

Remerciements

On remercie **DIEU** le tout puissant pour nous avoir donné le courage et la volonté d'achever ce travail.

Mes sincères reconnaissances pour notre promoteur **Mr MEGUELATI** pour toute sa confiance et l'aide précieuse qu'il nous a apporté tout **au long** de la réalisation de ce travail.

On tient à **remercies**, et exprimer notre respect aux membres du jury pour avoir accepté d'évaluer ce modeste travail, qu'il trouve ici l'expression de notre infini gratitude et l'assurance de notre haute considération.

On remercie **Mr Benouared** qui nous a beaucoup **aidé**.

On remercie toute personne ayant participé de près ou de loin à l'élaboration de ce travail.



DEDICACE

*Je remercie en premier lieu, DIEU qui ma **donner** la confiance et la sagesse de bien terminer ce travail.*

Nul ne peut égaler la tendresse et la gentillesse des parents.

Nul ne peut savourer le succès de quelqu'un autant que les parents, pour cette raison

Je dédie ce mémoire de fin d'étude à mes chers parents que DIEU me les gardes (AHMED et OM ESSAAD).

Je dédie ce travail à tout :

Mes sœurs : Hakima, Oufia, Souad et Amel

Mes frères : Kamel, Boubker, Nacir, Abd elmetalib, Abd elhalim et Adel

Et a tout ma belle famille AYAD et en oubliant pas la famille MALEK : Tata Yamina et ses filles surtout Ratiba, Faiza, Ferial et la petite Anais.

A mes chère sœurs dans la mousque ABOU AZAHRAE en particulier : Houria, Farida, Khawla, Zahia, Iman, Amina, Fatiha....

A tout mes amies de la cite 05 et mes collègues de département AERONAUTIQUE en particulier option Installation de cette promos.

A ma binôme Samia et toute sa famille.

AYAD YASMINA



LISTE DES ABREVIATIONS

HPBW: Ouvreur de mi-puissance (Half Power Beam Wide).

AG: L'algorithme génétique.

RS : recuit simulé.

MLP : Le perceptron multicouche (Multilayer perceptron).

GBP : Algorithme de la rétro- propagation (Gradient Backpropagation).

CG : Algorithme du gradient conjugué.

MCG : Algorithme du gradient conjugué modifié.

DFP: Davidon-Fletcher-Powell.

BFGS : Broyden-Fletcher-Goldfarb-Shannoa.

LM : Levenberg-Marquardt.

ن العمل المقدم في هـ كره يهدف إلى تحسين مخططات الأشعة التحليلية ات شبكة هوائي ية ن الطريه
في هذا النوع من وإيجاد شدة التيارات

ن التقريب المستعمل يعتمد على الشبكات العصبية ذات معطيات التمرن باستخدام طريقة Dolph-Tchebycheff والخوارزمية
الوراثية ، الاختيار الأمثل لهاته المعطيات تسمح إلى تحسين عمل الشبكات العصبية وتسمح بتقديم سريع للنتائج التحليلية المقارنة مع باقي
التقريبات .

Résumé

Le travail présenté dans ce mémoire se rapporte à l'optimisation de la synthèse des diagrammes de rayonnements de réseau d'antennes linéaires. La méthode de synthèse mise en œuvre pour ce type de réseaux permet d'approcher à un diagramme de rayonnement désiré et trouver les pondérations complexes adéquates.

L'approche utilisée repose sur les réseaux de neurones avec des données d'apprentissage utilisant la méthode de Dolph-Tchebycheff et les algorithmes génétiques. Un choix judicieux de ces données permet d'optimiser au mieux le fonctionnement du réseau de neurones et permettre de générer très rapidement des résultats de synthèse comparée à d'autres approches.

Abstract

The work presented in this memory refers to the optimization of the synthesis diagrams of radiations of network of vertical wire aeriels. The method of synthesis implemented for this type of networks makes it possible to approach with a diagram of desired radiation and to find weightings complex adequate.

The approach used rests on the networks of neurons with data of training using the method of Dolph-Tchebycheff and the algorithms genetic. A judicious choice of these data makes it possible as well as possible to optimize the operation of the network of neurons and to make it possible to very quickly generate results of synthesis compared with other approaches.

