurée selon une fonction objective prédéfinie relative au problème à résoudre.

La théorie de la socio-psychologie suggère que les individus sont influencés par leur comportement passé et par le succès de leurs voisins dans le réseau social et non pas nécessairement dans l'espace [16]. Dans ce contexte, chaque particule interroge chaque congénère de proximité sur sa meilleure performance. Sur la base des informations dont elle dispose, chaque particule modifie sa vitesse d'une manière stochastique en fonction de sa propre vitesse actuelle, sa meilleure performance et la meilleure performance de ses voisins.

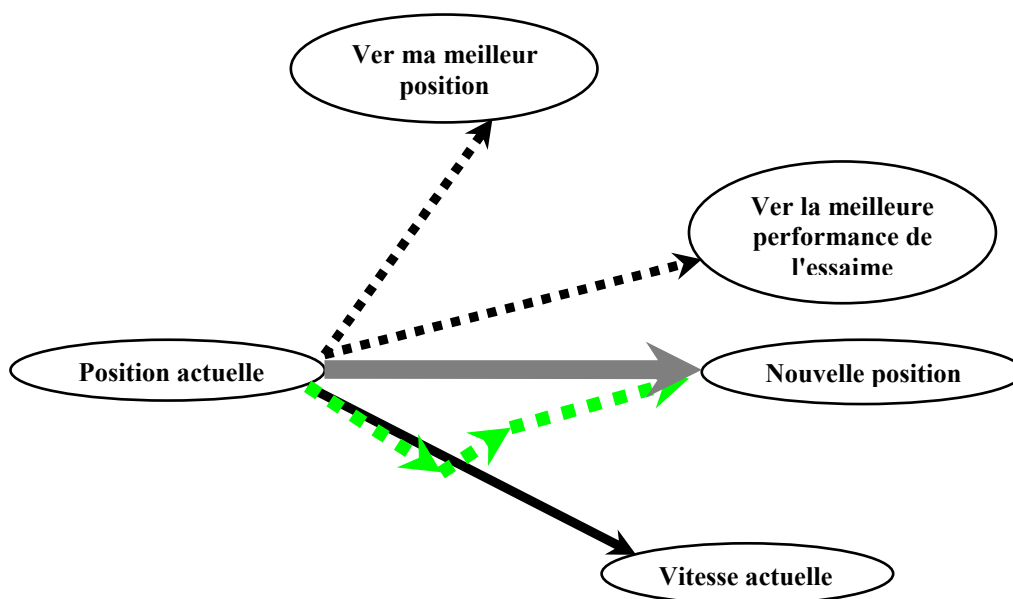


Figure (1.2) Principe de déplacement d'une particule dans PSO du optimum global

La figure (1.2) montre que la décision de déplacement d'une particule est basée sur une combinaison linéaire de trois tendances à savoir de suivre sa propre voie, revenir vers sa meilleure performance et aller vers la meilleure performance de ses voisins.

Chaque membre de l'essaim conserve une connaissance partielle de sa meilleure position p^i , et la meilleure position de toutes les particules de l'essaim g et ajuste dynamiquement sa propre vitesse en se basant sur ces informations. L'ajustement de vitesse est basé sur les comportements historiques des particules eux-mêmes aussi bien que leurs compagnons. La particule la mieux placée dans une région proche de l'endroit de la nourriture (optimum global) va motiver l'essaim d'explorer de proximité cette région et les régions adjacentes. De cette façon, les particules tendent à se rapprocher davantage de cet optimum d'une étape à l'autre durant le processus de recherche. Dans chaque étape on définit une nouvelle meilleure particule qui, à son tour, va attirer vers elle

de l'essaim.

La particule ajuste dynamiquement sa propre vitesse en se basant sur les comportements historiques des particules eux-mêmes aussi bien que leurs compagnons .

La vitesse de chaque particule peut être mise à jour itérativement selon l'équation suivante :

$$v^i(k + 1) = wv^i(k) + c_1 rand_1 (p^i - x^i(k)) + c_2 rand_2 (g - x^i(k)) \quad (1.1)$$

Où

k : est le nombre d'itérations,

w : est le coefficient d'inertie,

$v^i(k)$: est la vitesse courante de la particule i à l'itération k ,

$x^i(k)$: est la position actuelle de la particule i à l'itération k ,

c_1, c_2 : sont les coefficients d'accélération,

$rand_1, rand_2$: sont des nombres aléatoires de distribution uniforme sur $[0,1]$.

Chaque particule rejoint sa nouvelle position dans l'espace de recherche en utilisant l'équation suivante :

$$x^i(k + 1) = x^i(k) + v^i(k + 1) \quad (1.2)$$

Après un certain nombre d'itérations, les particules se regroupent par la suite dans la région où les solutions sont les plus convenables (probables). Si une particule tend à sortir de l'espace de recherche, on lui attribue la valeur du point frontière le plus proche avec ralentissement voir annulation de sa vitesse.

L'algorithme du optimum global a été décrite par plusieurs auteurs [1],[3],[10],[16]. L'organigramme de la figure (1.3) montre comment PSO du optimum global opère.

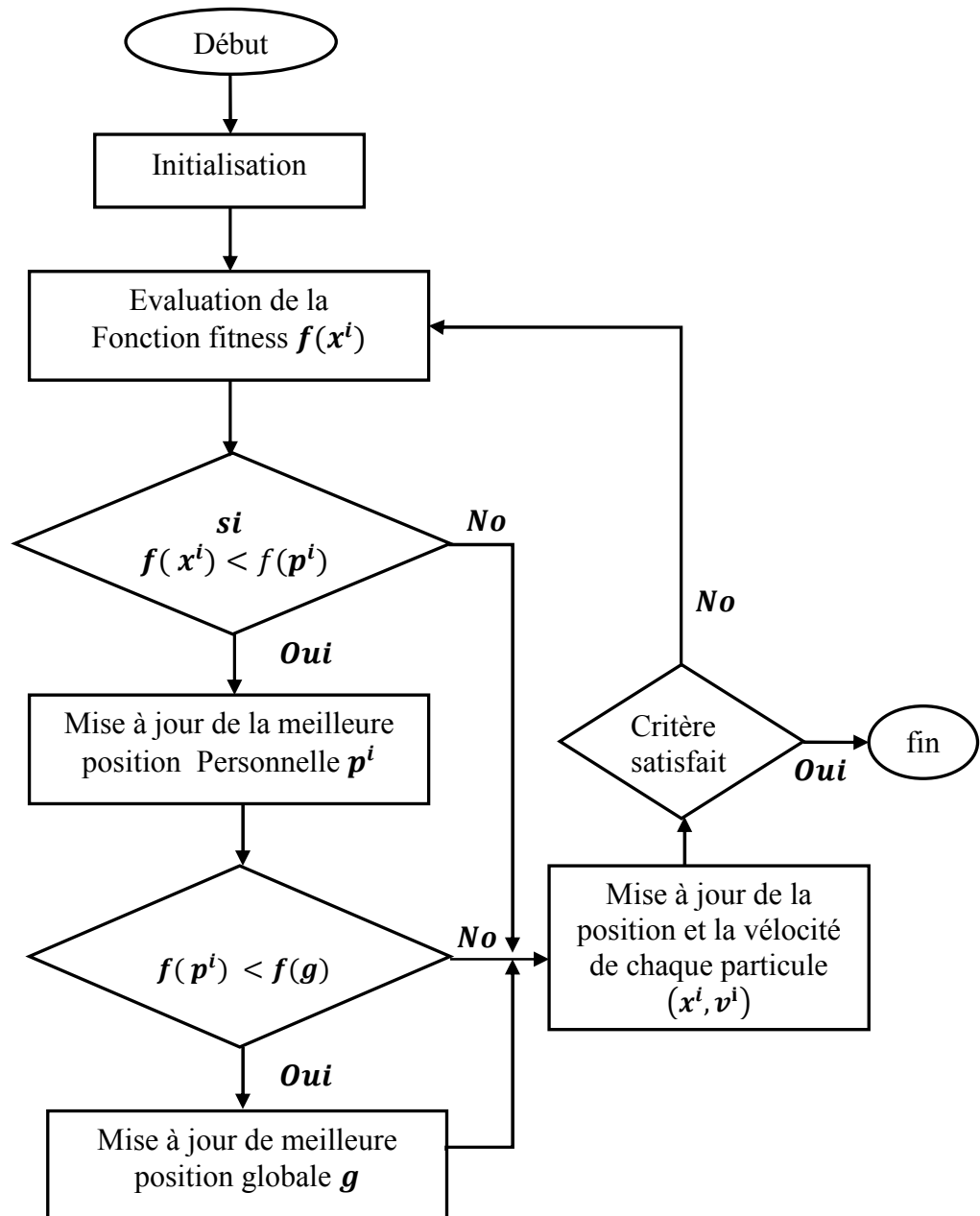


Figure (1.3) Organigramme de l’algorithme PSO du optimum global

1.2.1.2 PSO du optimum local

Dans cette version de PSO , utilisations d' une topologie de réseau sociale d'anneau où un plus petits voisinages sont définis pour chaque particule, figure (1.4). Le composant social reflète l'information échangée dans le voisinage de la particule, reflétant la connaissance locale de l'environnement. La contribution sociale à la vitesse de particules est proportionnelle à la distance entre une particule et la meilleure position a trouvé par le voisinage des particules.

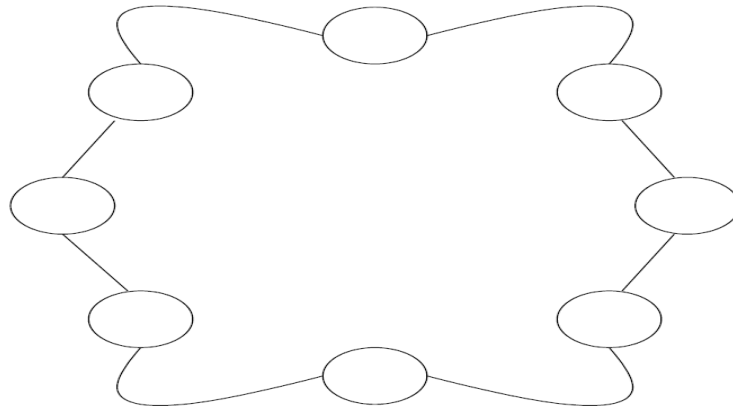


Figure (1.4) L'anneau topologie de voisinage

1.2.1.2.1 Principe de déplacement d'une particule dans PSO du optimum local

L'algorithme PSO dans sa version locale est initialisé par une population de solutions potentielles aléatoires interprétées comme des particules se déplaçant dans l'espace de recherche[3]. comme illustré sur la figure (1.5), montre comment des particules sont influencées par leurs voisins immédiats. Pour maintenir le graphique lisible, seulement certains des mouvements sont illustrés, et seulement la direction globale de vitesse est indiquée. Le *voisinage 1*, les particules *a* et *b* se déplacent vers la particule *c*, qui est la meilleure solution dans ce voisinage. Considérant le *voisinage 2*, les mouvements des particules *d* vers *f*, fait ainsi *e*. Pour la prochaine itération, *e* sera la meilleure solution pour le *voisinage 2*. Mouvement maintenant de *d* et de *f* vers *e* comme illustré sur la figure (1.5) (b) (seulement une partie de l'espace de solution est illustrée). Les blocs représentent les positions précédentes. Noter que *e* reste le meilleur solution pour le *voisinage 2*. Noter également le mouvement général vers le minimum .

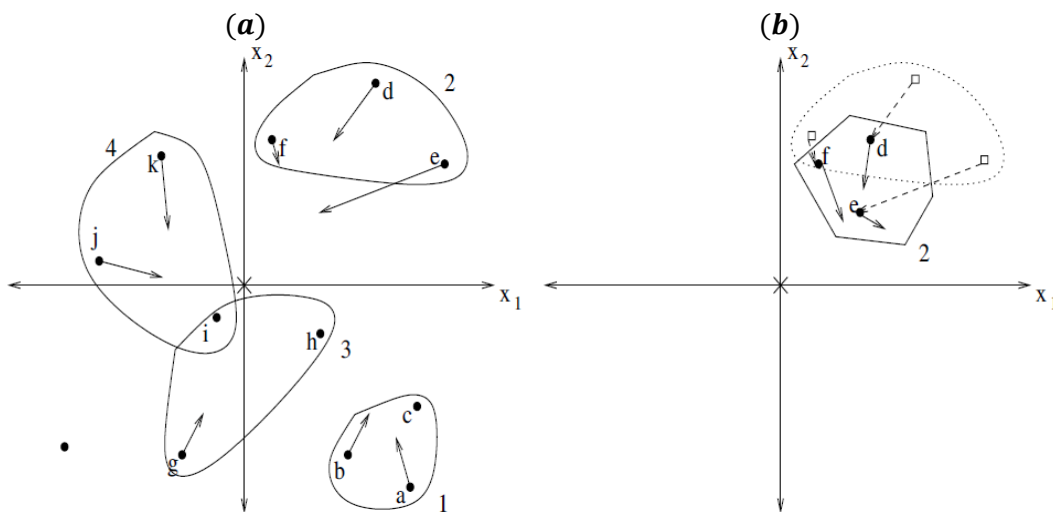


Figure (1.5) Principe de déplacement d'une particule dans PSO du optimum local

La particule ajuste dynamiquement sa propre vitesse en se basant sur les comportements historiques des particules eux-mêmes aussi bien que leurs voisins immédiats.

La vitesse de chaque particule peut être mise à jour itérativement selon l'équation suivante :

$$v^i(k+1) = wv^i(k) + c_1 \text{rand}_1 (p^i - x^i(k)) + c_2 \text{rand}_2 (g_n - x^i(k)) \quad (1.3)$$

Où

g_n : est la meilleure position découverte par les voisins immédiats .

la meilleure position découverte par les voisins immédiats N_i est défini comme suit :

$$g_n(k+1) \in \{N_i / f(g_n(k+1)) = \min\{f(x(k)), \forall x \in N_i\} \} \quad (1.4)$$

Il est important de noter que pour la PSO de base, les particules dans un voisinage N_i n'ont aucun rapport entre eux. Le choix des voisinages est fait à basé sur des index de particules. Cependant, des stratégies ont été développées où des voisinages sont formés ont basé sur la similitude spatiale.

L'algorithme PSO du optimum local a été décrite par plusieurs auteurs [1],[3],[10][16]. L'organigramme de la figure (1.6) montre comment PSO du optimum local opère.

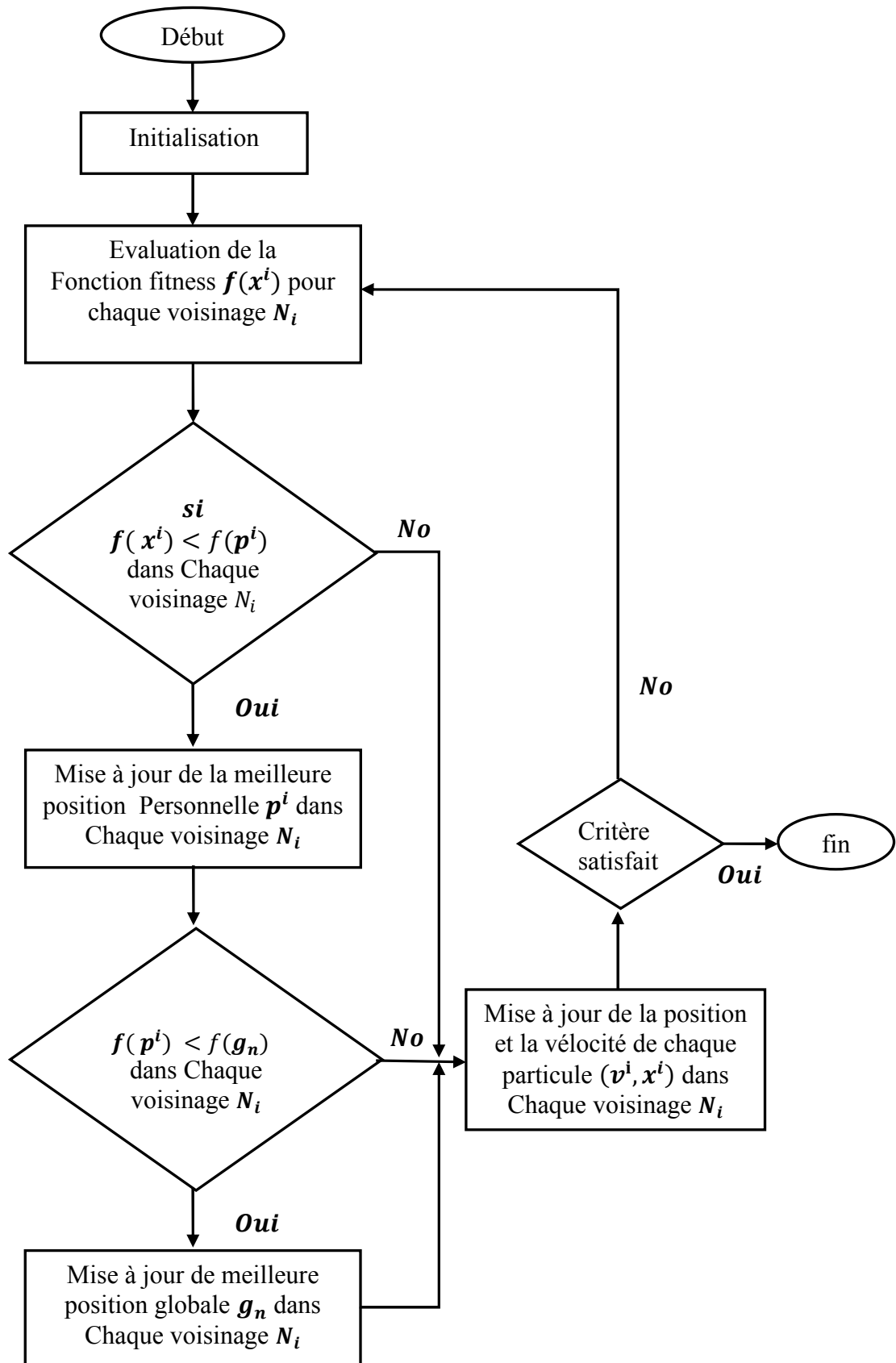


Figure (1.6) Organigramme de l’algorithme PSO du optimum local

1.2.2 Les Composants de vitesse

Le calcul de vitesse du i^{eme} particule comme donné dans les équations (1.1) et (1.2) compose de trois termes :

- **La vitesse précédente**, $v^i(k)$, qui sert de mémoire de la direction précédente de vol, c.-à-d. mouvement dans le passé immédiat. Cette terme de mémoire peut être vue comme élan, qui empêche la particule de la direction rigoureusement changeante, et à la polarisation vers la direction courante. Ce composant désigné également sous le nom du composant d'inertie.
- **Le composant cognitif**, $c_1 rand_1 (p^i - x^i(k))$, qui mesure la performance de la particule i relativement à sa performance passée. Dans une certaine mesure, le composant cognitif ressemble la mémoire individuelle du position qui était la meilleure pour la particule. L'effet de cette terme est que ces particules sont attirées à leurs propres meilleures positions, ressemblant la tendance des individus de retourner aux situations ou aux endroits qui les ont satisfaits plus dans le passé. *Kennedy et Eberhart* se sont également référés au composant cognitif comme « nostalgie » de la particule .
- **Le composant social**, $c_2 rand_2 (g - x^i(k))$, dans la version globale de PSO ou , $c_2 rand_2 (g_n - x^i(k))$, dans la version local ,qui mesure la performance de la particule i relativement à un groupe de particules, ou des voisins. Conceptuellement, le composant social ressemble à une norme ou à une norme de groupe que les individus cherchent à atteindre. L'effet du composant social est que chaque particule est attirée vers la meilleure position trouvée par sa voisinage.

La figure (1.7) présente un illustration géométrique des mises à jour de vitesse et de position:

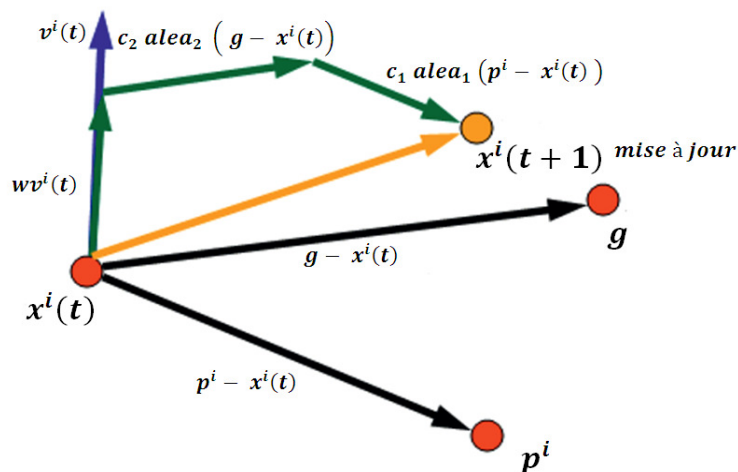


Figure (1.7) Schéma vectorielle du déplacement d'une particule

1.2.3 Les paramètres du PSO

Le PSO de base est influencé par un certain nombre de paramètres de contrôle, à savoir la dimension du problème, nombre de particules, coefficients d'accélération, poids d'inertie, voisinage taille, nombre d'itérations, et les valeurs aléatoires qui influencent la contribution des composants cognitifs et sociaux [3],[21].

- **Taille de l'essaim** : On lui a montré dans un certain nombre d'études empiriques que le PSO a la capacité de trouver les solutions optimales avec de petites tailles d'essaim de 10 à 30 particules.
- **Taille de voisinage** : La taille de voisinage définit l'ampleur de l'interaction sociale dans l'essaim. Plus les voisinages sont petits, moins l'interaction se produit. Tandis que de plus petits voisinages sont plus lents dans la convergence, ils ont une convergence plus fiable aux solutions optimales. De plus petites tailles de voisinage sont moins susceptibles des minimum locaux. Pour profiter des avantages de petites et grandes tailles de voisinage, commencer la recherche avec de petits voisinages et augmenter la taille de voisinage proportionnellement à l'augmentation en nombre des itérations. Cette approche assure une première diversité élevée avec une convergence plus rapide pendant que les particules se déplacent vers un secteur prometteur de recherche.
- **Nombre d'itérations** : Le nombre d'itérations pour atteindre une bonne solution est également dépendant à la problème. Trop peu d'itérations peuvent terminer la recherche prématurément. Un nombre trop grand d'itérations a la conséquence de la complexité du calcul supplémentaire inutile (à condition que le nombre d'itérations est le seul état d'arrêt).
- **Les coefficients d'accélération**: influencent la valeur maximale du déplacement qu'une particule peut prendre durant une seule itération. c_1 , c_2 sont désignés également sous le nom de paramètres cognitif et social, respectivement. Les nombres aléatoires $rand_1$, $rand_2$ sont employés pour affecter la nature stochastique de l'algorithme.
- **Le coefficient (poids) d'inertie** : a été développée par *Eberhart et al*, régit la quantité de vitesse actuelle qui devrait être maintenue pour influencer le calcul de la nouvelle vitesse. De cette façon, le paramètre w règle les capacités d'exploration globale et locale de l'essaim et influence le comportement de convergence de l'algorithme. Un grand poids d'inertie facilite l'exploration globale (exploration de nouveaux secteurs), alors qu'une petite valeur tend à faciliter l'exploration locale (exploration fine du secteur courant de recherche). Une valeur appropriée du poids de l'inertie fournit habituellement l'équilibre entre les capacités globales et locales d'exploration et par conséquent une réduction du

nombre d'itérations nécessaires pour localiser la solution optimale . En règle générale, il est souhaitable de commencer le poids d'inertie à une grande valeur, afin de faire une meilleure exploration globale de l'espace de recherche, et le diminue graduellement pour affiner l'exploration locale.

La figure (1.8) illustre le diminue progressif en poids de l'inertie:

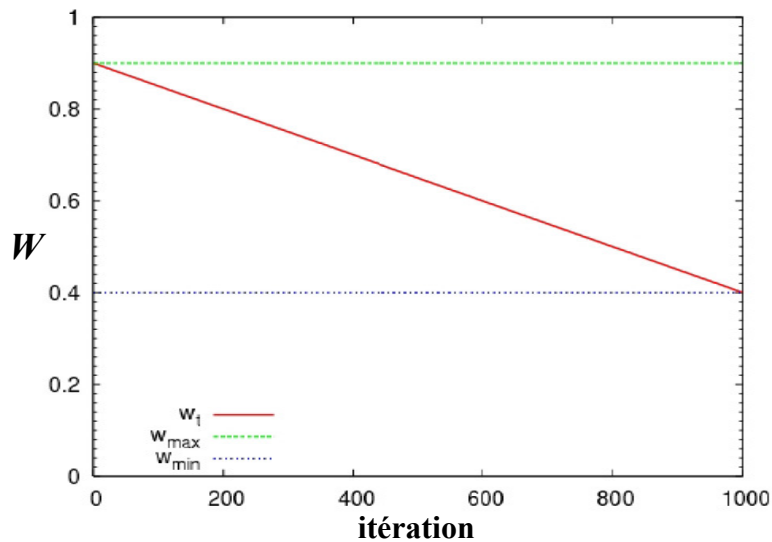


Figure (1.8) Le diminue progressif en poids de l'inertie

1.2.4 La comparaison entre les deux version

Il y a deux différences principales entre les deux approches en ce qui concerne leurs caractéristiques de convergence :

- En raison de l'interconnectivité plus grande de particules du PSO du optimum global, il converge plus rapidement que le PSO du optimum local. Cependant, cette convergence plus rapide vient au coût de moins de diversité que le PSO du optimum local.
- Par suite de sa plus grande diversité (qui a comme conséquence de plus grandes parties de l'espace de recherche étant couvert), le PSO du optimum local est moins susceptible de l'emprisonnement dans des minimum locaux. En général (selon le problème), le voisinage structure tels comme la topologie d'anneau utilisée dans le PSO du optimum local améliore les performances.

1.3 Test de l'efficacité du PSO

Dans le but de test les capacités du PSO , nous avons choisi de chercher l'optimum global de la fonction Rosenbrock défini par :

$$f(x_1, x_2) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 \quad (1.5)$$

Cette fonction, comme le montre la figure (1.9), a plusieurs optimums locaux et n'admet qu'un seul optimum global défini au point (1,1) tel que $f(1,1) = 0$.

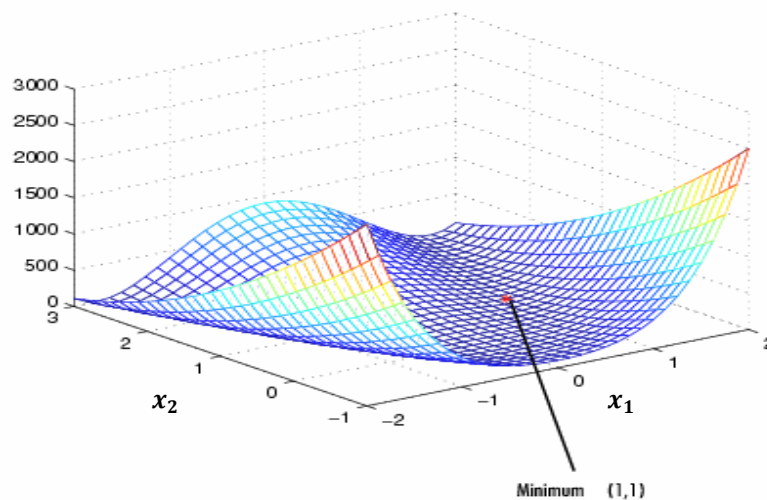


Figure (1.9) Fonction de Rosenbrock

On prend :

- $S = 20$; taille de l'essaim .
- $k = 1000$; nombre d'itérations.
- $c_1 = c_2 = 1.8$
- w ; d'après *Eberhart et al*
- Condition d'arrêt est ($k=1000$ Ou la valeur du fonction objective "erreur" =0)

Le résultat obtenu est : $x_1 = 1$, $x_2 = 1$, *erreur* $\cong 0$ *après 357 itiration*

La figure (1.10) montre la convergence de la fonction objective vers l'optimum global.

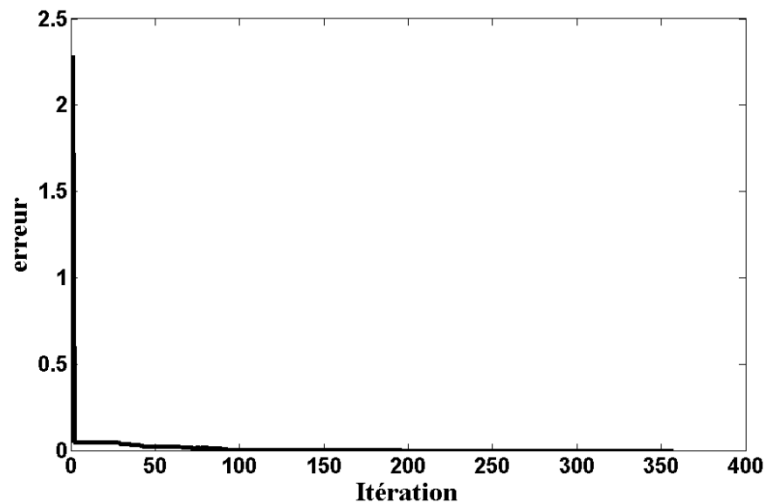


Figure (1.10) convergence de la fonction objective vers l'optimum global.

1.4 Les champs d'application du PSO

Cet algorithme se révèle efficace pour les problèmes d'optimisation non linéaires à variables continue ou discrète. Il a fait sa preuve dans des domaines variés entre autres comme l'optimisation de:

- l'apprentissage de réseaux de neurones [3],
- la commande par logique floue [3],
- les régulateurs PID [5],
- régulation et optimisation de l'écoulement de puissance [4],
- l'identification et le diagnostic des machines électriques [4]
- la commande des convertisseurs de puissance [4].
- ...etc.

1.5 Conclusion

Parce que elle est simple à comprendre, à programmer et à utiliser, PSO se révèle efficace pour les problèmes d'optimisation non linéaires. La puissance de cet algorithme découle de la nature même de son principe fondé sur la notion de coopération entre les particules. PSO semble très performant en termes de la précision des solutions trouvées, du temps de calcul et du nombre de paramètres à ajuster.

Pour notre système, nous avons opté pour l'algorithme PSO du optimum global car il semble bien adapté au problème posé en termes de temps de calcul.

Chapitre 2

Les Réseaux de Neurones Artificiels

2.1 Introduction

Avec l'apparition des ordinateurs l'homme a découvert un moyen d'effectuer diverses tâches avec deux capacités non négligeables que lui ne possède pas : la rapidité et la précision.

Cependant l'exécution d'une tâche pour l'ordinateur nécessite sa programmation préalable par l'homme. Cette caractéristique fait apparaître les ordinateurs comme des machines exécutant des ordres « aveuglement » et l'homme n'a pas désespéré de voir un jour construire une machine à son image, c'est à dire intelligente, capable d'apprendre, de raisonner, de réfléchir sans son intervention.

Ce sont des recherches basées sur le fonctionnement du cerveau qui ont constituées le point de départ de cette gigantesque recherche.

Des travaux de neurobiologistes ont, en effet, révélé que le cerveau est constitué d'un nombre extrêmement élevé d'unités de traitement élémentaire de l'information (les neurones biologiques) fortement interconnectées. Ces recherches sont l'origine de l'inspiration des réseaux de neurones artificiels.

Dans ce chapitre nous allons présenter les concepts fondamentaux des réseaux de neurones artificiels, les différentes architectures neuronales, les algorithmes d'apprentissage et les domaines d'application des réseaux de neurones.

2.2 Le modèle neurophysiologique

Les neurones sont les cellules nerveuses constituant le système nerveux. Chez l'être humain, on en compte environ cent milliards 10^{11} , ce qui correspond à environ 10^{15} connexions [22]. La figure (2.1) montre les éléments essentiels constituant le neurone biologique.

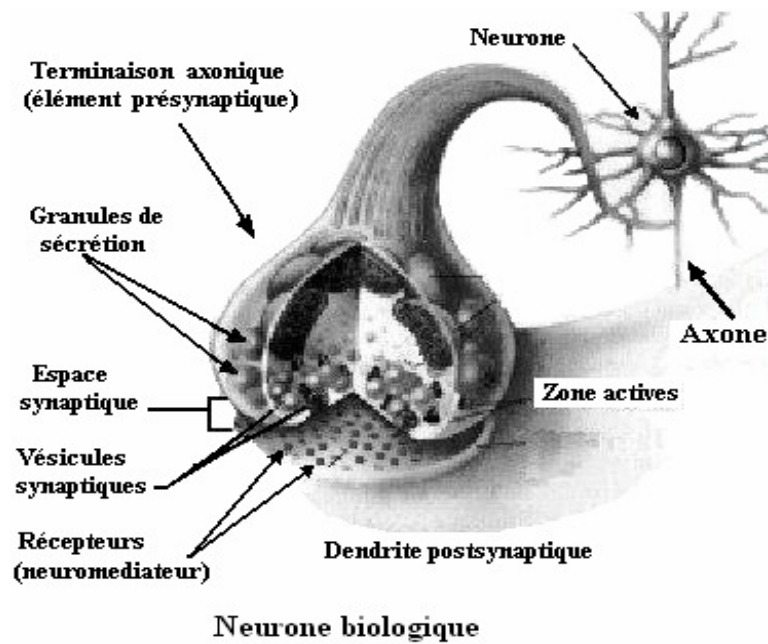


Figure (2.1) Les éléments constituant le neurone biologique

Un neurone biologique est composé de trois parties essentielles :

- Le corps cellulaire effectue les transformations biochimiques nécessaires à la vie du neurone.
- Les dendrites forment une arborescence autour du corps cellulaire et permettent au neurone de capter les signaux qui parviennent de l'extérieur.
- L'axone est une fibre nerveuse qui transporte les signaux émis par le neurone, il se ramifie en extrémité là où il communique avec les autres neurones à travers des synapses.

L'influx nerveux se propage toujours de la dendrite vers le corps cellulaire et de celui-ci vers l'axone. Chaque neurone reçoit des signaux excitateurs ou inhibiteurs par ses dendrites. Ces signaux pondérés sont combinés dans le corps cellulaire. Le potentiel résultant est comparé au seuil interne, s'il est supérieur à celui-ci, le neurone provoque l'émission dans l'axone d'un train d'impulsions vers les synapses, dans le cas contraire il reste inactif.

2.3 Le modèle mathématique (*Neurone Artificiel*)

Un neurone artificiel est une modélisation simplifiée du neurone biologique, c'est un processeur élémentaire qui reçoit un nombre d'entrées en provenance des autres neurones ou de l'extérieur. A chacune de ces entrées est associée un poids W abréviation de *weight* (poids en anglais) représentatif de la force (ou bien de la pondération) de la connexion.

La figure (2.2) montre le modèle du neurone artificiel.

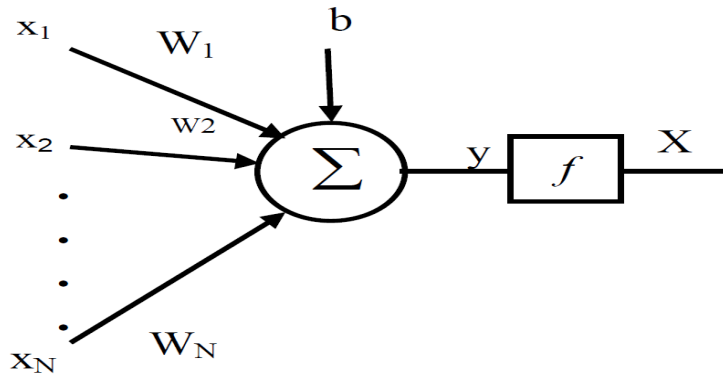


Figure (2.2) Modèle du neurone artificiel

La sortie du neurone est une somme pondérée de ses entrées plus un biais, multipliée par une fonction d'activation qui peut être linéaire ou non linéaire (identité, sigmoïde, tangente hyperbolique, etc.) [22]. Une fonction d'activation linéaire aboutira à une structure linéaire, qui défait le but d'une mise en oeuvre des réseaux de neurones.

$$y = \sum_{i=0}^n W_j x_j + b \quad (2.1)$$

$$X = f(y)$$

x_j est la valeur de la j^{eme} entrée du neurone, W_j est le poids synaptique correspondant au neurone j , b désigne le biais et $f(.)$ la fonction d'activation. La fonction la plus utilisée est de type sigmoïde [22]. Elle est définie par :

$$f(y) = \frac{1 - e^{-\delta \cdot y}}{1 + e^{-\delta \cdot y}} \quad (2.2)$$

Où

δ dénote le paramètre de la sigmoïde qui définit le degré de la non linéarité , $f(y) \in [-1, 1]$;

$\forall y \in IR$.

La représentation de la fonction sigmoïde pour différentes valeurs de δ est donnée par la figure(2.3).

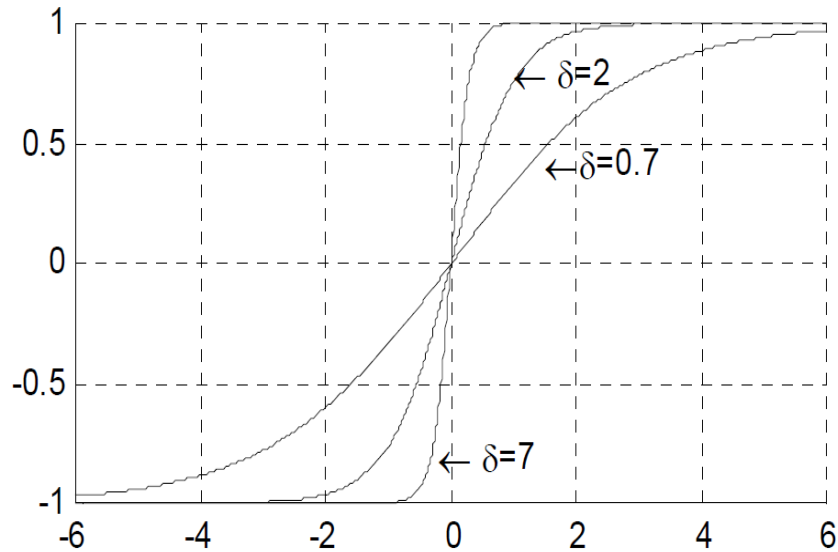


Figure (2.3) La fonction d'activation sigmoïde

2.4 Propriété fondamentale des réseaux de neurones : L'approximation Universelle

La propriété d'approximation universelle, justifie dans une large mesure, l'utilisation des réseaux de neurones dans les domaines des sciences de l'ingénieur. La propriété d'approximation universelle des réseaux de neurones peut s'énoncer comme suit [14] :

- Pour toute fonction déterministe suffisamment régulière, il existe au moins un réseau de neurone non bouclé, possédant une couche de neurones cachés et un neurone de sortie linéaire, qui réalise une approximation de cette fonction et ces dérivées successive, au sens de moindres carrées, avec une précision arbitraire.

Ce théorème alors ne s'applique que si les fonctions d'activités des neurones cachés sont continues et bornées, telles que les fonctions sigmoïdes et les fonctions à base radiale.

Nous signalons, que cette propriété n'est pas spécifique aux réseaux de neurones. Les polynômes, les séries de Fourier possèdent aussi cette particularité. La différence avec les réseaux de neurones réside dans leur parcimonie. En effet, pour obtenir une approximation d'une précision donnée, les réseaux de neurones utilisent moins de paramètres que les approximateurs usuels. En particulier, le nombre de paramètres varie essentiellement de manière linéaire en fonction du nombre de variable de la fonction que l'on cherche à approcher, alors qu'il varie beaucoup plus rapidement avec la dimension de l'espace des entrées dans le cas des approximateurs usuels.

En pratique, les RNA sont donc avantageux, par rapport à d'autres approximateurs Conventionnels, surtout pour des systèmes à plus d'une variable. Les réseaux de neurones offrent donc les avantages suivants [14] :

- l'estimation des fonctions est d'autant plus rapide que le nombre de paramètres à calculer est petit ;
- avec un minimum d'exemple à l'entrée du réseau, il est possible d'approximer une fonction délicate, et difficile à traiter analytiquement. Le fait d'utiliser moins de coefficients que les méthodes classiques de régression, permet donc une économie en nombre d'exemples d'entrées, ce qui peut être particulièrement important lorsque d'acquisition des exemples est coûteuse ou lente.

Il en résulte de tous ça, que :

- les RNA sont des outils statiques de régression qui permettent l'approximation, au sens des moindres carrés, de toute fonction non linéaire suffisamment régulière ;
- cette approximation est caractérisée par une économie, en nombre de coefficients, donc en nombre d'exemples.

2.5 Structure d'interconnexion

Les connexions entre les neurones qui composent le réseau décrivent la topologie du modèle. Elle peut être quelconque, mais le plus souvent il est possible de distinguer une certaine régularité [15]. Dans le présent chapitre, nous abordons en détail la topologie des multicouches .

2.6 Différentes topologies neuronales

2.6.1 Réseau multicouche (MLP)

Dans les réseaux MLP « Multi Layer Perceptron », les neurones sont arrangés par couche. Il n'y a pas de connexion entre neurones d'une même couche et les connexions ne se font qu'avec les neurones des couches avales. Habituellement, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante et à celle-ci seulement. Ceci nous permet d'introduire la notion de sens de parcours de l'information (de l'activation) au sein d'un réseau. On appelle couche d'entrée l'ensemble des neurones d'entrée, couche de sortie l'ensemble des neurones de sortie. Les couches intermédiaires n'ayant aucun contact avec l'extérieur sont appelées couches cachées. L'architecture du réseau multicouche MLP est représentée par la figure (2.4).

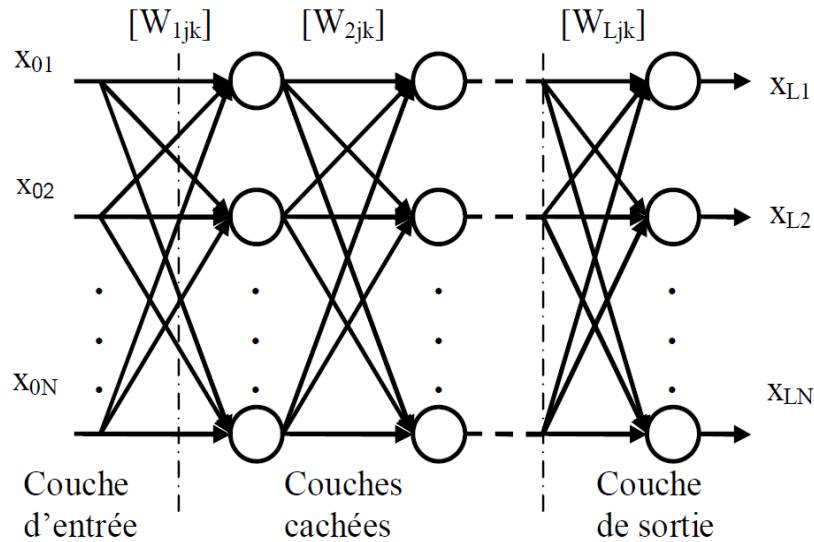


Figure (2.4) Architecture du perceptron multicouche MLP

Le réseau MLP contient N entrées, $(L-1)$ couches cachées de N_i neurones chacune ($1 < i < L-1$) et une couche de sortie contenant N_L neurones. L'indice de la couche est i , x_{ik} est la sortie du neurone k de la couche i , W_{ijk} est le poids reliant la sortie $x_{(i-1)j}$ au neurone k de la couche i , N_i est le nombre de neurones dans la couche i . Avec ces notations, la sortie x_{ik} du neurone (i, k) sera donnée par:

$$y_{ik} = \sum_{j=1}^{N(i-1)} W_{ijk} x_{(i-1)j} + b_{ik} \quad (2.3)$$

$$X_{ik} = f(y_{ik})$$

Le filtrage non linéaire du MLP dépend énormément du nombre de couches cachées, le réseau peut avoir plusieurs de ces couches [22]. Cependant, trop de couches cachées compliquera l'apprentissage et augmentera le coût de calcul de la structure et peu de couches cachées sera inadéquat pour créer la non linéarité suffisante.

Il a été démontré qu'un réseau MLP (Multi Layer perceptron) avec seulement deux couches cachées dont la fonction d'activation est de type sigmoïde et une sortie avec une fonction d'activation linéaire peut approximer n'importe quelle fonction continue pourvu qu'un nombre suffisant de neurones soit fourni [22]. Cette propriété est dite propriété d'approximation universelle.

2.6.2 Réseau à connexions locales

Il s'agit d'une structure multicouche, conserve une certaine topologie. Chaque neurone entretient des relations avec un nombre réduit et localisé de neurones de la couche avale, figure(2.5).

Les connexions sont donc moins nombreuses que dans le cas d'un réseau multicouche classique.