Table des matières

LISTE DES FIGURES.....	3
CHAPITRE I : GENERALITES SUR LES ANTENNES	4
I.1 Introduction.....	4
I.2 Historique des antennes	4
I.3 Définition d'une antenne.....	4
I.4 Mécanisme de rayonnement d'une antenne	5
I.5 Paramètres d'une antenne	6
I.5.1 Resistance d'antenne	6
I.5.2 Coefficient d'efficacité d'une antenne	7
I.5.3 Impédance d'une antenne.....	7
I.5.4 Fréquence d'utilisation.....	7
I.5.5 Directivité.....	7
I.5.6 Gain en puissance et rendement	8
I.5.7 Diagramme de rayonnement	8
I.5.8 Lobe principal, lobes secondaires et angle d'ouverture.....	10
I.5.9 Polarisation d'une antenne	10
I.5.10 Surface effective d'une antenne en réception.....	11
I.6 Equation de transmission (FRIIS).....	11
I.7 Classification de types d'antenne.....	12
I.8 Les réseaux d'antennes	13
I.8.1 Facteur de réseau	13
I.8.2 Réseaux linéaires uniformes.....	13
I.8.3 Réseaux plans	15
I.8.4 Réseaux circulaires.....	15
I.9 Influence des paramètres dans la conception des réseaux	16
I.9.1 Effet de la distance d	16
I.9.2 Effet du nombre N de sources	16
I.9.3 Effet du déphasage	17
I.10 Conclusion	17
CHAPITRE II : SYNTHESE DE RESEAUX D'ANTENNES.....	18
II.1 Introduction.....	18
II.2 Position du problème	18
II.2.1 Contraintes sur tout le diagramme	18
II.2.2 Contraintes sur une partie du diagramme.	19
II.2.3 Autre contraintes	20
II.3 Paramètres pouvant être utilisés pour la synthèse	21
II.3.1 Synoptique du logiciel de synthèse.....	21
II.4 Méthodes classiques de synthèse de diagrammes de rayonnement.....	23
II.4.1 Les différentes méthodes	23
II.4.2 Méthode polynomiale de Schelkunoff	24
II.4.3 Méthode de Dolph-Tchebycheff	25
II.5 Méthodes d'optimisation globale	27
II.5.1 Les différents types	27
II.5.2 Algorithmes génétiques.....	28
II.5.3 La méthode du recuit simulé.....	32
II.6 Réseaux de neurones.....	32
II.7 Conclusion	33
CHAPITRE III : les RESEAUX DE NEURONES.....	34
III.1 Introduction	34
III.2 Réseaux de Neurones : Théories fondamentales.....	34

III.3 Le neurone biologique.....	35
III.3.1 Description.....	35
III.3.2 Fonctionnement	36
III.4 Le neurone formel (Artificiel).....	37
III.4.1 Le modèle mathématique.....	37
III.4.2 Fonction de transfert	39
III.5 Réseaux de neurones artificiels	40
III.5.1 Réseaux neuronaux multicouches de type Feed- Forward	40
III.5.2 Réseaux récurrents.....	41
III.5.3 Réseaux cellulaires	41
III.6 Les méthodes d'apprentissages des réseaux de neurones	42
III.6.1 Apprentissage non supervisé	42
III.6.2 Apprentissage supervisé	42
III.7 Le perceptron multicouche	43
III.8 Mise en œuvre des réseaux neuronaux.....	44
III.9 Algorithmes d'apprentissage	45
III.9.1 Retropropagation du gradient	46
III.9.2 Accélération de l'algorithme avec le momentum.....	47
III.9.3 Méthodes d'optimisation du second ordre.....	47
III.10 Boîte à outils Matlab Réseaux de Neurones-(Neural Network toolbox).....	48
III.10.1 Création d'un réseau	48
III.10.2 Apprentissage	49
III.10.3 Simulation (ou activation) d'un réseau.....	49
III.11 Application des réseaux de neurones	49
III.12 Conclusion.....	50
CHAPITRE IV : SIMULATIONS ET COMMENTAIRES	51
IV.1 Introduction	51
IV.2 Application de réseaux de neurones à la synthèse de réseau d'antennes linéaires	51
IV.2.1 Procédure de développement d'un réseau de neurones	51
IV.2.2 Algorithme et paramètres d'apprentissage	53
IV.2.3 Validation et résultats de simulation	53
IV.3 Programmation de la méthode de Dolph-Tchebycheff.....	53
IV.4 Simulation des réseaux de neurones avec apprentissage par Dolph	56
IV.5 Simulation des réseaux de neurones avec apprentissage par algorithmes génétiques	63
IV.6 Conclusion.....	65
CONCLUSION GENERALE.....	67
ANNEXES.....	68
BIBLIOGRAPHIE.....	69