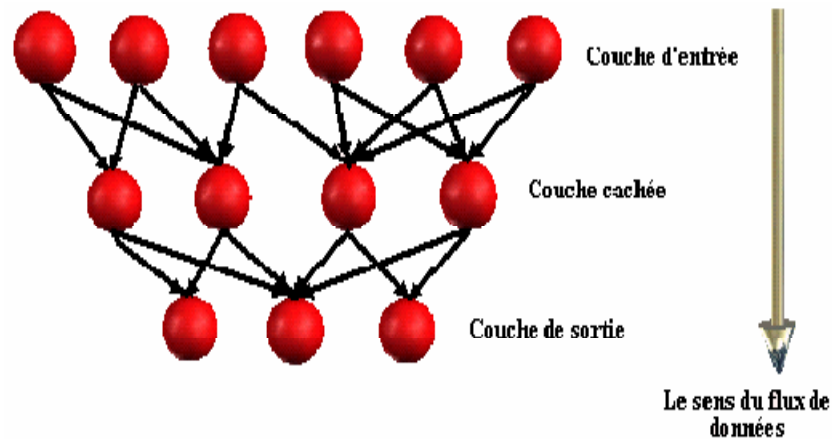


Figure (2.5) Réseau à connexion local.

2.6.3 Réseau à connexions récurrentes

Les connexions récurrentes, ramènent l'information en arrière par rapport au sens de propagation, défini dans un réseau multicouche. Ces connexions sont le plus souvent locales , figure(2.6).

Les réseaux récurrents sont appelés aussi réseaux dynamiques. Cela est dû à la caractéristique dynamique de ces réseaux, ils évoluent dans le temps car la sortie actuel $y(t)$ de ces réseaux dépendent aussi des valeurs de sorties précédente $y(t-1)$. La notion du temps dans ces réseaux est importante.

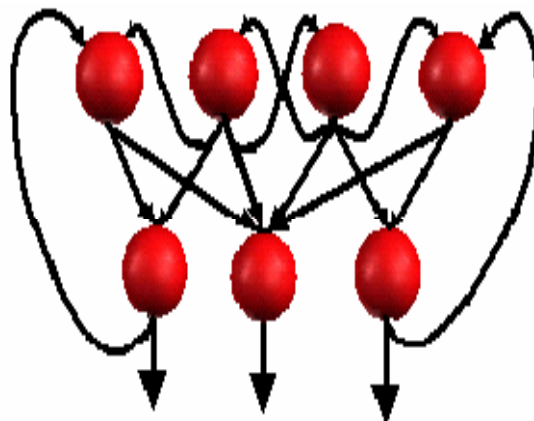


Figure (2.6) Réseau à connexions récurrentes.

2.6.4 Réseau à connexion complète

C'est la structure d'interconnexion la plus générale. Chaque neurone est connecté à tous les neurones du réseau et à lui-même, figure (2.7).

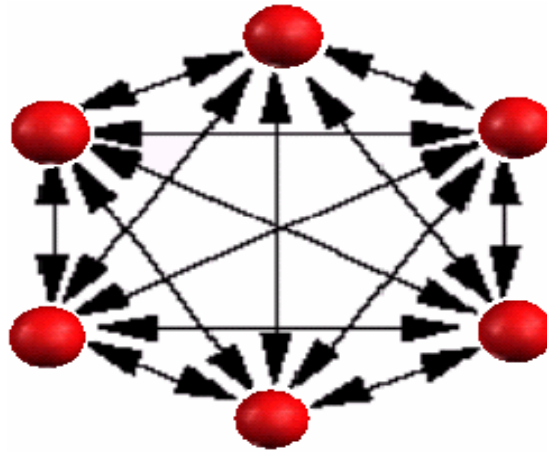


Figure (2.7) Réseau à connexion complète.

Le réseau à connexion complète appelé encore le modèle de Hopfield fut présenté en 1982. Ce modèle est basé essentiellement sur le principe des mémoires associatives.

2.7 Apprentissage des réseaux de neurones

L'apprentissage est la propriété la plus intéressante des réseaux neuronaux. Durant cette phase, le comportement du réseau est modifié à travers les poids en fonction d'un jeu de données présentées en entrée, jusqu'à l'obtention du comportement désiré.

L'apprentissage peut être supervisé ou non supervisé selon la présence ou l'absence de la réponse désirée $d(n)$.

2.7.1 Apprentissage supervisé

L'apprentissage est dit supervisé lorsque le réseau est forcé à converger vers un état final précis, ce qui nécessite la connaissance *à priori* de la réponse désirée $d(n)$. La méthode la plus utilisée est la rétropropagation du gradient [22]. Elle consiste à présenter des exemples au réseau, calculer sa sortie, ajuster les poids de façon à réduire l'écart entre cette sortie et la réponse désirée pour satisfaire un certain critère de performance, figure (2.8).

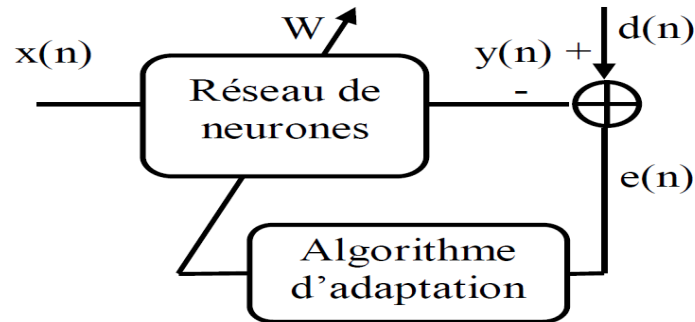


Figure (2.8) Mode d'apprentissage des réseaux de neurones supervisé

2.7.2 Apprentissage non supervisé

Dans l'apprentissage non supervisé, seules les valeurs d'entrée sont disponibles et le réseau est laissé libre de converger vers n'importe quel état final. La connaissance *à priori* de la sortie désirée n'est pas nécessaire, la procédure d'apprentissage est basée uniquement sur les valeurs d'entrées. Le réseau s'auto-organise de façon à optimiser une certaine fonction de coût, figure (2.9).

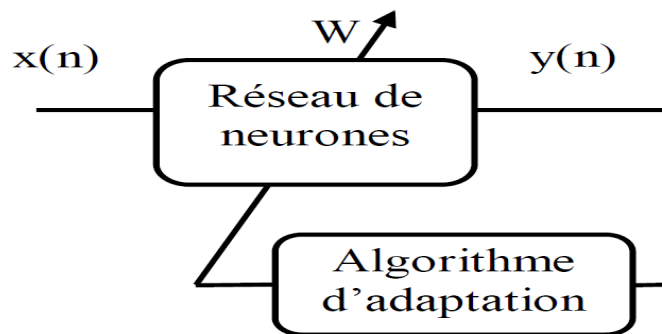


Figure (2.9) Mode d'apprentissage des réseaux de neurones non supervisé

2.8 L'algorithme de rétropropagation

La rétropropagation du gradient BP (Back Propagation) est une généralisation de la règle de « Widrou-Hoff » pour un réseau multicouche [22]. Inventée par Rumelhart, Hinton et Williams en 1986 [22]. C'est un algorithme supervisé dont l'objectif est d'ajuster les poids du réseau MLP de façon à minimiser une fonction de coût différentiable, telle que l'erreur quadratique entre la sortie du réseau et la sortie désirée :

$$E(n) = \|d(n) - x_L(n)\|^2 \quad (2.4)$$

Où

$x_L(n)$: est la sortie du réseau au temps n ,

$d(n)$:la sortie désirée au temps n .

L'algorithme de BP assure la descente du gradient sur le critère d'erreur pour atteindre le minimum. Le gradient de $E(n)$ est calculé pour tous les poids de la manière suivante :

$$\frac{\partial E(n)}{\partial W_{ijk}} = \frac{\partial E(n)}{\partial y_{ik}} \cdot \frac{\partial y_{ik}}{\partial W_{ijk}} = \frac{\partial E(n)}{\partial y_{ik}} \cdot x_{(i-1)j} \quad (2.5)$$

Dans le cas de la couche de sortie ($i = L$) on évalue le terme d'erreur de sortie δ_{Lk} comme suit :

$$\delta_{Lk} = \frac{\partial E(n)}{\partial y_{Lk}} = 2f'(y_{Lk})(d_k - x_{Lk}) \quad (2.6)$$

Où

$$f'(\) : \text{ est la dérivée de la fonction d'activation : } f'(x) = \frac{df(x)}{dx} .$$

Pour les couches cachées, le terme d'erreur δ_{ik} du neurone (i, k) est donné par :

$$\delta_{ik} = f'(y_{ik}) \sum_{j=1}^{N(i+1)} \delta_{(i+1)j} W_{(i+1)kj} \quad (2.7)$$

La modification des poids et des biais est obtenue selon les équations suivantes :

$$\begin{aligned} W_{ijk}(n+1) &= W_{ijk}(n) + \mu \delta_{ik} x_{(i-1)j} + \alpha (W_{ijk}(n) - W_{ijk}(n-1)) \\ b_{ik}(n+1) &= b_{ik}(n) + \mu \delta_{ik} \end{aligned} \quad (2.8)$$

Où

μ : est le pas d'apprentissage,

α : le terme d'inertie (momentum).

Le choix du pas d'apprentissage influe beaucoup sur la rapidité de convergence, un pas trop petit ralentit l'apprentissage, un pas trop important provoque un risque d'instabilité. Il est même possible que l'algorithme rencontre un minimum local.

La figure (2.10) représente une forme 3D d'une fonction de coût E ayant des minimum locaux et un minimum global, où sa projection sur le plan de base formé par deux variables (poids W_1 et W_2) donne un contour de plusieurs niveaux de la fonction de coût. La figure (2.11) représente la Minimisation de la fonction de coût E par la méthode du gradient.

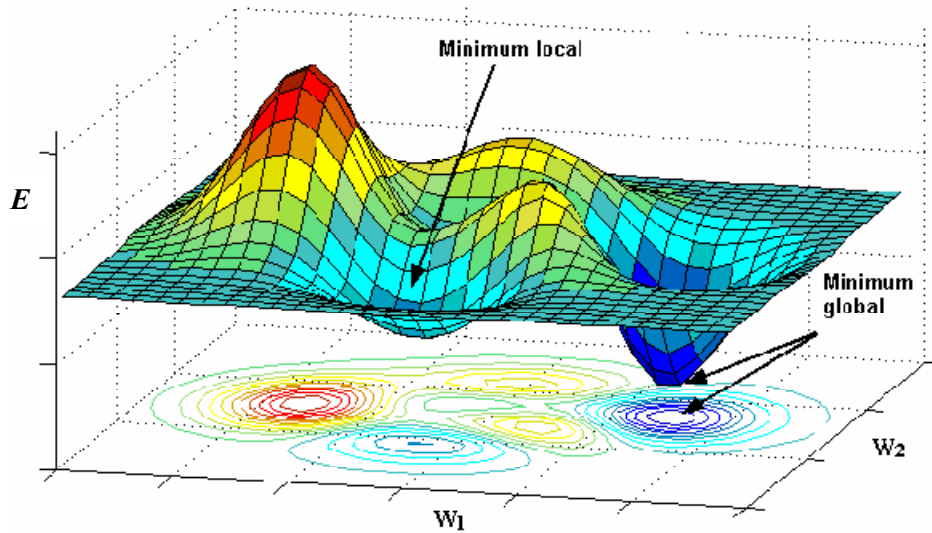


Figure (2.10) Représentation de la fonction de coût E d'un neurone à deux entrées pondérées W_1 et W_2 .

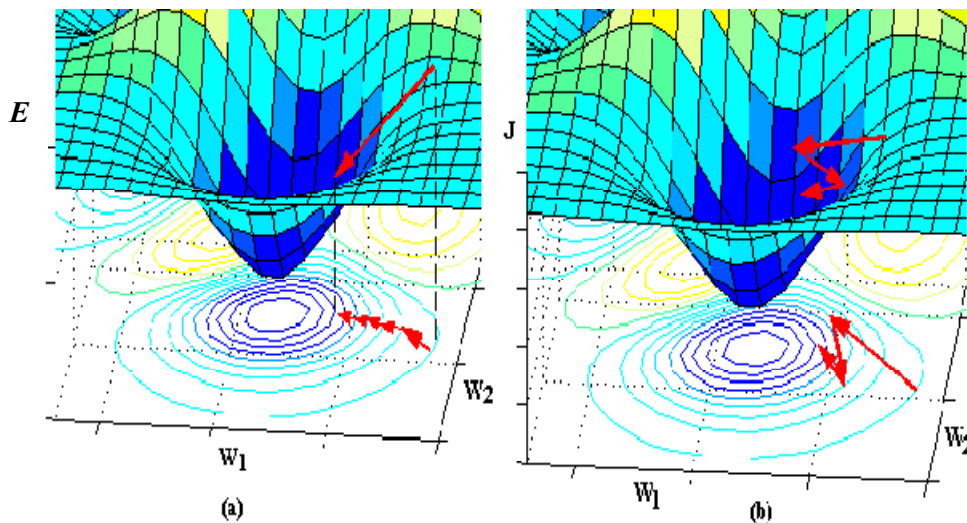


Figure (2.11) Minimisation de la fonction de coût E par la méthode du gradient.
 (a) Pas du gradient est petit, convergence lente, le minimum global peut être atteint.
 (b) Pas du gradient est grand, convergence rapide, le minimum global est rarement atteint.

Le terme d'inertie (momentum $0 < \alpha < 1.0$) permet de sortir du minimum local et de limiter les oscillations lors de l'apprentissage ; en prenant en compte les évolutions des étapes précédentes et donc de converger plus rapidement [22].

La phase d'apprentissage est souvent arrêtée lorsque l'erreur calculée sur l'ensemble de la base d'apprentissage est inférieure à un seuil déterminé par l'utilisateur. Il est possible d'arrêter l'apprentissage en fixant une limite au nombre d'itérations.

2.9 Algorithme de Levenberg-Marquardt

Parmi les algorithmes de la famille quasi-Newton, la méthode de Levenberg-Marquardt est un standard pour l'optimisation de l'erreur quadratique due à ses propriétés de convergence rapide et de robustesse. Elle s'appuie sur les techniques des moindres carrés non-linéaires et de l'algorithme de Gauss-Newton à voisinage restreint [7].

Le problème résolu par cet algorithme est le suivant : soit E une fonction, dépendant d'un vecteur de paramètres W . On ne connaît pas l'expression de E , mais on sait calculer numériquement quelques caractéristiques de cette fonction (valeur ou dérivée en un point par exemple). On cherche, numériquement également, le vecteur W qui minimise la quantité $E(W)$.

Dans le contexte des réseaux de neurones, $E(W)$ est la fonction de performance, généralement l'erreur quadratique moyenne calculée sur une base d'apprentissage, et W est un vecteur regroupant tous les poids du réseau. Un algorithme d'optimisation permet d'effectuer un apprentissage du réseau, c'est à dire de trouver les poids qui minimisent l'erreur quadratique moyenne.

L'algorithme ci-après ne trouvera qu'un minimum local de la fonction. Si $E(W)$ possède plusieurs minima locaux, celui vers lequel convergera l'algorithme dépend du point de départ choisi et des paramètres de l'algorithme. On ne peut garantir de converger vers un minimum global.

Une autre manière de diminuer le nombre d'itérations d'un algorithme d'optimisation est d'utiliser les dérivées secondes de E . En effet le gradient donne une direction vers laquelle se déplacer pour trouver le minimum, mais ne donne pas le pas. Dans la descente de gradient classique ce pas est un coefficient fixe, et dans la variante adaptative il peut varier à chaque itération. Mais la dérivée seconde de E est liée au rayon de courbure de la fonction, et permet donc de déterminer ce pas de manière plus fine [18]font partie de cet algorithme.

En effet si l'on suppose que E est une fonction quadratique :

$$E(W) = a + b^T W + W^T C W \quad (2.9)$$

Où

b^T est la transposée du vecteur b et C est une matrice symétrique.

On peut trouver l'extremum de la fonction, il s'agit du point auquel la dérivée de E s'annule

$$\nabla E = 0 \Rightarrow b + 2C W = 0 \quad (2.10)$$

Soit:

$$W = -2C^{-1} b \quad (2.11)$$

A condition que C soit inversible. Pour une fonction E quelconque, il est possible de l'approximer localement en un point W_i par une fonction quadratique, en utilisant ses dérivées première et seconde, et avec l'équation (2.11) déterminer le vecteur pour l'itération suivante d'un algorithme d'optimisation plus évolué que la descente de gradient. Mais le calcul des dérivées secondes peut être très long, tout d'abord parce que le nombre de dérivées secondes est le carré de celui des dérivées premières, et également parce que la dérivée seconde de E peut être assez complexe. De nombreux algorithmes, peut-être abusivement appelés algorithmes d'ordre 2, utilisent en fait une approximation des dérivées secondes calculées à partir de dérivées premières. Cependant ils gardent l'avantage de nécessiter beaucoup moins d'itérations qu'une descente de gradient.

L'algorithme de Levenberg-Marquardt [18] fait partie de ces algorithmes, et s'applique au cas particulier où E est une erreur quadratique moyenne. On peut donc l'exprimer sous la forme:

$$E(W) = \langle (g(x, W) - y)^2 \rangle \quad (2.12)$$

Où

g : désigne une fonction de deux vecteurs x et W ,

$\langle \dots \rangle$: désigne la moyenne calculée sur un ensemble de couples (x, y) .

L'on se place dans le cas où g est une fonction scalaire afin de simplifier la notation, mais la même démarche peut être faite si g est une fonction vectorielle. Dans la suite de cette section toutes les dérivées sont en fonction du vecteur W . C'est en effet uniquement ce vecteur que l'on fait varier afin de trouver le minimum de E .

On suppose, comme pour la descente de gradient, que l'on se trouve à une itération numéro i , et que l'on cherche à calculer un nouveau vecteur W_i en fonction de W_{i-1} , tel que $E(W_i)$ se rapproche plus d'un minimum local de E . Pour cela on calcule une approximation quadratique \hat{E} de E à partir d'une approximation linéaire \hat{g} de g autour du point W_{i-1} . En déterminant le point W auquel le gradient de \hat{E} s'annule, on obtient :

$$W = W_{i-1} - H^{-1} d \quad (2.13)$$

Avec:

$$d = \langle (g(x, W_{i-1}) - y) \nabla g(x, W_{i-1}) \rangle \quad (2.14)$$

$$H = \langle (g(x, W_{i-1}) - y) \nabla g(x, W_{i-1})^T \rangle$$

A condition que H soit inversible. La matrice H est une approximation du Hessian de E , calculée à partir du gradient de g . L'équation précédente pourrait servir dans un algorithme d'optimisation, qui permet de calculer W_i à partir de W_{i-1} au cours de l'itération i . Mais ceci n'est efficace en pratique que si g est effectivement proche d'une droite autour du point W_{i-1} . Dans le cas contraire cet algorithme donne de très mauvais résultats.

L'idée de Levenberg est donc d'utiliser cette approche quadratique dans les zones où g est quasi-linéaire, et une descente de gradient dans les autres cas. Le pas d'une itération de cet algorithme est calculé de la manière suivante :

$$W = W_{i-1} - (H + \lambda I)^{-1} d \quad (2.15)$$

Lorsque λ est faible, cette équation est équivalente à (2.13) et le nouveau vecteur de paramètres est déterminé avec l'approximation quadratique de E . Lorsque λ est grand, cette équation est équivalente à :

$$\begin{aligned} W_i &= W_{i-1} - \frac{1}{\lambda} d \\ &= W_{i-1} - \frac{1}{\lambda} \langle (g(x, W_{i-1}) - y) \nabla g(x, W_{i-1}) \rangle \\ &= W_{i-1} - \frac{1}{2\lambda} \nabla E(x, W_{i-1}) \end{aligned} \quad (2.16)$$

Ce qui correspond bien à une descente de gradient. Pour des valeurs intermédiaires de λ l'algorithme est un mélange entre la descente de gradient et l'approche quadratique basée sur l'approximation linéaire de g . Ce coefficient λ est modifié à chaque itération, comme pour la descente de gradient adaptative. Si $E(W_i)$ diminue au cours de l'itération, on diminue λ (en le divisant par 10 par exemple), et l'on se rapproche ainsi de la méthode quadratique. Au contraire si $E(W_i)$ augmente, cela signifie que nous nous trouvons dans une région dans laquelle g n'est pas très linéaire, et donc on augmente λ (en le multipliant par 10 par exemple) afin de se rapprocher de la descente de gradient.

Cet algorithme a ensuite été amélioré par Marquardt, le pas de l'itération étant défini cette fois par:

$$W_i = W_{i-1} - (H + \lambda \text{diag}(H))^{-1} d \quad (2.17)$$

La matrice identité a été remplacée par la diagonale de H . Le but est ici de modifier le comportement de l'algorithme dans les cas où λ est grand, c'est à dire lorsque l'on est proche d'une descente du gradient. Avec cette modification l'on se déplace plus vite dans les directions vers

lesquelles le gradient est plus faible, afin d'éviter de passer de nombreuses itérations sur un plateau. Ceci est appelé l'algorithme de Levenberg -Marquardt.

En pratique cet algorithme, en particulier dans le cas des réseaux de neurones, permet de converger avec beaucoup moins d'itérations. Mais chaque itération demande plus de calculs, en particulier pour l'inversion de la matrice H , et son utilisation se limite donc aux cas où le nombre de paramètres à optimiser n'est pas très élevé. En effet le nombre d'opérations nécessaires à l'inversion d'une matrice est proportionnel à N^3 , N étant la taille de la matrice, et ici également la taille du vecteur W .

2.9.1 Avantage de La méthode Levenberg-Marquardt

La méthode Levenberg-Marquardt est un condensé de deux techniques. En effet, cette méthode tend vers la méthode de Newton pour une valeur de μ_k petite mais est équivalente à la méthode du gradient simple pour un pas $\eta = \frac{1}{\mu_k}$ pour une valeur de μ_k grande. Le Hessien est toujours défini positif ce qui assure la convergence vers un minimum de la solution.

2.10 Le réseau à fonction de base radiale

Le réseau à fonction de base radiale RBF (Radial Basis Function) est basé sur une architecture qui s'organise en deux couches seulement ; une couche cachée et une couche de sortie [22] comme le montre la figure (2.12).

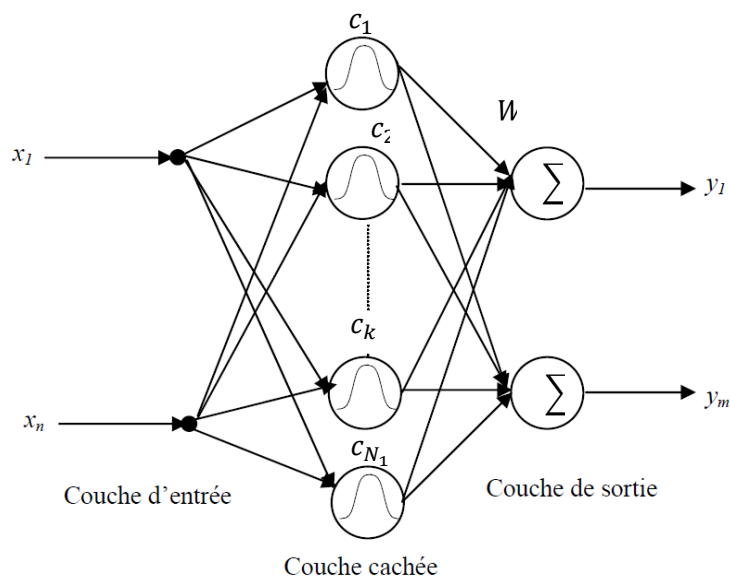


Figure (2.12) Architecture du réseau RBF

La couche cachée, constituée des noyaux (ou neurones) RBF effectue une transformation non linéaire de l'espace d'entrée. La couche de sortie calcule une combinaison linéaire des sorties de la couche cachée. Chaque noyau élémentaire calcule la distance entre l'entrée et son centre qu'il passe ensuite dans une non linéarité concrétisée par une fonction d'activation $\phi(\cdot)$ qui est généralement de type gaussienne [22]. La valeur que prend la sortie du noyau gaussien est d'autant plus importante que l'entrée est plus proche de son centre et tend vers zéro, lorsque la distance entrée centre devient importante. La sortie du réseau RBF est donnée par :

$$y_i = \sum_{k=1}^{N_1} W_{kj} \phi_k(\|x - c_k\|) \quad (2.18)$$

$$\phi(\xi) = \exp\left(\frac{-\xi^2}{2\eta^2}\right) \quad (2.19)$$

$\|\cdot\|$ dénote la norme euclidienne, x le vecteur d'entrée, c_k est le centre associé au noyau k . N_1 le nombre de noyaux de la couche cachée et W_{kj} les poids associés à la couche de sortie. Le paramètre ' η ' permet de contrôler la vitesse de décroissance de la fonction ϕ .

L'utilisation habituelle des RBF conserve une fonction d'activation linéaire en sortie mais l'utilisation d'une fonction non linéaire reste possible [22].

A l'instar du réseau MLP, le réseau RBF possède aussi la propriété d'approximation c'est à dire que le réseau est capable d'approximer n'importe quelle fonction continue avec une précision donnée pourvu qu'on fournisse un nombre de neurones suffisant [22].

L'apprentissage des réseaux RBF est composé d'une règle non supervisée pour l'adaptation des centres et une règle d'apprentissage supervisé pour l'adaptation des poids [Annexe A].

Le choix de la position des centres et le nombre de neurones reste généralement arbitraire.

2.11 Comparaison entre le réseau MLP et le réseau RBF

Les réseaux de neurones de type MLP et RBF sont capables d'approximer n'importe quelle fonction non linéaire ; la différence porte sur l'architecture qui est figée en deux couches pour les RBF et peut comporter plusieurs couches pour le MLP.

Le réseau RBF souffre d'une difficulté dans le calcul de la distribution optimale des centres. Si les centres ne sont pas identifiés avec une grande précision, il se produira des dégradations dans les performances du système [22]. En outre, la complexité de calcul augmente exponentiellement avec le nombre d'entrée, augmentant ainsi la difficulté du problème [22]. Cependant l'addition d'une couche cachée supplémentaire peut améliorer la généralisation du MLP; il peut aussi réduire

le nombre de neurones requis et, donc, être une solution plus désirable [22].

2.12 Mise en œuvre d'algorithme d'apprentissage du RNA

Nous allons présentés dans cette section, les différents aspects de la mise en œuvre algorithmique pour expliquer les spécifications des outils d'apprentissage et de généralisation de réseau de neurones.

Nous avons posé précédemment le problème de l'apprentissage par réseau de neurones comme un problème d'optimisation d'une fonction de coût qui représente le but de l'apprentissage. L'algorithme de rétropropagation du gradient est parmi le plus utilisé dans le cas d'un problème de classification supervisée, l'apprentissage de réseau de neurones par cet algorithme consiste à [18] :

- Choisir un couple (entrée, sortie désirée) ;
- Initialisation aléatoire des poids et des biais ;
- Calculer la sortie actuelle suivant la fonction d'activation choisie (on prend une fonction non-linéaire dans le cas de l'application en classification)
- Calcul de l'erreur (la fonction coût) ;
- Si le réseau prend une décision correcte, les poids restent inchangés ;
- Sinon, rétropropager l'erreur, après avoir calculer le terme d'adaptation des poids (mise à jour des poids)
- Présenter les paramètres (poids et biais) pour une nouvelle itération jusqu'à que les coefficients synaptiques se stabilisent autour d'une valeur et l'erreur quadratique totale du réseau soit inférieur à un seuil.

En plus, il est possible d'arrêter l'apprentissage en fixant une limite au nombre d'itérations, généralement le pas d'apprentissage et le monmomentum doivent être adapté quand le nombre d'itération augmente. Les détails mathématiques de la phase d'apprentissage de réseau de neurone par l'algorithme de rétropropagation sont décrits dans le paragraphe (2.8).

Les paramètres d'entrée du programme d'apprentissage sont les suivantes :

- Base de données. Vecteurs d'entrée et la classe correspondante ;
- Les poids, les biais initiaux ;
- La fonction d'activation ;
- Le nombre d'itérations.

La structure générale du programme d'apprentissage de l'algorithme de rétropropagation du gradient suit les étapes suivantes :

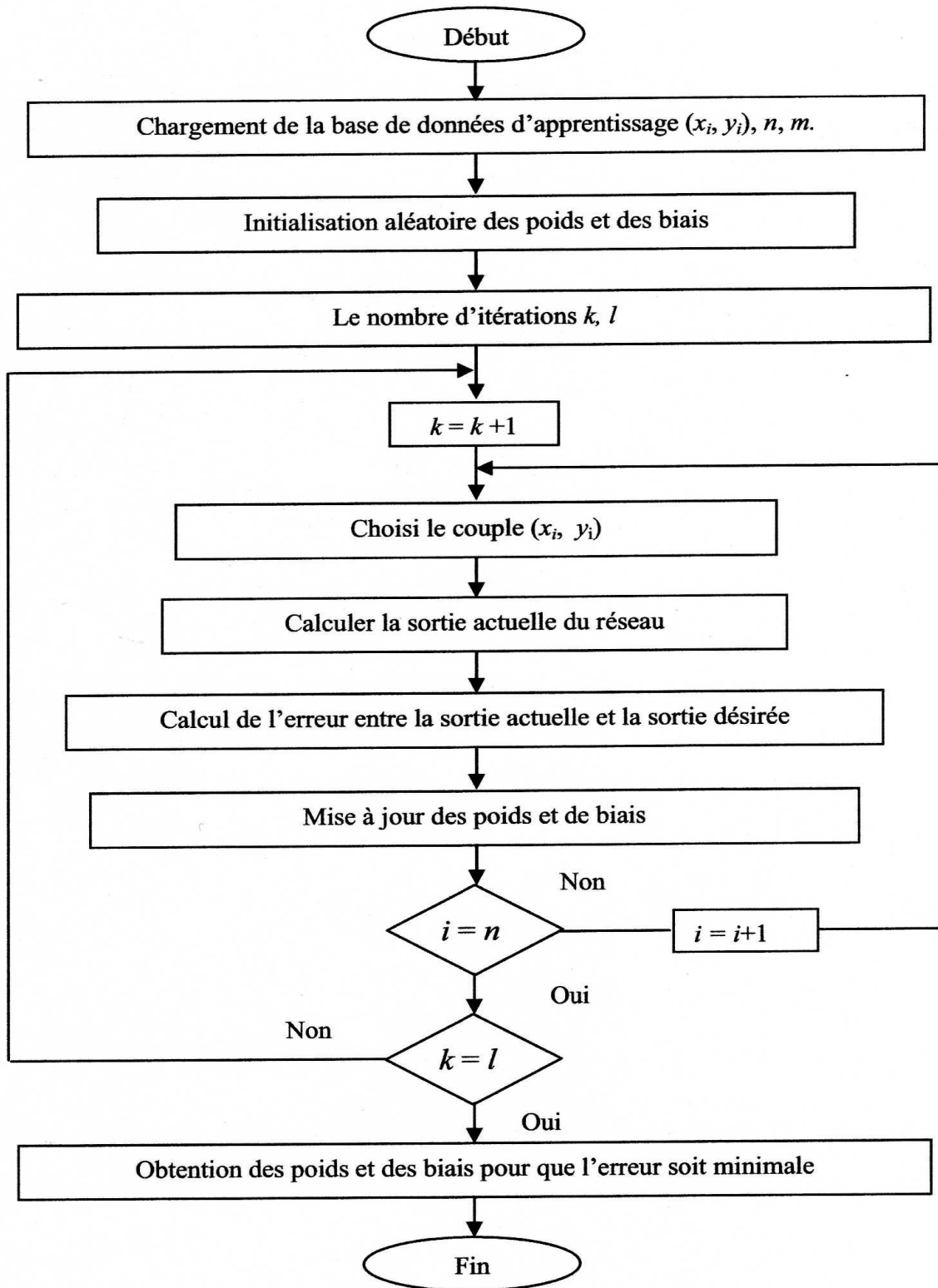


Figure (2.13) Structure générale du programme d'apprentissage par réseau de neurones [18]

2.13 Généralisation d'un réseau MLP (Validation croisée)

L'un des nombreux intérêts des réseaux MLP pour l'approximation de fonctions est leur capacité à généraliser, c'est à dire leur capacité à donner une bonne réponse à une donnée qui n'appartient pas à la base d'apprentissage qui a servi à l'éduquer.

Cette capacité à traiter des données non connues de la base d'apprentissage, la faculté de généralisation, permet de faire de l'interpolation ou de l'extrapolation.

Pour estimer la qualité d'un réseau de neurones on fait généralement appel à la racine carrée de l'erreur quadratique moyenne .

$$MSE_i = \sqrt{\frac{1}{p} \sum_{i=1}^p (y_i - \hat{y}_i)^2} \quad (2.20)$$

Avec :

p : Le nombre d'échantillons dans la base d'apprentissage.

y_i : La composante i de la sortie désirée.

\hat{y}_i : La composante estimée (calculée) du neurone i de la couche de sortie du réseau MLP.

Pour analyser le comportement du Réseau de Neurones, on observe donc indépendamment ces deux quantités y_i et \hat{y}_i respectivement. L'indicateur d'un bon réseau sera le taux de généralisation : c'est-à-dire avoir le même MSE mais sur une base d'exemples autre que celle utilisée durant l'apprentissage (la base de test ou de généralisation).

C'est à travers la valeur du MSE que l'on peut connaître la capacité de généralisation d'un réseau de neurones face à des données hors de la base d'apprentissage. Malheureusement, ce taux de généralisation n'est pas accessible directement, on cherche donc à l'estimer. Pour cela, la base de données est partagée en deux :

- la base d'apprentissage sur laquelle on configure le réseau pour réaliser au mieux la tâche qui lui est attribuée.
- La base de généralisation sur laquelle on teste l'aptitude du réseau à bien traiter des données hors de la base d'apprentissage.

Nous avons évoqué plusieurs fois le problème du surapprentissage, qui est provoqué par la capacité d'un réseau de neurones, possédant un nombre d'unités de mémorisation plus que

nécessaire (on parle aussi de ‘ surparamétrisation ’), à apprendre parfaitement les exemples de la base d’apprentissage.

Typiquement, l’évolution de l’erreur quadratique sur la base d’apprentissage en fonction du nombre de cycles d’apprentissage, se comporte comme la courbe A de la figure (2.14). Sur cette même figure est également représentée l’évolution de l’erreur quadratique en généralisation calculée à partir d’une base de données différente de la base d’apprentissage (courbe B).

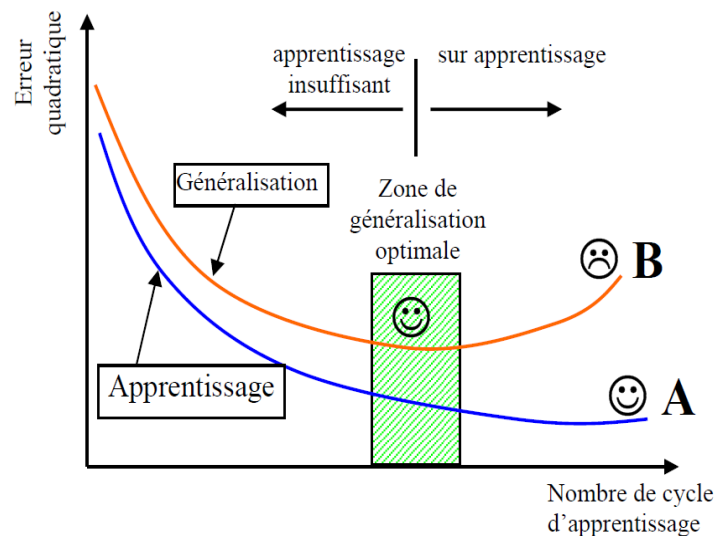


Figure (2.14) Evolution de l’erreur quadratique de l’apprentissage et de généralisation.

Afin d’arrêter l’apprentissage juste avant que ne se produise ce phénomène de surapprentissage, plusieurs méthodes ont été proposées.

La plus simple consiste à disposer de trois bases de données distinctes : une base d’apprentissage, une base de test et une base dite de « validation croisée » [15].

Cette dernière base est utilisée pendant l’apprentissage afin d’examiner le comportement du réseau pour des données qui lui sont inconnues. Ainsi, l’apprentissage est arrêté lorsque l’erreur sur cette courbe B (base de validation croisée) atteint un minimum .

Notons que cette technique nécessite d’avoir suffisamment de données pour constituer trois bases à la fois représentatives et distinctes. Dans le cas où on ne disposerait pas de suffisamment de données, une technique a été développée pour n’utiliser que la base d’apprentissage ; elle consiste à effectuer deux apprentissages successifs.

Pour le premier apprentissage, la base d’apprentissage est divisée en deux de manière à constituer une plus petite base d’apprentissage et une base de validation croisée.

Le nombre de cycles d'apprentissage nécessaire pour parvenir à la zone de généralisation optimale est conservé. Ce nombre est ensuite utilisé pour stopper l'apprentissage avec la base d'apprentissage globale.

2.14 Les champs d'application des RNA

Les propriétés d'apprentissage et d'approximation des réseaux de neurones ont permis une importante application de ces derniers dans les différents domaines pratiques, notamment, les domaines suivants : [11]

- Identification des processus
- Contrôle des processus
- Classification des signaux
- Mémoires associatives
- ...etc.

2.15 Les limitations d'un réseau de neurones [9]

2.15.1 Avantages

- Implémentation du parallélisme.
- Apprentissage.
- Robustesse: données bruitées ou incomplètes.
- Généralisation à des Modèles similaires.
- Trouve des solutions aux problèmes non linéaires.
- Trouve des solutions aux problèmes qui n'ont pas une modélisation.

2.15.2 Inconvénients

- N'ont pas encore expliqué le fonctionnement du cerveau.
- Les poids ne sont pas interprétables.
- L'apprentissage n'est pas toujours évident.
- Ne sont pas extensibles (l'ajout d'un neurone).

2.16 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.

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.

Pour notre système, nous avons opté pour un réseau de neurone multicouche car il semble bien adapté au problème posé .

Chapitre 3

Modélisation de l'onduleur triphasé cascadié à sept niveaux

3.1 Introduction

Le domaine de la conversion statique n'a cessé de proposer des évolutions, non seulement au niveau des composants mais aussi sur la manière de les associer. L'association des composants de puissance a suscité un grand intérêt de la communauté scientifique faisant apparaître les onduleurs multiniveaux.

Deux motivations principales sont à l'origine des onduleurs multiniveaux, à savoir d'une part l'augmentation de la puissance par le biais de la génération de tensions plus élevées, au-delà de celles compatibles avec les tensions de blocage des composants de puissance. D'autre part, on cherche à obtenir des grandeurs de sortie ayant un meilleur contenu harmonique [4],[14].

Les onduleurs multiniveaux, de plus en plus utilisés dans les applications d'entraînement à vitesse variable, s'avèrent très utiles pour la commande des machines électriques[4]. En effet, l'amélioration en termes de contenu harmonique des tensions de sortie de l'onduleur améliore considérablement les performances de la machine [4],[7],[14]. En outre, les formes d'onde en échelon des onduleurs multiniveaux vont réduire d'une façon notable la contrainte liée à la très haute rapidité de montée des impulsions de tension appliquées à la machine, ce qui augmente la sûreté de fonctionnement et réduit les radiations électromagnétiques[4].

Dans ce chapitre, nous proposons une étude de l'onduleur cascadié à sept niveaux. Cet onduleur permet d'échelonner la tension continue sur sept niveaux différents. On aboutit ainsi à une tension de forte valeur à partir des dispositifs à semi-conducteurs à tension réduite. Nous

commençons par la modélisation de l'onduleur triphasé cascadié de tension à sept niveaux . Une simulation de cet onduleur sera présentée dans la dernière partie de ce chapitre.

3.2 Structure de l'onduleur triphasé cascadié à sept niveaux

La figure (3.1) représente la structure de l'onduleur triphasé cascadié à sept niveaux . Un bras d'un tel onduleur comprend huit transistors avec leurs diodes antiparallèles. Chaque transistor est dimensionné de façon à bloquer la tension aux bornes d'une des sources du côté continu. Cette caractéristique permet l'augmentation de la tenue en tension de l'onduleur.

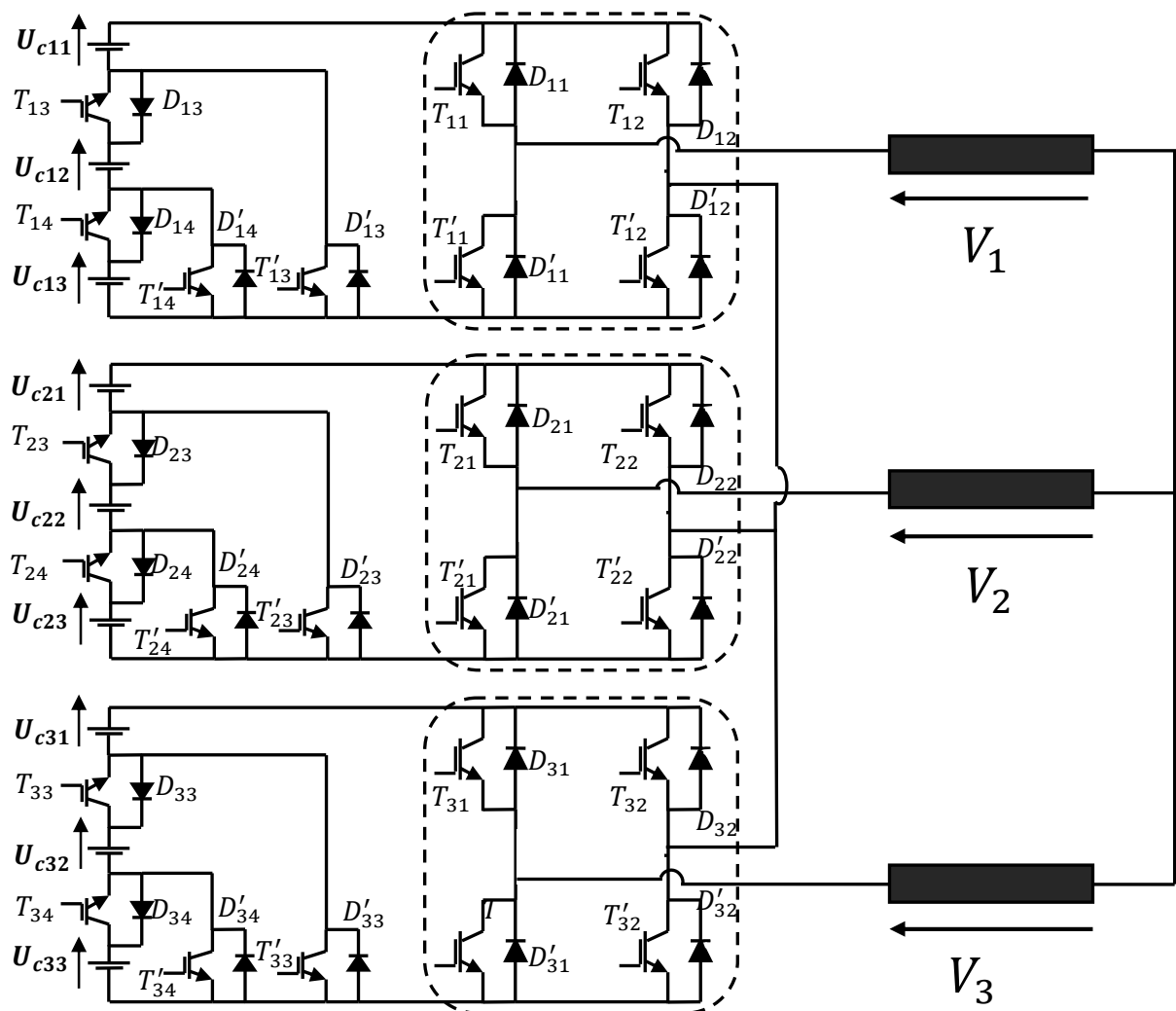


Figure (3.1) Structure de l'onduleur triphasé cascadié à sept niveaux

3.3 Modélisation aux valeurs instantanées de l'onduleur triphasé cascadié à sept niveaux

La modélisation permet d'étudier l'évolution des différentes grandeurs régissant le fonctionnement de l'onduleur triphasé cascadié à sept niveaux tout en considérant les commutations des interrupteurs.

3.3.1 Hypothèses

Afin de simplifier l'étude, les hypothèses retenues pour la mise en équations sont les suivantes :

- Chaque paire transistor-diode est représentée par un interrupteur bidirectionnel,
- Les interrupteurs sont idéalisés (tension de saturation, courant de fuite et temps de commutation sont supposés nuls),
- Les interrupteurs étant parfaits, les temps morts sont supposés nuls,
- Les sources de tension sont supposées parfaites,
- Les transitions entre les configurations ne dépendent que des commandes des transistors.

3.3.2 Commande complémentaire

Afin d'éviter de laisser en circuit ouvert la source de courant (ouverture de tous les interrupteurs) ou la court-circuit de la source de tension (fermeture de tous les interrupteurs), une commande complémentaire doit être adoptée [4],[12]. Si K_{bi} représente la fonction de connexion de l'interrupteur T_{bi} , définie par $K_{bi} = 1$ lorsque cet interrupteur est fermé et $K_{bi} = 0$ lorsqu'il est ouvert, la commande complémentaire retenue est caractérisée par une relation liant les fonctions de connexion des interrupteurs d'un bras b comme suit :

$$K'_{bi} = 1 - K_{bi}, \quad i = 1, \dots, 4, \quad b = 1, 2, 3 \quad (3.1)$$

Le tableau (3.1) donne les états possibles d'un bras b tout en tenant compte de la commande complémentaire choisie. L'état 1 signifie que l'interrupteur correspondant est fermé et l'état 0 qu'il est ouvert. Ce tableau montre que cette commande complémentaire permet d'exploiter les sept niveaux de tension de sortie possibles pour un bras de l'onduleur.

Tableau (3.1) Etats de l'onduleur à sept niveaux et sa tension entre deux points A et M

K_{b1}	K_{b2}	K_{b3}	K_{b4}	K'_{b4}	K'_{b3}	K'_{b2}	K'_{b1}	V_{bAM}
1	0	1	1	0	0	1	0	$U_{cb1} + U_{cb2} + U_{cb3}$
1	0	1	0	1	0	1	0	$U_{cb1} + U_{cb2}$
1	0	0	0	1	1	1	0	U_{cb1}
0	0	0	0	1	1	1	1	0
0	1	0	0	1	1	0	1	$-U_{cb1}$
0	1	1	0	1	0	0	1	$-U_{cb1} - U_{cb2}$
0	1	1	1	0	0	0	1	$-U_{cb1} - U_{cb2} - U_{cb3}$

3.3.3 Différentes configurations d'un bras de l'onduleur triphasé cascadié à sept niveaux

La topologie de l'onduleur fournit à la sortie de chacune de ses phases une tension entre deux points A et M possédant sept niveaux différents. Le nombre de niveaux de tension d'un onduleur représente le nombre de potentiels différents de l'étage continu qu'il est possible d'imposer à la tension de sortie. Cependant, il est à noter que chaque niveau de tension n'est obtenu que par l'intermédiaire d'une unique configuration de cet onduleur. la tension de sortie V_{bAM} donnée par le tableau (3.2) est une tension à sept niveaux.

Tableau (3.2) Tension V_{bAM} relative à chaque configuration

Configuration	Tension V_{bAM}
E_1	$U_{cb1} + U_{cb2} + U_{cb3}$
E_2	$U_{cb1} + U_{cb2}$
E_3	U_{cb1}
E_4	0
E_5	$-U_{cb1}$
E_6	$-U_{cb1} - U_{cb2}$
E_7	$-U_{cb1} - U_{cb2} - U_{cb3}$

Les configurations possibles en conduction continue sont regroupées dans la figure (3.2).

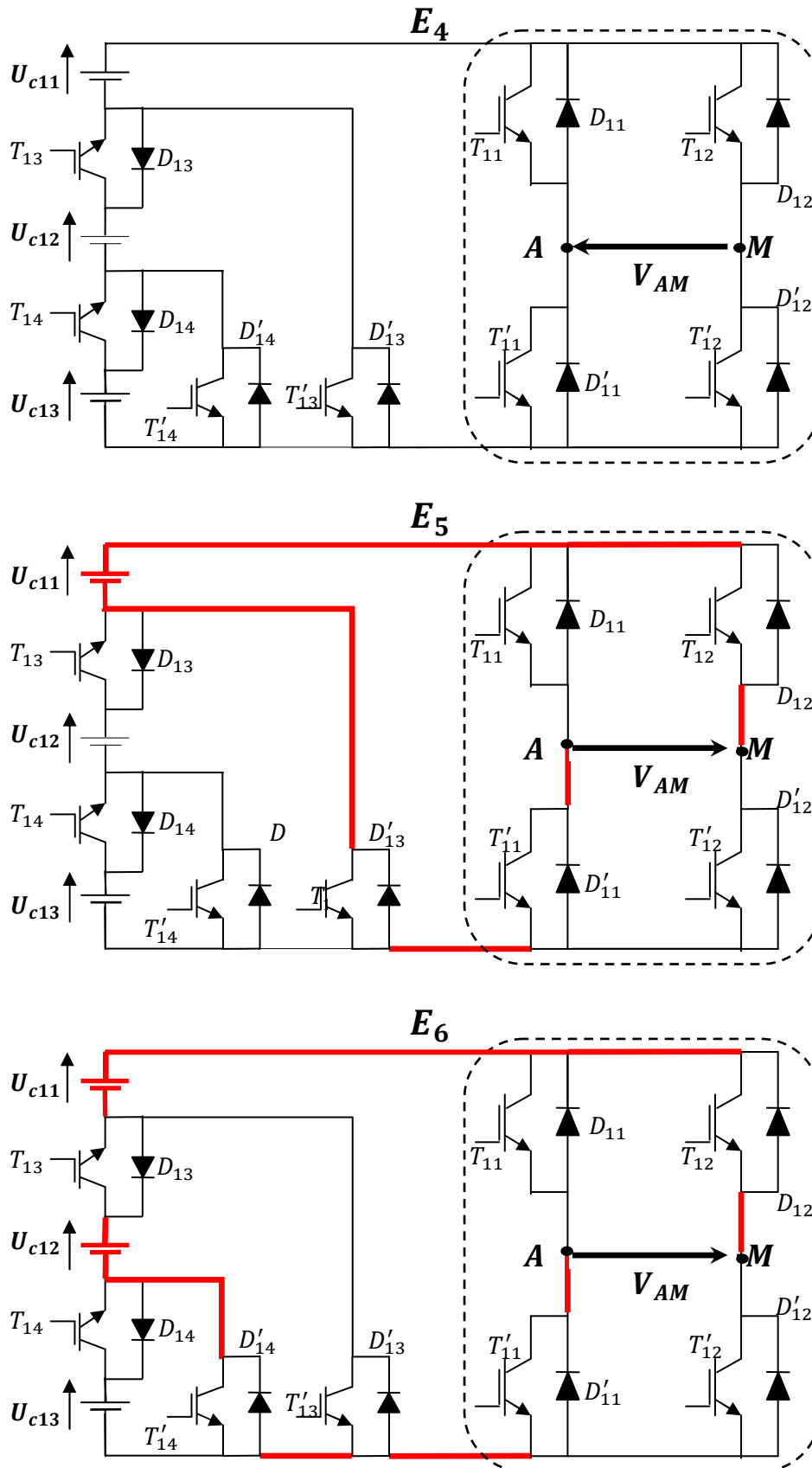


Figure (3.2) Configurations possibles d'un bras de l'onduleur triphasé cascadié à sept niveaux

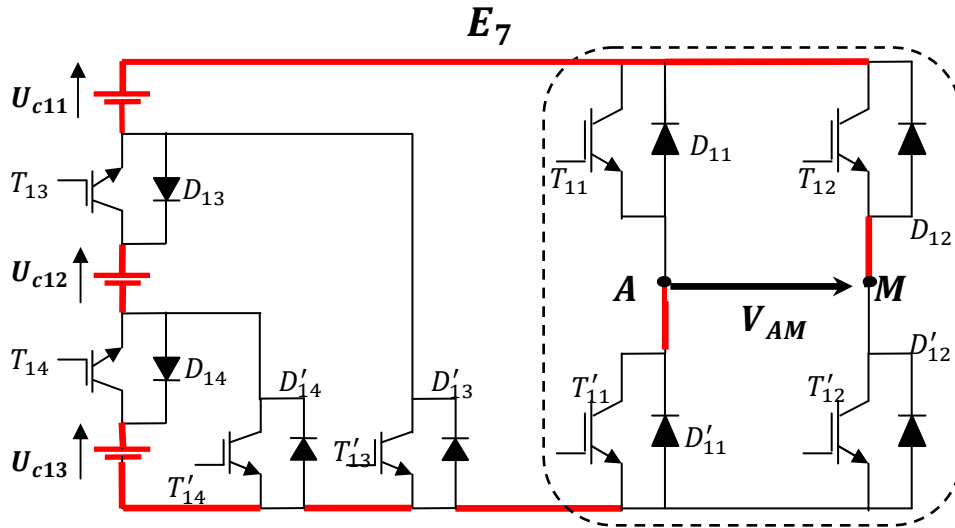


Figure (3.2) Configurations possibles d'un bras de l'onduleur triphasé cascadié à sept niveaux (suite)

3.3.4 Mise en équations

Admettons que l'onduleur est associé à une charge triphasée équilibrée et connectée en étoile. Le point neutre de la charge est supposé non connecter.

Dans le cas où les sources continues sont identiques nous posons:

$$\begin{aligned}
 U_{cb1} &= U_{cb1} = U_{cb1} = U_{c1} \\
 U_{cb2} &= U_{cb2} = U_{cb2} = U_{c2} \\
 U_{cb3} &= U_{cb3} = U_{cb3} = U_{c3}
 \end{aligned}
 \tag{3.2}$$

Le potentiel du noeud b ($b = 1,2,3$) l'onduleur triphasé cascadié à sept niveaux entre deux points A et M est donné par l'équation suivante :

$$\begin{aligned}
 V_{bAM} &= K_{b1} K_{b3} K_{b4} (1 - K_{b2}) (U_{c1} + U_{c2} + U_{c3}) + K_{b1} K_{b3} (1 - K_{b2}) (1 - K_{b4}) (U_{c1} + U_{c2}) \\
 &+ K_{b1} (1 - K_{b2}) (1 - K_{b3}) (U_{c1}) - K_{b2} K_{b3} K_{b4} (1 - K_{b1}) (U_{c1} + U_{c2} U_{c3}) \\
 &- K_{b2} K_{b3} (1 - K_{b1}) (1 - K_{b4}) (U_{c1} + U_{c2}) - K_{b2} (1 - K_{b1}) (1 - K_{b3}) (U_{c1})
 \end{aligned}
 \tag{3.3}$$

Afin de simplifier l'équation (3.3), on définit les fonctions de connexion des demi-bras K_b^t et K_b^b associées respectivement au demi-bras du haut et du bas. Pour un bras b , K_b^t et K_b^b s'expriment à l'aide des fonctions de connexion des interrupteurs comme suit :

$$\begin{aligned}
 K_b^t &= K_{b1} K_{b3} K_{b4} (1 - K_{b2}) \\
 K_b^b &= K_{b2} K_{b3} K_{b4} (1 - K_{b1})
 \end{aligned}
 \tag{3.4}$$

Posons aussi:

$$\begin{aligned}
K_b^{t1} &= K_{b1} K_{b3} (1 - K_{b2}) (1 - K_{b4}) \\
K_b'^{t1} &= K_{b2} K_{b3} (1 - K_{b1}) (1 - K_{b4}) \\
K_b^{t2} &= K_{b1} (1 - K_{b2}) (1 - K_{b3}) \\
K_b'^{t2} &= K_{b2} (1 - K_{b1}) (1 - K_{b3})
\end{aligned} \tag{3.5}$$

En introduisant (3.4) et (3.5) dans l'équation (3.3) on obtient:

$$\begin{aligned}
V_{bAM} &= K_b^t (U_{c1} + U_{c2} + U_{c3}) + K_b^{t1} (U_{c1} + U_{c2}) + K_b^{t2} U_{c1} \\
&\quad - K_b'^t (U_{c1} + U_{c2} + U_{c3}) - K_b'^{t1} (U_{c1} + U_{c2}) - K_b'^{t2} U_{c1}
\end{aligned} \tag{3.6}$$

L'équation (3.6) peut être réécrite sous la forme suivante:

$$\begin{aligned}
V_{bAM} &= (K_b^t K_b^{t1} K_b^{t2}) U_{c1} + K_b^t K_b^{t1} U_{c2} + K_b^t U_{c3} \\
&\quad - (K_b'^t K_b'^{t1} K_b'^{t2}) U_{c1} - (K_b'^t K_b'^{t1}) U_{c2} - K_b'^t U_{c3}
\end{aligned} \tag{3.7}$$

Cette équation montre qu'un onduleur à sept niveaux est une mise en série de six onduleurs à deux niveaux.

Les tensions composées de l'onduleur à sept niveaux sont exprimées à l'aide des fonctions de connexion des interrupteurs comme suit:

$$\begin{aligned}
U_{12} &= V_{1AM} - V_{2AM} = (K_1^t - K_2^t) (U_{c1} + U_{c2} + U_{c3}) + (K_1^{t1} - K_2^{t1}) (U_{c1} + U_{c2}) + (K_1^{t2} - K_2^{t2}) U_{c1} \\
&\quad - (K_1'^t - K_2'^t) (U_{c1} + U_{c2} + U_{c3}) - (K_1'^{t1} - K_2'^{t1}) (U_{c1} + U_{c2}) - (K_1'^{t2} - K_2'^{t2}) U_{c1} \\
U_{23} &= V_{2AM} - V_{3AM} = (K_2^t - K_3^t) (U_{c1} + U_{c2} + U_{c3}) + (K_2^{t1} - K_3^{t1}) (U_{c1} + U_{c2}) + (K_2^{t2} - K_3^{t2}) U_{c1} \\
&\quad - (K_2'^t - K_3'^t) (U_{c1} + U_{c2} + U_{c3}) - (K_2'^{t1} - K_3'^{t1}) (U_{c1} + U_{c2}) - (K_2'^{t2} - K_3'^{t2}) U_{c1} \\
U_{31} &= V_{3AM} - V_{1AM} = (K_3^t - K_1^t) (U_{c1} + U_{c2} + U_{c3}) + (K_3^{t1} - K_1^{t1}) (U_{c1} + U_{c2}) + (K_3^{t2} - K_1^{t2}) U_{c1} \\
&\quad - (K_3'^t - K_1'^t) (U_{c1} + U_{c2} + U_{c3}) - (K_3'^{t1} - K_1'^{t1}) (U_{c1} + U_{c2}) - (K_3'^{t2} - K_1'^{t2}) U_{c1}
\end{aligned} \tag{3.8}$$

Les tensions simples sont liées aux tensions composées par la relation suivante :

$$\begin{aligned}
V_1 &= \frac{1}{3} (U_{12} - U_{31}) \\
V_2 &= \frac{1}{3} (U_{23} - U_{12}) \\
V_3 &= \frac{1}{3} (U_{31} - U_{23})
\end{aligned} \tag{3.9}$$

En remplaçant (3.8) dans (3.9), il vient

$$\begin{aligned}
\begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} &= \frac{1}{3} \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix} \left\{ \begin{pmatrix} K_1^t & K_1^{t1} & K_1^{t2} \\ K_2^t & K_2^{t1} & K_2^{t2} \\ K_3^t & K_3^{t1} & K_3^{t2} \end{pmatrix} U_{c1} + \begin{pmatrix} K_1^t & K_1^{t1} \\ K_2^t & K_2^{t1} \\ K_3^t & K_3^{t1} \end{pmatrix} U_{c2} + \begin{pmatrix} K_1^t \\ K_2^t \\ K_3^t \end{pmatrix} U_{c3} \right. \\
&\quad \left. - \begin{pmatrix} K_1^t & K_1^{t1} & K_1^{t2} \\ K_2^t & K_2^{t1} & K_2^{t2} \\ K_3^t & K_3^{t1} & K_3^{t2} \end{pmatrix} U_{c1} - \begin{pmatrix} K_1^t & K_1^{t1} \\ K_2^t & K_2^{t1} \\ K_3^t & K_3^{t1} \end{pmatrix} U_{c2} - \begin{pmatrix} K_1^t \\ K_2^t \\ K_3^t \end{pmatrix} U_{c3} \right\} \quad (3.10)
\end{aligned}$$

3.4 Stratégie de commande de l'onduleur

Nous avons montré auparavant que l'onduleur à sept niveaux est équivalent à la mise en série de six onduleurs à deux niveaux. De ce fait, les stratégies de commande développées pour ces derniers peuvent être étendues facilement à l'onduleur à sept niveaux.

La qualité de la tension de sortie d'un onduleur dépend fortement de la technique de commande utilisée. De nombreuses stratégies de commande ont été proposées pour les onduleurs à niveaux multiples et la technique de stratégie d'élimination d'harmoniques est l'une des techniques les plus répandues .

Dans cette section , la méthode d'élimination d'harmoniques sera discutée brièvement.

3.4.1 Commande de l'onduleur à sept niveaux par la stratégie d'élimination d'harmoniques

L'idée de cette stratégie a été introduite pour la première fois par *Turnbull* en 1967, puis développée par *Patel* et *Hoft* en 1973 [14]. Son principe consiste d'abord à formuler l'expression générale de l'amplitude des harmoniques, en se basant sur le développement en séries de Fourier. L'expression obtenue est une fonction des angles α_i de commutation. Ensuite, un système d'équations algébriques non linéaires est obtenu, en imposant la valeur désirée du fondamental et en annulant certains harmoniques. La résolution de ce système non linéaire permet de déterminer les angles α_i , par conséquent les instants de commande des interrupteurs à semi – conducteurs.

Dans cette section nous développons uniquement la modulation par paliers et la modulation par élimination sélective des harmoniques qui sont introduire dans la stratégie d'élimination d'harmoniques .

3.4.1.1 Modulation par paliers

Dans la modulation par paliers appelée OHSW (Optimized Harmonic Stepped-Waveform), la faible fréquence de commutation des interrupteurs de l'onduleur permet de surmonter le

problème des pertes par commutation, aussi bien le problème d'interférence électromagnétique ce qui a comme conséquence l'amélioration du rendement [4].

La tension d'une phase, entre deux points A et M , est représentée à la figure (3.3). Il s'agit d'une tension de forme en escalier constituée donc par des paliers non modulés. Cette tension est impaire et elle présente une symétrie par rapport au quart et à la demi période. De ce fait, sa décomposition en série de Fourier ne contient que les harmoniques impaires en sinus.

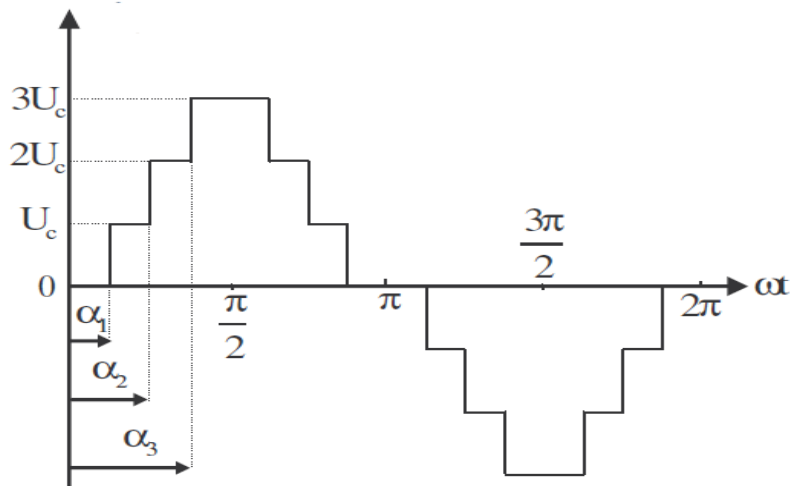


Figure (3.3) Tension V_{AM} de l'onduleur à sept niveaux commandé par la technique OHSW

L'amplitude de l'harmonique de rang n est donnée par l'expression suivante :

$$h_n = \begin{cases} \frac{4 U_c}{\pi n} \sum_{k=1}^3 \cos(n \alpha_k) & \text{si } n \text{ est impair} \\ 0 & \text{si } n \text{ est pair} \end{cases} \quad (3.11)$$

Les angles de commutation sont calculés de manière à répondre à certains critères portant sur l'amélioration de la tension de sortie de l'onduleur. Les critères usuellement retenus sont [4],[7][14]:

- Elimination d'un certain nombre d'harmoniques de rangs spécifiés,
- Minimisation de la distorsion harmonique totale,
- Maximisation de l'amplitude du fondamental.

Où

Le taux de distorsion harmonique (Total Harmonic Distorsion : THD) est une mesure qui permet de déterminer l'altération d'un signal par rapport à un signal purement sinusoïdal. Le THD est défini comme suit :

$$THD\% = \frac{\sqrt{\sum_{h=2}^{N_h} V_h^2}}{V_f} \times 100 \quad (3.12)$$

Avec

V_f : est la valeur efficace du fondamental ;

V_h : est la valeur efficace de l'harmonique d'ordre h ;

N_h : est le nombre d'harmoniques retenues. Dans cette étude N_h est fixé à 100.

3.4.1.2 Modulation par élimination sélective des harmoniques

Cette stratégie, dite SHEM (Selective Harmonic Elimination Modulation), est basée sur l'introduction des encoches (coupures) dans la forme escalier de la tension V_{AM} . La forme qui en résulte permet l'élimination d'un certain nombre d'harmoniques, tout dépend du nombre d'encoches introduites. La figure (3.4) représente un motif de la tension V_{AM} parmi plusieurs qui peuvent être imaginés [4].

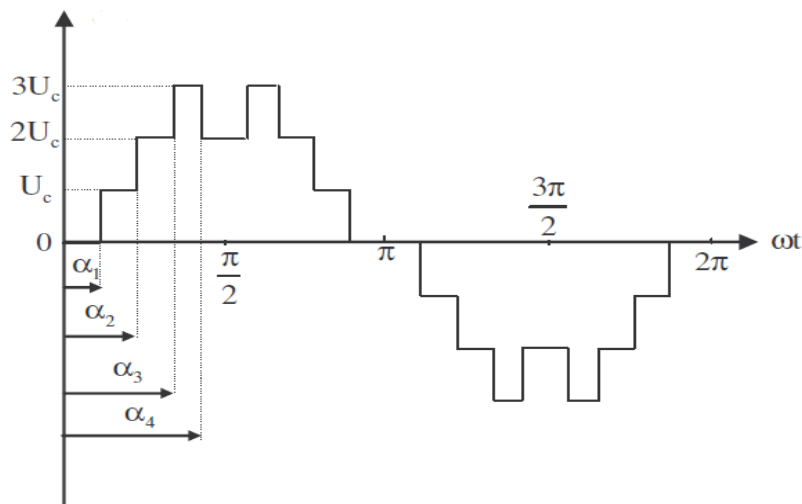


Figure (3.4) Tension V_{AM} de l'onduleur à sept niveaux commandé par la technique SHEM

L'expression de l'amplitude de l'harmonique de rang n est fonction du motif choisi, dans ce cas, l'amplitude du n^{eme} harmonique est donnée par:

$$h_n = \frac{4 U_c}{\pi n} (\cos(\alpha_1) + \cos(\alpha_2) + \cos(\alpha_3) - \cos(\alpha_4)) \text{ avec } n \text{ impair} \quad (3.13)$$

Les (c) angles de commutation ($\alpha_1, \dots, \alpha_c$) dans un motif donné sont déterminés soit en annulant les (c) premiers harmoniques, soit en imposant l'amplitude du fondamental et en annulant les ($c-1$) premiers harmoniques. La suppression des harmoniques de faibles fréquences

est faite en tenant compte que le troisième harmonique et ses multiples seront annulés dans un système triphasé [4],[14].

3.5 Résultats de simulation

Pour les simulations réalisées par la suite, on admet que les neuf sources de tension continue sont identiques de valeurs 73.33V.

La figure (3.5) représente la tension V_{AM} de la première bras et les signaux de commande des interrupteurs de ce bras et la figure (3.6) représente la tension simple de la première phase.

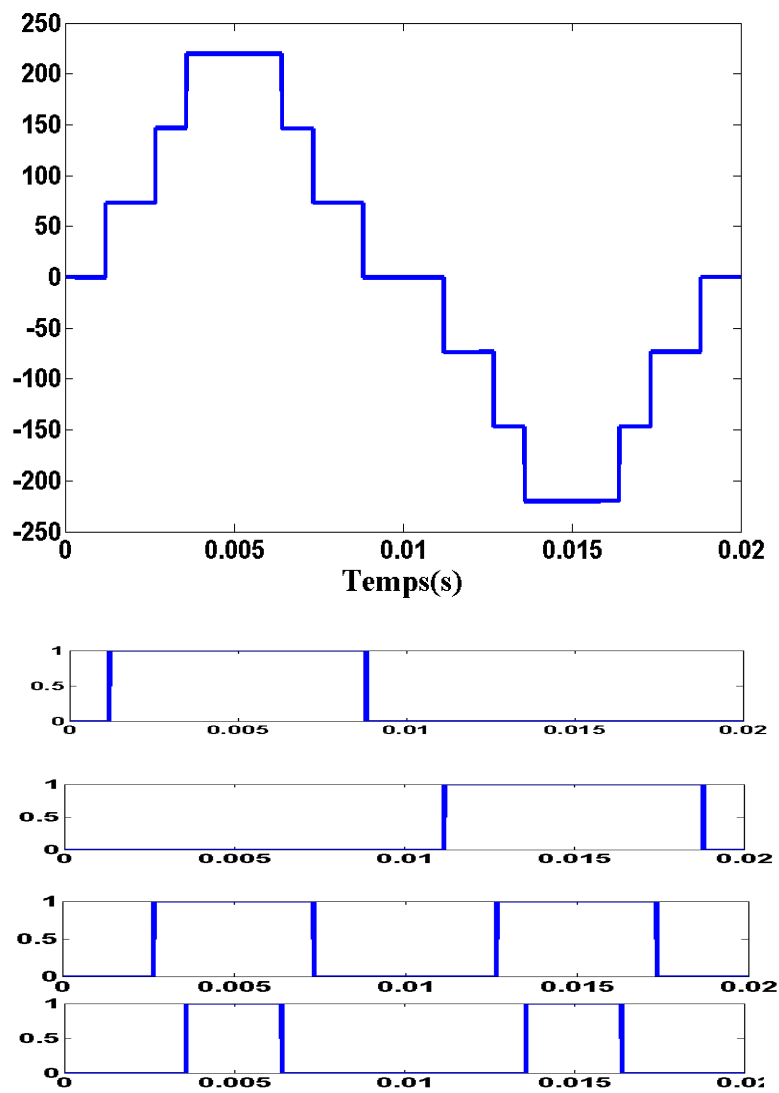


Figure (3.5) la tension V_{AM} de la première bras et les signaux de commande pour $r = 0.86$ (OHSW)

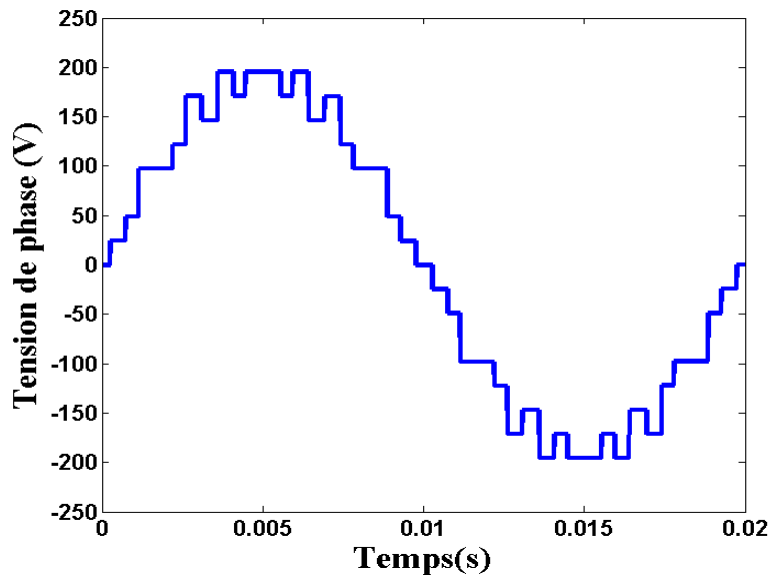


Figure (3.6) Tension simple de la première phase pour $r = 0.86$ (OHSW)

3.6 Conclusion

Dans ce chapitre, la structure de l'onduleur triphasé cascadié à sept niveaux a été présentée. Nous avons élaboré également un modèle instantané décrivant le fonctionnement de l'onduleur triphasé cascadié à sept niveaux en utilisant la notion des fonctions de connexion des interrupteurs. Le résultat intéressant qui ressort de cette analyse est la contribution apportée par les fonctions de connexion dans l'extrapolation facile du modèle de l'onduleur à niveaux inférieur à celui d'un niveau supérieur.

Nous allons par la suite intéresser par l'implémentation de l'algorithme PSO pour l'optimisation de la technique d'élimination des harmoniques, et à la commande on-line de l'onduleur triphasé cascadié à sept niveaux par les réseaux de neurones artificiels (RNA).

Chapitre 4

Résultats de simulation:

Application du PSO et RNA sur de l'onduleur triphasé cascadié à sept niveaux

4.1 Introduction

Afin que la tension délivrée par l'onduleur soit proche de la forme sinusoïdale idéale et avec un faible taux d'harmoniques, la stratégie d'élimination d'harmoniques est souvent proposée pour les onduleurs de tension. L'objectif est de calculer les instants adéquats d'ouverture et de fermeture des interrupteurs du convertisseur, pour annuler les harmoniques indésirables (de rang 5, 7, 11,...) et contrôler le fondamental. En effet, la présence de tels harmoniques dans une source de tension alimentant une charge provoque plusieurs contraintes : création des pertes au niveau des équipements, perturbation des performances de charge, diminution du facteur de puissance.

Cette technique offre donc plusieurs avantages :

- Réduction du coût des filtres.
- Réduction des pertes par commutation aux bornes des interrupteurs.
- Augmentation de la durée de vie des composants.
- Amélioration de la fiabilité du système (convertisseur - machine).

La commande, en temps réel, l'onduleur triphasé à sept niveaux par la stratégie d'élimination d'harmoniques, exige d'une part, la détermination de tous les angles d'amorçage des interrupteurs, et d'autre part, le stockage de toutes les valeurs obtenues. En effet, cette opération nécessite une très grande capacité mémoire, du fait que le calcul est effectué dans certains cas avec un pas de 10^{-6} concernant le taux de modulation r ; aussi la réalisation de cette mémoire devient coûteuse.

Dans la première partie de ce chapitre, nous allons élaborer l'algorithme PSO qui permet de calculer les angles de commutation de l'onduleur triphasé à sept niveaux . Par la suite, pour éviter l'usage des mémoires électroniques, une autre solution est préconisée consiste à utiliser la technique des réseaux de neurones artificiels (RNA) . Ces derniers sont largement appliqués dans les domaines de la classification, mémorisation, filtrage et approximation. Dans ce chapitre, nous allons surtout exploiter leur propriété d'approximation universelle. L'objectif est de reproduire toutes les caractéristiques d'allumage non linéaires, déjà calculées par PSO. Pour cela, sont mis en œuvre des réseaux multicouches entraînés par l'algorithme de rétropropagation.

4.2 Elimination d'harmoniques

4.2.1 Formulation du problème

Notre objectif est de calculer les angles de commutation afin de supprimer les harmoniques indésirables tout en contrôlant la composante fondamentale . Ce but peut être ramené à un problème d'optimisation.

On refaire à l'équations (3.11) et (3.13) dans la chapitre précédent , dans le cas où les sources continues sont identiques, il est judicieux de travailler avec des expressions relatives à la tension continue U_c . Les amplitudes relatives des harmoniques sont :

$$H_i = \frac{h_i}{U_c} \quad \text{avec } i = 1,5,7 \dots \quad (4.1)$$

Le fondamental doit être contrôlé donc à sa valeur relative de référence définie comme suit :

$$H_1^{ref} = \frac{V^{ref}}{U_c} \quad (4.2)$$

Où : V^{ref} est la tension de référence désirée.

Afin d'assurer que le fondamental va tendre vers sa valeur de référence et que les harmoniques indésirables seront nuls, le problème d'optimisation peut être formulé de la manière suivante :

1- Minimiser la fonction suivante :

$$f(\alpha_1, \alpha_2, \alpha_3) = (3r - H_1)^2 + H_5^2 + H_7^2 \quad \text{cas de la technique OHSW} \quad (4.3)$$

$$f(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (3r - H_1)^2 + H_5^2 + H_7^2 + H_{11}^2 \quad \text{cas de la technique SHEM}$$

Où : r est le taux de modulation défini dans ce cas par:

$$r = \frac{V^{ref}}{3U_c} \quad (4.4)$$

2- La solution doit satisfaire la contrainte (4.5) donnée par :

$$0 \leq \alpha_1 \leq \alpha_2 \leq \alpha_3 \leq \frac{\pi}{2} \quad (OHSW) \quad (4.5)$$

$$0 \leq \alpha_1 \leq \alpha_2 \leq \alpha_3 \leq \alpha_4 \leq \frac{\pi}{2} \quad (SHEM)$$

Les objectifs que l'on désire atteindre sont introduits dans la formulation du problème, le transformant en un problème uni-objectif. Une fois le problème d'optimisation est bien défini, nous procédons par la suite à l'utilisation les essais particulières pour trouver les angles de commutation optimaux .

4.2.2 Elimination d'harmoniques par PSO

Le but de l'algorithme PSO du optimum global est d'optimiser une fonction continue en cherchant son maximum ou son minimum dans un espace donné. L'application du PSO du optimum global dans le problème d'élimination d'harmoniques de l'onduleur à sept niveaux consiste à chercher le minimum de l'équation (4.3).

L'intégration des PSO dans le schéma fonctionnel du Commande de l'onduleur à sept niveaux par la stratégie d'élimination d'harmoniques, donné par la figure (4.1).

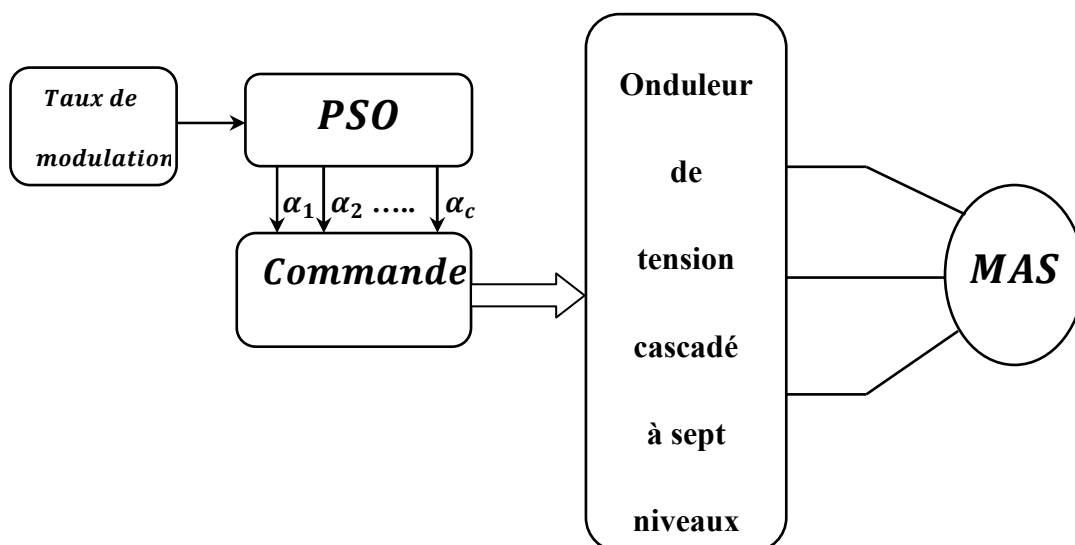


Figure (4.1) Bloc diagramme d'implantation du PSO pour calculer les angles de commutation .

Le pseudo-code adopté pour le calcul des angles de commutation se résume dans les étapes suivantes :

Données

- La taille de l'essaim;
- Le nombre maximal d'itérations $itération^{max}$;
- Le paramètre cognitif c_1 ;
- Le paramètre social c_2 ;
- Les valeurs maximale w^{max} et minimale w^{min} du coefficient d'inertie ;

Etape 1: Initialisation

Itération=0 ;

for $i=1$: nombre de particules

- Initialiser la position $\alpha^i(0) = [\alpha_1^i(0) \ \alpha_2^i(0) \ \alpha_3^i(0)]$ de chaque particule avec des angles

aléatoires entre 0 et $\pi/2$ vérifiant la contrainte (2.7);

- Initialiser la vitesse $v_{\alpha}^i(0) = [v_{\alpha_1}^i(0) \ v_{\alpha_2}^i(0) \ v_{\alpha_3}^i(0)]$. de chaque particule à une valeur aléatoire;

- Initialiser la valeur de Fitness $_p_i$ de la particule i ;

end for

- Initialisation de Fitness $_g$ de l'essaim ;

While nombre maximal d'itérations non atteint

$itération = itération + 1$;

- Calculer le coefficient d'inertie w

$$w = w^{max} - \frac{w^{max} - w^{min}}{itération^{max}} itération ;$$

Contraintes : Les angles de commutation retenus doivent respecter impérativement la contrainte(4.5).

for $i=1$: nombre de particules

Etape 2: Evaluation de la fonction objective

- Calculer la valeur de la fonction $Fitness_i$ de chaque particule i de l'essaim en utilisant l'équation (4.3);

Etape 3: Mise à jour de la meilleure position du particule i

if $Fitness_i < Fitness_{p_i}$

then $Fitness_{p_i} = Fitness_i$ et $\alpha_{p_i} = \alpha^i$

end if

Etape 4: Mise à jour de la meilleure position de l'essaim

If $Fitness_{p_i} < Fitness_g$

Then $Fitness_g = Fitness_{p_i}$ et $\alpha_g = \alpha_{p_i}$

end if

end for

for $i = 1$: nombre de particules

Etape 5: Mise à jour de la vitesse et de la position du particule i

$$v_{\alpha}^i = wv_{\alpha}^i + c_1rand_1(\alpha_{p_i} - \alpha^i) + c_2rand_2(\alpha_g - \alpha^i)$$

$$\alpha^i = \alpha^i + v_{\alpha}^i$$

end for

end while

4.2.3 Résultats de simulation

Pour montrer la capacité des essaims de particules dans l'optimisation de la commande de l'onduleur triphasé cascadié à sept niveaux, les paramètres de l'algorithme PSO utilisé dans l'optimisation de la commande OHSW sont :

- $S = 20$; taille de l'essaim .
- $k = 1000$; nombre d'itérations.
- $c_1 = c_2 = 1.8$
- w ; d'après *Eberhart et al*
- Condition d'arrêt est ($k=1000$)

Toutefois, pour l'étude du comportement de la meilleure particule dans l'essaim, donné par la figure (4.2), nous avons fixé le nombre maximal d'itérations à 500.

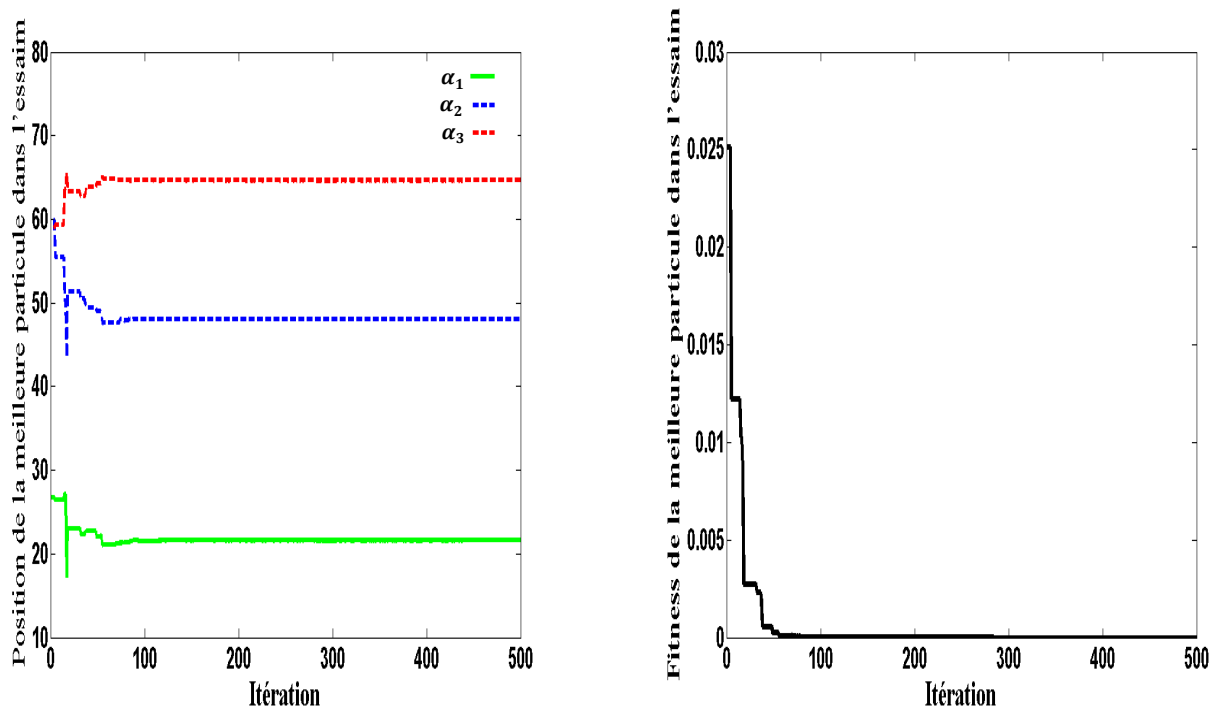


Figure (4.2) Comportement de la meilleure particule dans l'essaim pour $r=0.86$

Les angles de commutation sont déterminés de façon à annuler les harmoniques de rangs cinq et sept. La figure (4.3) représente la variation de tous les angles de commutation possibles en fonction du taux de modulation. L'analyse de cette figure montre que la fonction à optimiser ne possède des solutions que pour un taux de modulation compris entre 0.500 et 1.069. En outre, des solutions multiples existent pour r , dans ce cas, les angles de commutation adéquats sont sélectionnés sur la base de la meilleure valeur du taux de distorsion harmonique. Ceux qui donnent la plus faible valeur du THD. La figure (4.4) représente une évaluation de la fonction d'adaptation en fonction du taux de modulation et donne une idée sur la précision par laquelle nous avons retenu les solutions.

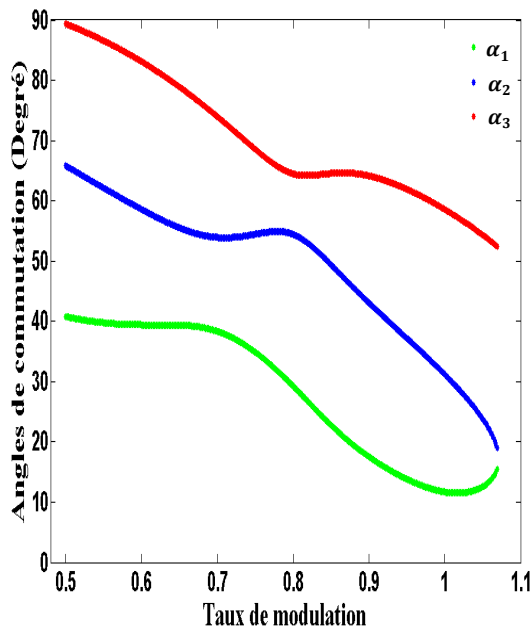


Figure (4.3) Angles de commutation minimisant le THD (OHSW)

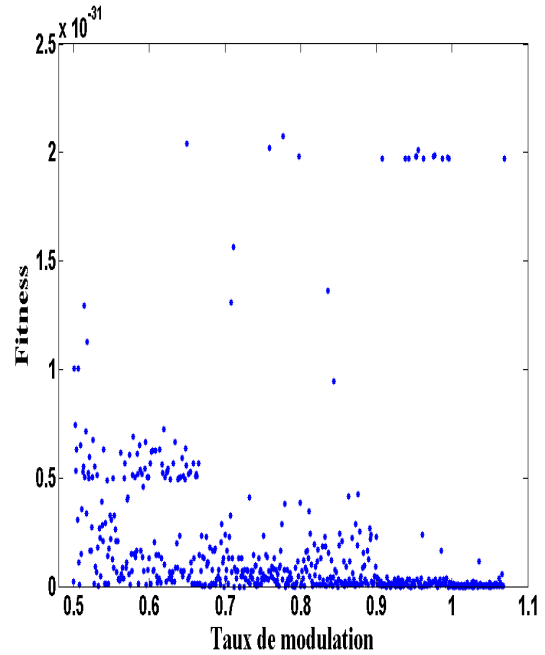


Figure (4.4) Fitness des solutions minimisant le THD (OHSW)

L'efficacité de la méthode PSO particulières se manifeste dans les faibles valeurs des amplitudes des harmoniques éliminés comme le montre la figure (4.5) . La figure (4.6) donne l'allure du THD le plus faible. Le taux de distorsion harmonique diminue au fur et à mesure que le taux de modulation augmente. C'est la caractéristique qui doit être retenue pour assurer un fonctionnement optimal de l'onduleur.

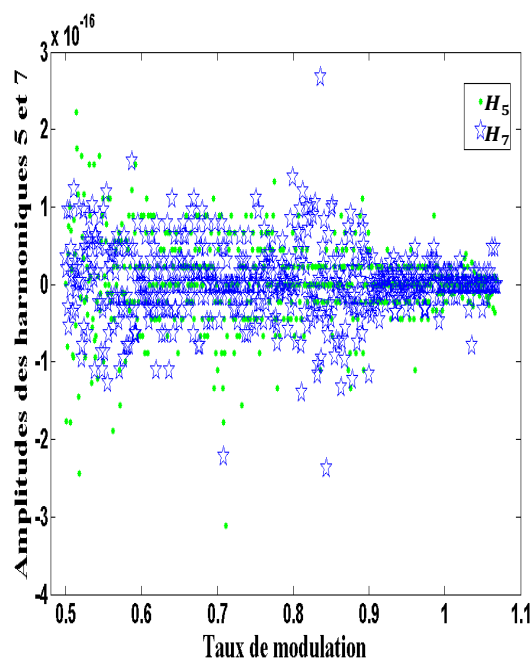


Figure (4.5) Amplitudes des harmoniques éliminés (OHSW)

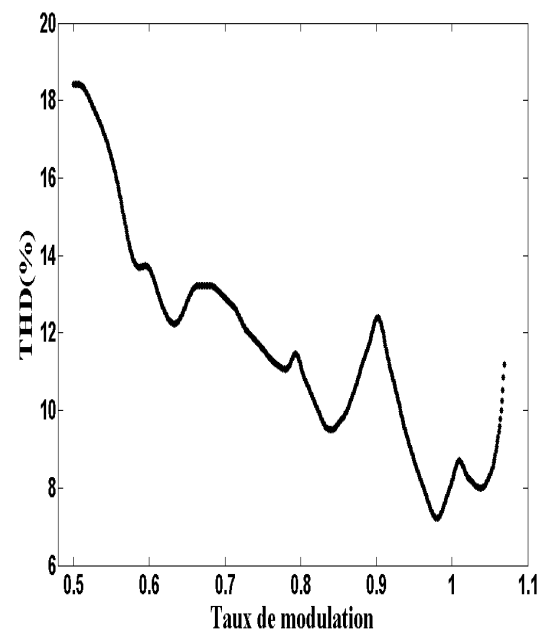


Figure (4.6) THD le plus faible (OHSW)

La figure (4.7) donne l'allure de la tension simple et son spectre harmonique dans le cas d'une commande OHSW. L'analyse harmonique de la tension de phase montre que les harmoniques d'ordre cinq et sept sont effectivement disparus.

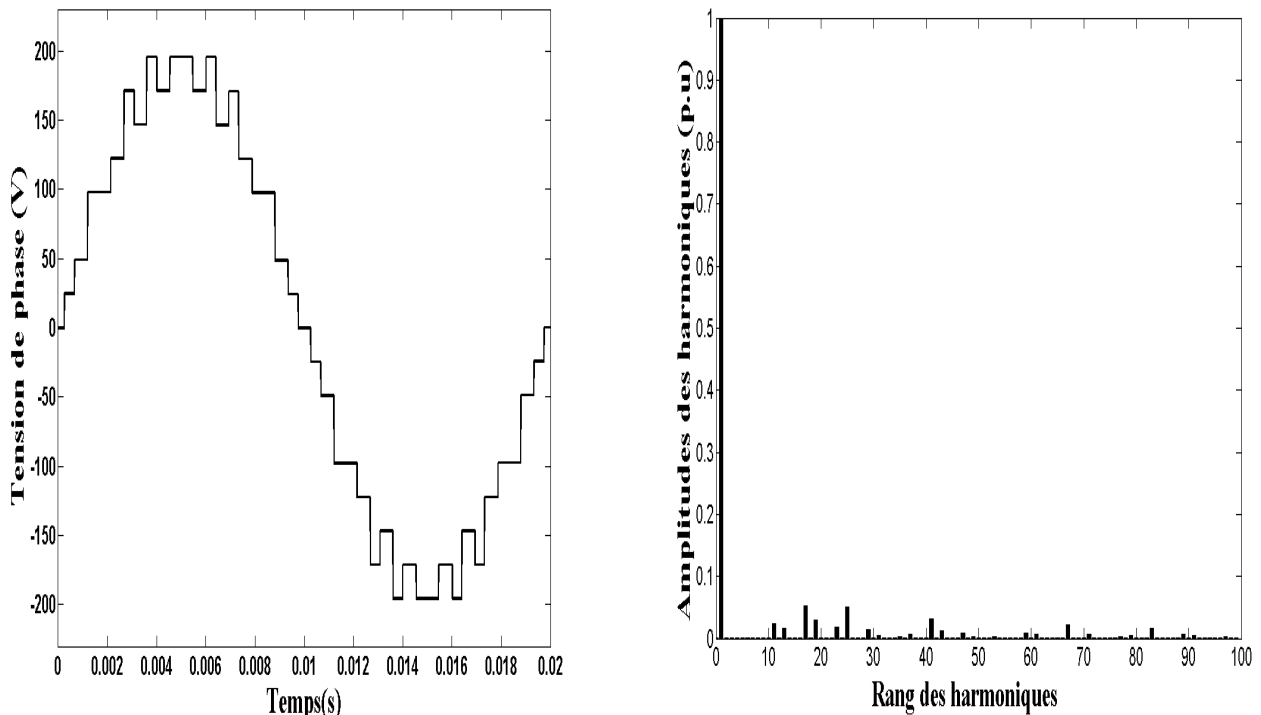


Figure (4.7) Tension simple et son spectre d'harmoniques pour $r=0.86$

En ce qui concerne l'optimisation de la technique SHEM, nous avons opté pour les mêmes paramètres c_1 , c_2 et w par contre la taille de l'essai a été augmentée, en adoptant les paramètres suivants:

- $S = 30$; taille de l'essai .
- $k = 1000$; nombre d'itérations.
- $c_1 = c_2 = 1.8$
- w ; d'après *Eberhart et al*
- Condition d'arrêt est ($k=1000$)

On remarque qu'il existe des solutions multiples pour les valeurs du taux de modulation . Toutefois, pour les valeurs du taux de modulation inférieures à 0.575 ou supérieures à 1.020, aucune solution n'est possible. Quand des solutions multiples existent, une possibilité serait de choisir les angles de commutation qui donnent le THD le plus faible .Dans la figure (4.8) la variation des angles de commutation en fonction du taux de modulation présente des discontinuités ; ce qui reflète la non linéarité de cette caractéristique.La valeur de la fonction d'adaptation représentée par la figure 4.9 montre que les solutions sont calculées avec une précision assez faible.

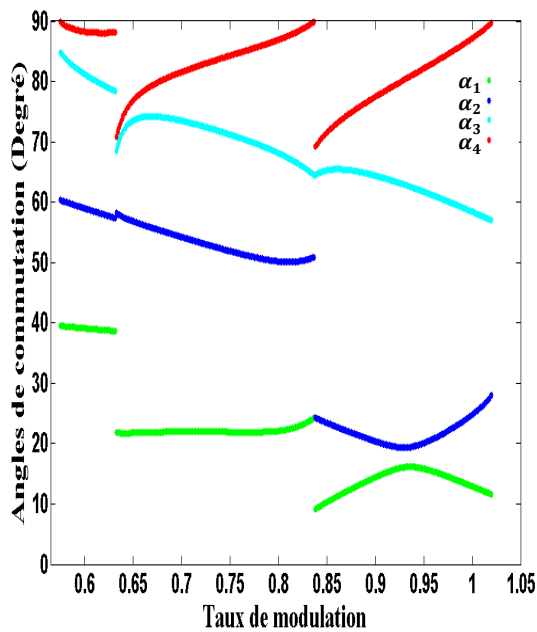


Figure (4.8) Angles de commutation minimisant le THD (SHEM)

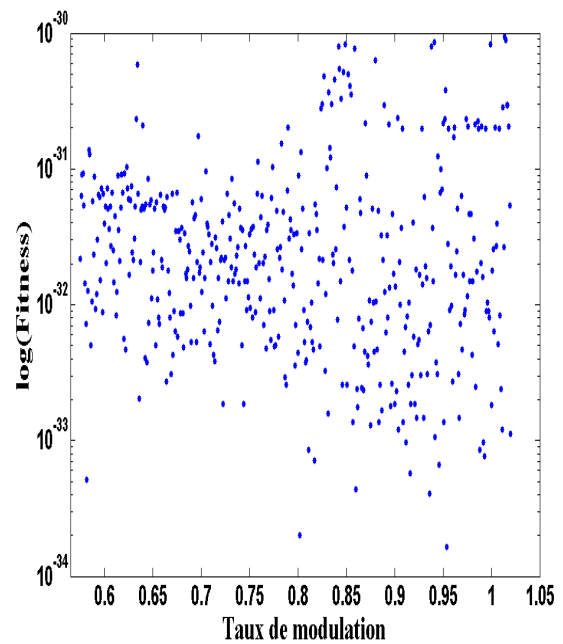


Figure (4.9) Fitness des solutions minimisant le THD (SHEM)

La figure (4.10) donne l'allure du THD le plus faible est . Comparé au THD de la technique OHSW, le THD de la technique SHEM est amélioré d'une façon sensible. La figure (4.11) , il apparaît bien que les harmoniques cinq, sept et onze sont nuls.

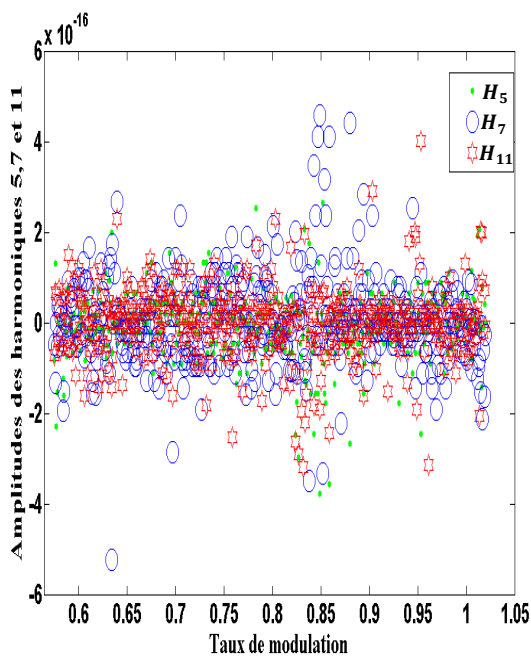
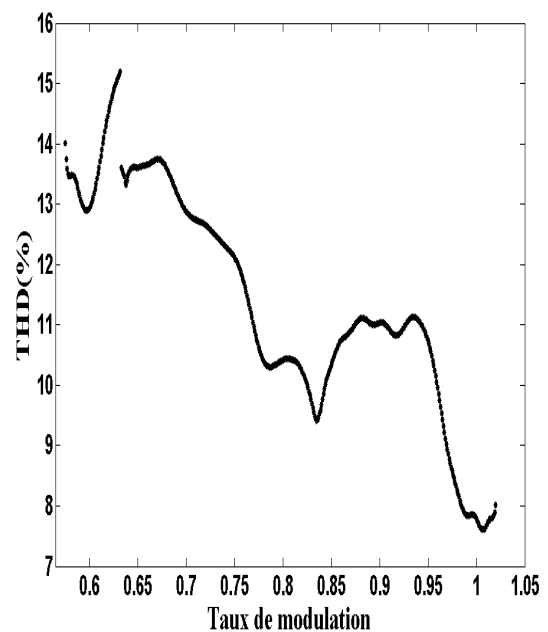


Figure (4.10) Amplitudes des harmoniques éliminés (SHEM)



Figure(4.11) THD le plus faible (SHEM)

La figure (4.12) donne l'allure de la tension simple et son spectre harmonique dans le cas d'une commande SHEM . L'analyse harmonique de la tension de phase montre que les harmoniques d'ordre cinq , sept et onze sont effectivement disparus.

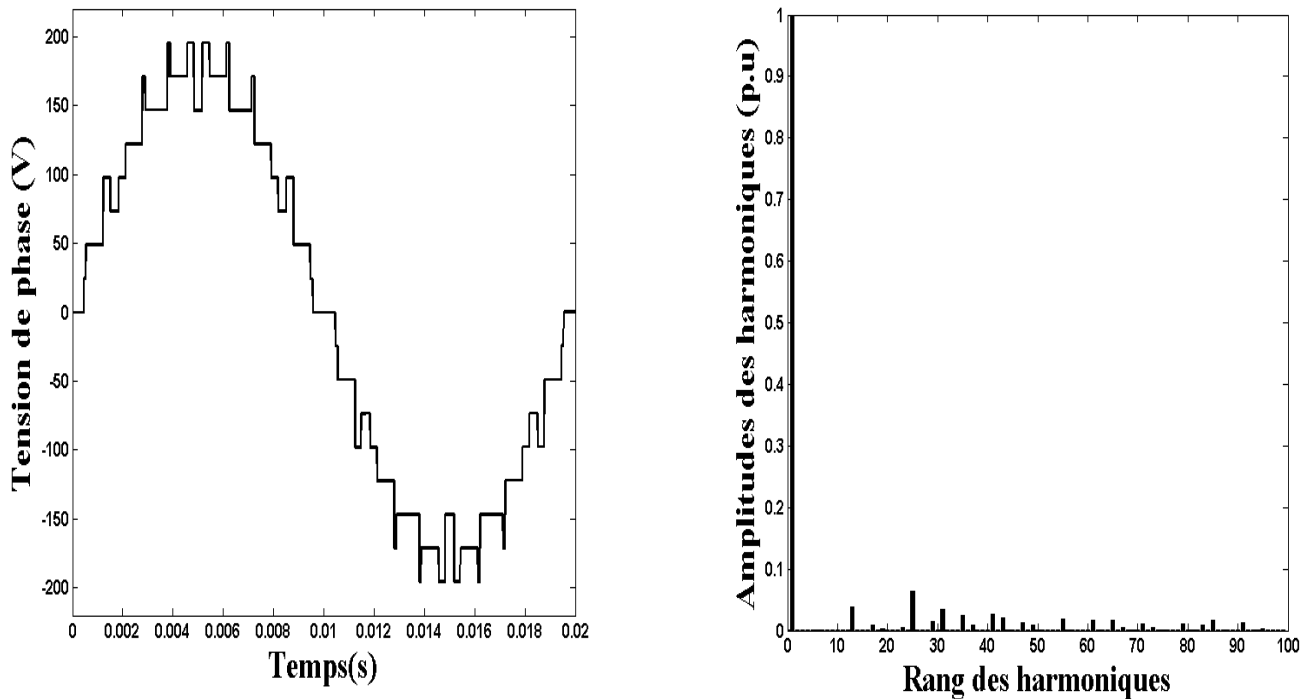


Figure (4.12) Tension simple et son spectre d'harmoniques pour $r=0.80$

4.3 Commande de l'onduleur triphasé à sept niveaux par les réseaux de neurones

La commande par élimination d'harmoniques de l'onduleur nécessite la connaissance des angles de commutation lesquels varient en fonction du nombre d'harmoniques H à éliminer ($c - 1$), du taux de modulation r . Au section précédent, nous avons déterminé l'évolution de ces angles de commutation en fonction de r . Les valeurs de ces angles sont stockées dans des tables qui dépendent de r et H .

De plus, pour capter réellement les non linéarités dures affectant les courbes $\alpha_i = f(r)$ $i = 1,2,3$ de l'onduleur triphasé à sept niveaux commandé par OHSW l'incrémentation Δr de la variable r doit être petite ; dans notre exemple $\Delta r = 10^{-3}$. Ceci conduit impérativement à une taille importante de la table d'où une implémentation difficile et coûteuse.

Pour remédier à ce problème, il est possible d'exploiter la propriété d'approximation universelle des réseaux de neurones formels pour copier, les courbes $\alpha_i = f(r)$ par un seul réseau de neurones.

Aussi, un réseau de taille raisonnable permet de fournir en temps réel les valeurs des angles α_i du fait que le processus est parallèle au niveau du réseau. Mais, le plus important avantage concerne leur propriété de généralisation qui permet au RNA de fournir les valeurs des angles même pour les valeurs non incluses dans la table.

De l'étude bibliographique, il ressort que les réseaux multicouches constituent les modèles les plus efficaces pour la généralisation des fonctions non linéaires. Néanmoins, le problème de la détermination du nombre de neurones cachés se pose toujours. Pour assurer leur apprentissage, la méthode de rétropropagation avec l'approche de Levenberg -Marquardt représente l'algorithme qui permette d'obtenir une bonne approximation.

La figure (4.13) montre que les angles de commutation sont fournis par RNA en temps réel.

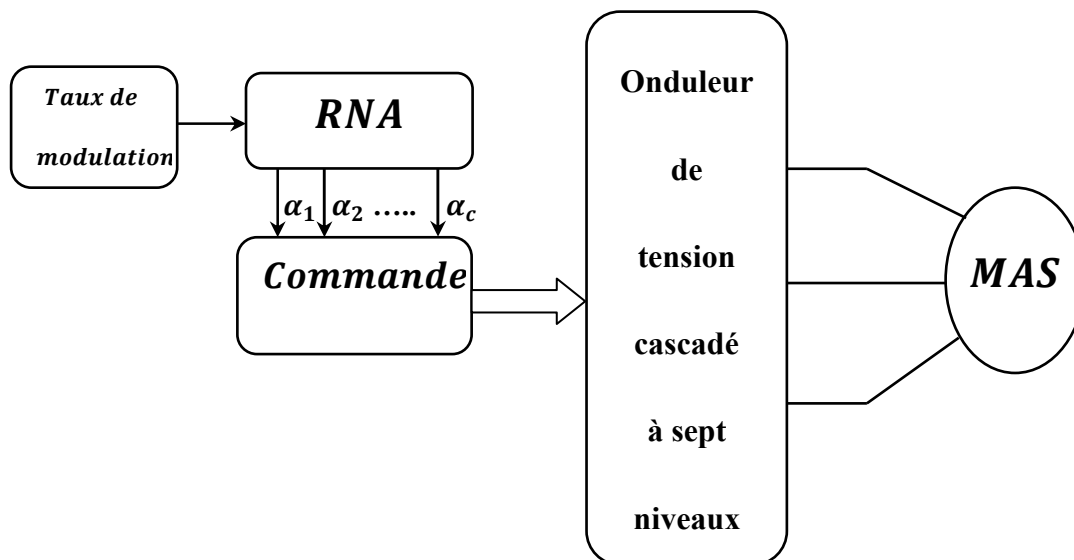


Figure (4.13) Bloc diagramme d'implantation des RNA pour Commander l'onduleur triphasé à sept niveaux.

4.3.1 Architecture du réseau élaboré

Comme le réseau doit recevoir à son entrée la valeur $r(k)$ et doit fournir à sa sortie les angles α_i avec. Donc le réseau doit comporter un seul neurone à l'entrée et 3 neurones à sa sortie correspondant aux 3 angles de commutation qu'il doit générer. Pour cette étude le réseau est à une seule couche cachée comporte 30 neurones.

4.3.2 Résultats de simulation

Pour l'élimination d'harmoniques d'ordre cinq et sept, les exemples d'apprentissage sont tels que le vecteur d'entrée *Taux de modulation* est donné par :

$$\text{Taux de modulation} = [r(1), \dots, r(k), \dots, r(n)]$$

et la matrice de sortie désirée donnée par :

$$\text{Angles de commutation} = \begin{bmatrix} \alpha_1(1) \cdots \alpha_1(k) \cdots \alpha_1(n) \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ \alpha_3(1) \cdots \alpha_3(k) \cdots \alpha_3(n) \end{bmatrix}$$

Le nombre n représente le nombre de valeurs mesurées ou le nombre d'exemples. En outre, à l'entrée $r(k)$, le réseau doit fournir les angles de sortie désirée $\alpha^T(k) = [\alpha_1(k), \dots, \alpha_3(k)]$. Lors de l'apprentissage, la valeur $r(k)$ est présentée à l'entrée du réseau puis propagée jusqu'à la sortie pour obtenir les angles de sortie $\hat{\alpha}^T(k) = [\hat{\alpha}_1(k), \dots, \hat{\alpha}_3(k)]$. La différence entre α et $\hat{\alpha}$ constitue l'erreur d'apprentissage sur l'exemple k . Ce processus est répété pour tous les exemples (k varie de 1 à n). Par la suite, l'erreur globale sur les angles est exploitée par l'algorithme de backpropagation pour adapter les valeurs des paramètres du réseau (poids et biais). De même, des itérations sur tous les exemples sont effectuées jusqu'à l'obtention de la convergence du réseau.

A la fin de cette phase d'apprentissage, tous les paramètres sont adaptés, de telle manière que lorsque le réseau reçoit une valeur actuelle $r(k)$, il fournira automatiquement les angles correspondant à l'entrée introduite et cela grâce aux poids synaptiques calculés lors de la phase d'apprentissage. Le réseau est donc prêt, après la phase d'adaptation, d'être implémenté pour une commande en temps réel.

La figure (4.14) représente la variation de tous les angles de commutation obtenus par les réseaux élaborés en fonction du taux de modulation. L'analyse de cette figure montre que les deux caractéristiques (sans RNA et avec RNA) coïncident pratiquement. La figure (4.15) représente une évaluation de l'erreur quadratique entre $\hat{\alpha}_i$ et α_i en fonction du taux de modulation, et illustre que un bon apprentissage est réalisé par le réseau obtenu, et donne une idée sur la précision par la quelle nous avons retenu les solutions.

Où

$$e_1 = (\alpha_1 - \hat{\alpha}_1)^2$$

$$e_2 = (\alpha_2 - \hat{\alpha}_2)^2$$

$$e_3 = (\alpha_3 - \hat{\alpha}_3)^2$$

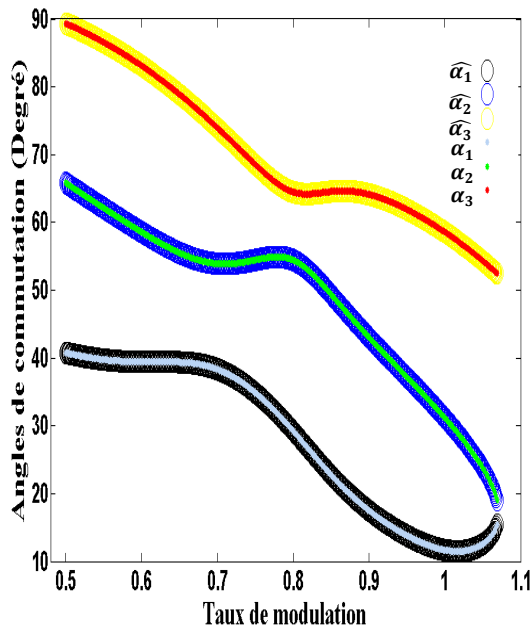


Figure (4.14) Les angles de commutation donnés par les RNA et par la méthode de PSO pour (OHSW)

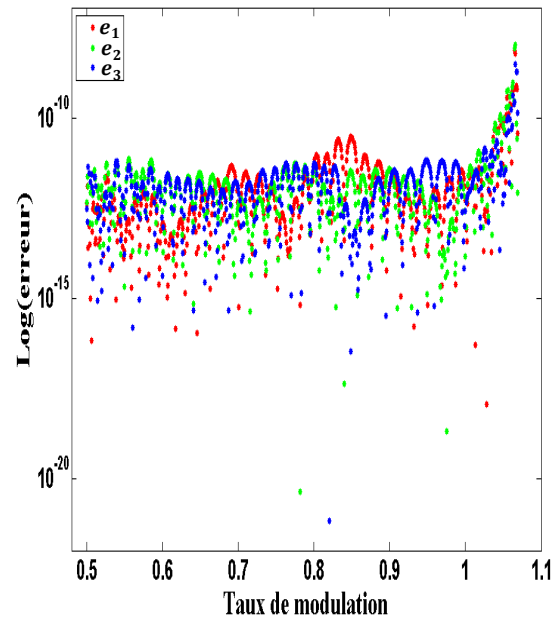


Figure (4.15) L'erreur entre les angles donnés par RNA et par la méthode de PSO pour (OHSW)

La valeur de la fonction d'adaptation représentée par la figure (4.16) donne une idée sur la précision par la quelle nous avons retenu les solutions. L'efficacité de la RNA se manifeste dans les faibles valeurs des amplitudes des harmoniques éliminés comme le montre la figure (4.17).

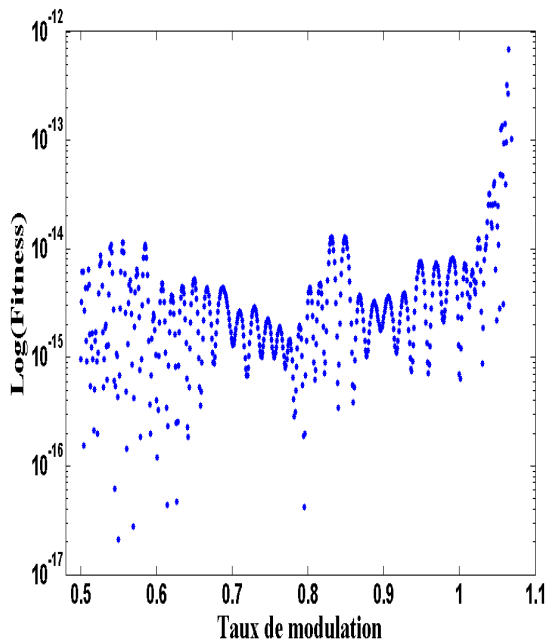


Figure (4.16) Fitness des solutions de RNA (OHSW)

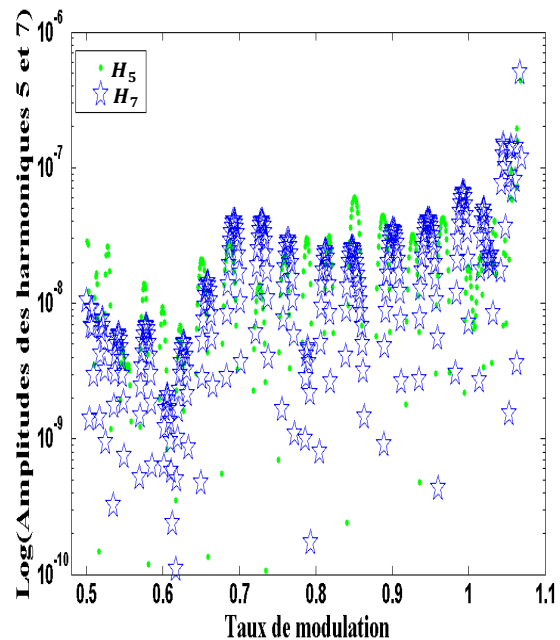


Figure (4.17) Amplitudes des harmoniques éliminés par RNA (OHSW)

La figure (4.18) donne l'allure du THD. C'est la caractéristique qui doit être retenue pour assurer un fonctionnement optimal de l'onduleur.

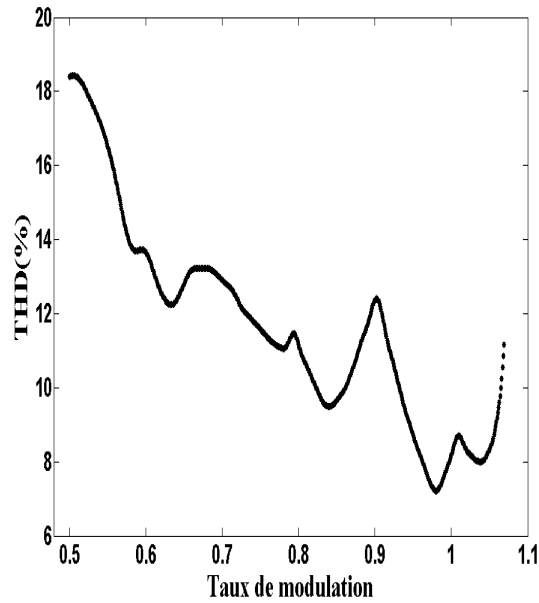


Figure (4.18) THD pour (OHSW)

De plus, les réseaux sont capables de fournir les angles de commutation pour la stratégie d'élimination d'harmoniques même pour les exemples non présentés lors de l'apprentissage. Par conséquent, un bon apprentissage est réalisé par le réseau obtenu.

La figure (4.19) représente la variation de tous les angles de commutation obtenus par les réseaux élaborés en fonction du taux de ,mais dans ce cas avec un $pas = 10^{-6}$. La figure (4.20) représente une évaluation de la fonction d'adaptation en fonction du taux de modulation. Cette évaluation montre que la valeur du *Fitness* est très faible .

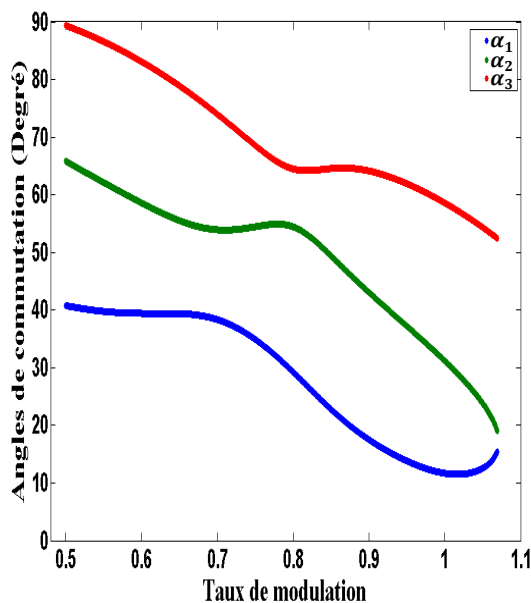


Figure (4.19) Angles de commutation donnés par RNA (OHSW)

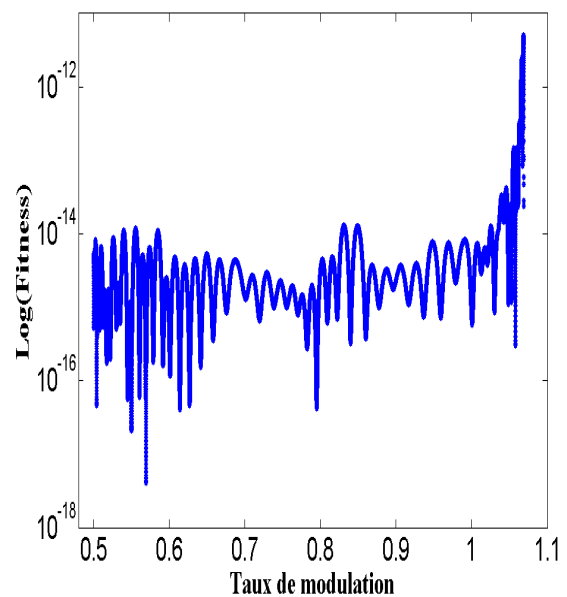


Figure (4.20) Fitness des solutions de RNA (OHSW)

L'efficacité de la RNA se manifeste dans les faibles valeurs des amplitudes des harmoniques éliminés comme le montre la figure (4.21). La figure (4.22) donne l'allure du THD.

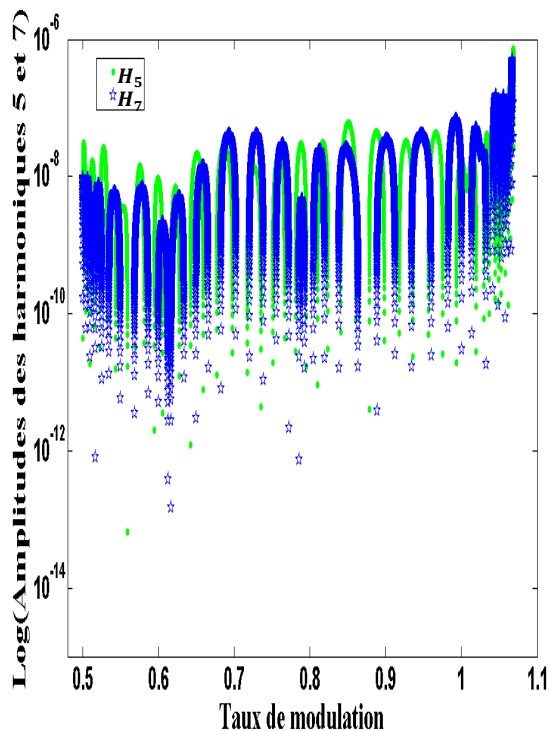


Figure (4.21) Amplitudes des harmoniques éliminés par RNA (OHSW)

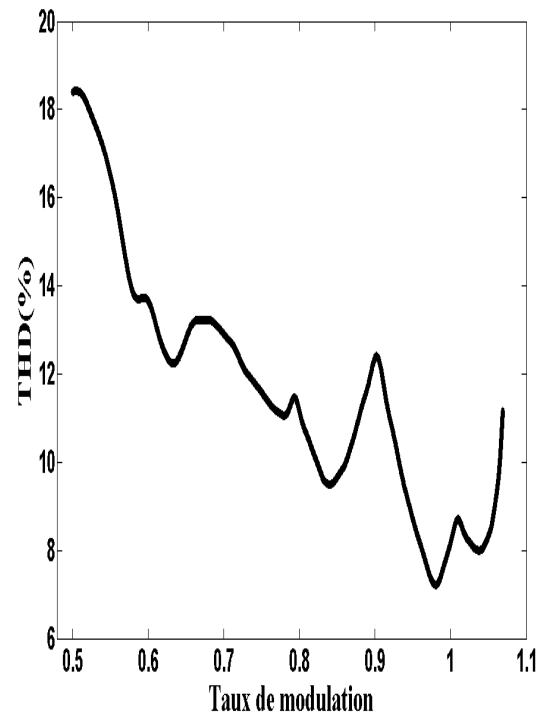


Figure (4.22) THD pour (OHSW)

4.4 Conclusion

La méthodologie adoptée dans ce chapitre consiste à transformer le problème d'élimination d'harmoniques en un problème d'optimisation. Un algorithme d'optimisation est mis en oeuvre pour sa résolution. Il s'agit de l'algorithme des essais particuliers. Les résultats obtenus ont montré que la méthode a réussi d'atteindre l'objectif d'une manière très satisfaisante en termes de précision et de temps de calcul.

La stratégie d'élimination d'harmoniques offre un taux de distorsion harmonique faible. En outre, cette stratégie permet, non seulement le contrôle du fondamental mais également d'annuler les harmoniques les plus gênants. D'autant plus qu'elle réduit énormément les pertes par commutation ce qui augmente la longévité des composants de puissance.

Du fait la stratégie d'élimination d'harmoniques ne dépend pas de la topologie de l'onduleur utilisé, cette étude est applicable donc à d'autres topologies et peut être étendue facilement à d'autres niveaux de tension.

De cette étude, il ressort que le réseau multicouches statiques sont les plus appropriés pour le problème d'approximation des fonctions non linéaires.

Les caractéristiques de commutation obtenues par les RNA sont pratiquement similaires à celles déterminées par la méthode des essais particulières. Ce qui valide, sans doute, l'excellente performance des réseaux élaborés.

Conclusion générale

Le but principal des travaux dans ce mémoire s'articule fondamentalement sur deux idées, à savoir l'optimisation de la commande de l'onduleur triphasé cascadié à sept niveaux par un algorithme intelligent et le développement de la commande de l'onduleur par les réseaux de neurones artificiels.

Pour atteindre ces objectifs, la première étape du présent travail était consacrée à la présentation de l'algorithme intelligent d'optimisation par essaim particulaire (PSO), l'apprentissage des réseaux de neurones artificiels, et l'établissement d'un modèle de connaissance de l'onduleur ce qui a permis de dévoiler son comportement vis-à-vis des commandes développées et de le situer par rapport à une structure conventionnelle.

A travers la deuxième partie, nous avons appliqué PSO et RNA sur l'onduleur. L'application du PSO dans l'optimisation de la technique de commande par élimination d'harmoniques permet de surmonter les problèmes posés par les méthodes déterministes. Pour assurer alors la commande du convertisseur en temps réel, nous avons proposé d'exploiter la propriété d'approximation universelle des réseaux de neurones artificiels. Aussi, nous avons élaboré des réseaux de neurones multicouches entraînés par l'algorithme de rétropropagation, pour approximer toutes les caractéristiques non linéaires, déjà déterminées par la méthode PSO. Par conséquent, l'entrée du réseau reçoit les valeurs de taux de modulation r et la sortie délivre automatiquement les angles de commutation correspondant à la valeur de r introduite à l'entrée.

Dans le contexte de l'optimisation de la technique de commande par élimination d'harmoniques, des travaux déjà réalisés ont exploité principalement la méthode de Newton-Raphson [2],[7],[8],[14]. Pour assurer la convergence de cet algorithme, pour n'importe quelle valeur du taux de modulation, de multiples points de départ sont nécessaires. Chaque point de départ est défini dans un intervalle spécifique du taux de modulation. La définition de ces intervalles et les points de départ associés est une opération relativement fastidieuse. Pour cette

raison, nous avons exploré la voie qu'offrent les essaims particulaires appliqués à l'optimisation de la commande de l'onduleur. Cette méthode ont l'avantage d'être beaucoup moins sensibles au point de départ et aux autres contraintes liées à la dérivation de la fonction à optimiser qui sont propres aux algorithmes d'optimisation classiques.

En effet, de façon générale les méthaheuristiques démarrent avec un ensemble de points de départ éparpillés sur tout l'espace de recherche et identifient rapidement les sous domaines susceptibles de contenir le minimum global.

L'objectif assigné pour le développement de la commande de l'onduleur est l'exploitation des réseaux de neurones pour l'approximation des caractéristiques de commutation de l'onduleurs. Nous avons déterminé les réseaux capables de reproduire les angles de commutation assurant l'élimination des harmoniques non désirés. Dans ce contexte, il ressort que le réseau multicouches statiques sont les plus appropriés pour le problème d'approximation des fonctions non linéaires. De plus leur apprentissage supervisé est rendu possible grâce au formidable développement de la technique rétropropagation (backpropagation).

Pour les perspectives de ce mémoire, nous proposons les idées suivantes :

- Développement de l'algorithme du PSO du optimum local pour la résolution des systèmes d'équations non linéaires.
- L'élimination d'un nombre élevé d'harmoniques nécessite des algorithmes d'optimisation plus puissants. l'algorithme du PSO du optimum local apparaissent la solution la plus adaptée pour ce type de problème;
- Développement de l'apprentissage des réseaux de neurones artificiels par l'algorithme du PSO du optimum local.

A. APPRENTISSAGE DU RESEAU RBF

L'apprentissage des réseaux RBF est composé d'une règle d'apprentissage non supervisée pour l'adaptation des centres et d'une règle d'apprentissage supervisé pour l'adaptation des poids.

A.1. Apprentissage des centres [22]

L'apprentissage des centres se fait selon les étapes suivantes :

1- Présenter un signal x_n au réseau

2- Calculer la distance entre le vecteur d'entrée et les centres $D_i(n) = \|x(n) - c_i(n)\|$.

3- Déterminer le centre le plus proche du signal d'entrée I de façon que :

4-

$$D_I(n) = \min D_I(n)$$

5- Mettre à jour les centres c_I selon l'équation suivante :

$$c_I(n+1) = c_I(n) + \mu(x(n) - c_I(n))$$

6- Aller à l'étape 1.

A.2. Adaptation des poids [22]

L'adaptation des poids est réalisée par la règle de rétropropagation du gradient, elle est utilisée sur la seule couche de sortie :

1- Présenter une paire des entrées et de sortie désirée (x_n, d_n) .

2- Calculer l'erreur entre la sortie du réseau et la sortie désirée : $e_j(n) = d_j(n) - y_i$

3- Mise à jour des poids de la couche de sortie selon l'équation :

$$w_{kj}(n+1) = w_{kj}(n) + \alpha e_j(n) \phi_k(\|x - c_k\|)$$

α est une constante petite positive

4- Aller à l'étape 1.

Bibliographie

- [1] Aleksandar Lazinica, Particle Swarm Optimization , Published by In-Tech , Croatia , 2009.
- [2] ALEX Q & JASON Lai , Chair & DUSAN Borojevic . Huang , Optimized harmonic stepped-waveform for multilevel inverter , Mémoire de Master , the Faculty of the Virginia Polytechnic Institute and State University In partial fulfillment of the requirements for the degree of ,1999.
- [3] ANDRIES P. Engelbrecht ,Computational Intelligence ,British Library ,second edition , 2007.
- [4] BARKAT Said , Modélisation et commande d'un onduleur à sept niveaux à diodes flottantes : Application à la conduite d'une machine asynchrone, Thèse de doctorat , de l'Ecole Nationale Polytechnique d'Alger, 2008.
- [5] BAZI Smail , Contribution à la commande robuste d'une machine asynchrone par la technique PSO « particle swarm optimization », Mémoire de Magister de Université de Batna, 2009.
- [6] BELHACHAT Faiza, Commande neuro-floue d'un hacheur MPPT, Mémoire de Magister de l'Ecole Nationale Polytechnique d'Alger, 2007.
- [7] BOUZIDI Chaâbane & CHOUIDIRA Adel & GASMI Abdelkrim , Application des réseaux de neurones à la commande des onduleurs multiniveaux cascades symétriques , Mémoire d'ingénieur de l'Université Mohamed Boudiaf de M'sila ,2007.
- [8] BRIK Youcef & OUAGUENI Fayssal , Réalisation d'un onduleur monophasé cascadié à sept Niveaux , Mémoire d'ingénieur de l'Université Mohamed Boudiaf de M'sila ,2007.
- [9] CHEKIRED Fathya Etude et implémentation d'un commende MPPT neuro-floue sur FPGA, Mémoire de Magister de l'Ecole Nationale Polytechnique d'Alger, 2008.
- [10] CHRISTIAN Blum & DANIEL Merkle ,Swarm Intelligence Introduction and Applications , Verlag Berlin Heidelberg 2008.
- [11] FATIMA ZOHRA Aiouaz SAÏFI Aouichat , Identification et controle par reseaux de neurones, Mémoire d'ingénieur de l'Université Mohamed Boudiaf de M'sila ,2006.
- [12] FRANCIS Labrique & GUY Segulier & ROBERT Bausiere , les convertisseur de l'électronique de puissance "volume 4 les convertisseur continu-alternatif"
- [13] FREDERIC Bonnans ,Optimisation continue ,
- [14] Imarazene Khoukha, Applications des réseaux de neurones à la commande par élimination d'harmoniques des onduleurs multiniveaux, Mémoire de Magister de l'Ecole Nationale Polytechnique d'Alger, 2005.
- [15] ISSAADI Salim,Commande d'une poursuite du point de puissance maximum (MPPT) par les Réseaux de Neurones , Mémoire de Magister de l'Ecole Nationale Polytechnique d'Alger, 2006.
- [16] James Kennedy & Russell C. Eberhart, Swarm Intelligence, Academic Press, 2001.

- [17] Keith Jeremy McKenzie , Eliminating Harmonics in a Cascaded H-Bridges Multilevel Inverter Using Resultant Theory, Symmetric Polynomials, and Power Sums , Mémoire de Master , The University of Tennessee, Knoxville ,2004.
- [18] LADJEL Mohammed , Traitement et fusion multisensorielle Appliqués à la surveillance des eaux Potables, Mémoire de Magister de Université Mohamed Boudiaf de M'sila ,2006.
- [19] Maurice Clerc, L'essaim de particules vu comme un système dynamique : convergence et choix des paramètres, http://www.particle swarm.net/oep_2003.
- [20] : Maurice Clerc, L'optimisation par essaim particulaire, Tutorial pour PSO 2003-
http://www.particle swarm.net/oep_2003.
- [21] Maurice Clerc, Particle Swarm Optimization, British Library, 2006.
- [22] ZERDOUMI Zahra , Application des réseaux de neurones Artificiels à la poursuite des non Linearités fluctuantes des systèmes Satellitaires , Mémoire de Magister de l'Université Mohamed Boudiaf de M'sila ,2006.

Résumé:

Ce travail s'intéresse à l'optimisation des systèmes non-linéaires, Il s'agit en premier lieu à la présentation de l'optimisation par essaim particulaire, et les réseaux de neurones artificiels, d'établir un modèle mathématique de l'onduleur triphasé cascadié à sept niveaux qui nous permet de mettre en valeur cette structure par rapport une structure conventionnelle. Nous nous sommes penchés par la suite sur l'optimisation de la commande par élimination d'harmoniques moyennant des essais particulaires qui s'avère plus adapté à ce sujet par rapport aux méthodes classiques d'optimisation, et à la développement de la commande par élimination d'harmoniques de l'onduleur en exploitant les réseaux de neurones artificiels (RNA).

Mots clés : l'optimisation par essaim particulaire (PSO), Réseau de neurones, onduleur, stratégie d'élimination d'harmoniques.