LISTE DES FIGURES

Fig. I.1 : Antennes en émission et en réception	5
Fig. I.2 : Champ magnétique créée par un courant I	5
Fig.I.3 : Champ électriques créent par un courant I dans un conducteur	5
Fig.I.4 : Onde électromagnétique émise à partir d'un conducteur alimenté en courant non continu.	6
Fig.I.5 Diagramme en trois dimensions	9
Fig.I.6 : Diagramme en coordonnées polaires	9
Fig.I.7 : Diagramme en coordonnées cartésiennes (échelle logarithmique)	9
Fig.I.8 : Ouverture mi-puissance	10
Fig.I.9 : Les différentes polarisations d'une antenne	10
Fig.I.10 : Liaison radio	11
Fig.I.11 : Types d'antennes	12
Fig.I.12 : Configuration d'un réseau linéaire uniforme de N sources	14
Fig.I.13 : Réseau plan MxN	15
Fig.I.14 : Réseau circulaire de N éléments	16
Fig.I.15 : Effet du nombre de sources	17
Fig.I.16 : Effet de la phase	17
Fig.II.1: Définition des spécifications sur le diagramme de rayonnement à partir d'un Gabarit	18
Fig.II.2 : Exemple de gabarit avec zone d'annulation	19
Fig.II.3 : Exemple de gabarit avec largeur de lobe principal fixée	20
Fig.II.4 : Synoptique d'un logiciel de synthèse	21
Fig.II.5 : Méthode à placement de zéros	24
Fig.II.6 : Diagramme par la méthode Dolph (25 antennes et $R = 60\text{dB}$)	25
Fig.II.7 : Relation entre polynôme de Tchebycheff et diagramme de rayonnement	26
Fig.II.8 : (a) Recherche locale (b) recherche globale ou p est la probabilité de Transition	27
Fig.II.9 : Classification des méthodes d'optimisation	27
Fig.II.10: Structure générale des algorithmes génétiques	29
Fig.III.1: Les neurones schématisés	35
Fig.III.2: Schéma fonctionnel du neurone forme	37
Fig.III.3: Fonction signe	38
Fig.III.4 : Fonction sigmoïde	38
Fig.III.5 : Structure feed-forward de type MLP	39
Fig.III.6 : Structure d'un réseau neuronal récurrent	40
Fig.III.7 : Réseau cellulaire	40
Fig.III.8 : Apprentissage non supervisé	41
Fig.III.9 : Apprentissage supervisé	41
Fig.III.10 : Perceptron multicouche	42
Fig.IV.1 : Organigramme de synthèse par la méthode Dolph-Tchebycheff	52

Fig.IV.2 : Diagramme pour N=8 et R=20dB	53
Fig.IV.3 : Diagramme pour N=8 et R=30dB	53
Fig.IV.4 : Diagramme pour N=8 et R=40dB	53
Fig.IV.5 : Phase d'apprentissage	54
Fig.IV.6 : Evolution de la fonction d'erreur	55
Fig.IV.7 : Phase de test	55
Fig.IV.8 : Test pour R=20,30,40 et 50dB ; N=8	56
Fig.IV.9 : Test pour R=25dB N=8	57
Fig.IV.10 : Test pour R=35 N=8	57
Fig.IV.11 : Test pour R=45 N=8	57
Fig.IV.12 : Test pour R=55 N=8	58
Fig.IV.13 : Test pour R=20,30,40 et 50dB ,N=16	58
Fig.IV.14 : Test pour R=25dB N=16	59
Fig.IV.15 : Test pour R=35dB N=16	59
Fig.IV.16 : Test pour R =45dB N=16	60
Fig.IV.17 : Test pour R=55dB N=16	60
Fig.IV.18 : Test pour Nul=17.3° et N=16	61
Fig.IV.19 : Test pour Nul=23.5° et N=16	61
Fig.IV.20 : Test pour Nul=33.5° et N=16	62
Fig.IV.21 : Test pour R=20,30,40 et 50dB ;N=16	62

LISTE DES TABLEAUX

Tab III.1 : Analogie entre neurone biologique et artificiel	37
Tab IV.1 : Coefficient pour N=8 et R=20dB	54
Tab IV.2: Coefficient pour N=8 et R=30dB	54
Tab IV .3: Coefficient pour N=8 et R=40dB	54
Tab IV.4 : Pondération Dolph et NN R=25dB N=8	58
Tab IV.5 : Pondération Dolph et NN R=35dB N=8	58
Tab IV.6 : Pondération Dolph et NN R=45dB N=8	58
Tab IV.7 : Pondération Dolph et NN R=55dB N=8	59
Tab IV.8 : Pondération Dolph et NN R=25dB N=16	60
Tab IV.9 : Pondération Dolph et NN R=35dB N=16	60
Tab IV.10 : Pondération Dolph et NN R=45dB N=16	61
Tab IV.11 : Pondération Dolph et NN R=55dB N=16	61
Tab IV.12 : Pondération AG et NN Nul=17,3° N=16	62
Tab IV.13 : Pondération AG et NN Nul=23,5° N=16	62
Tab IV.14 : Pondération AG et NN Nul=33,5° N=16	63

INTRODUCTION GENERALE

Dans le cadre du présent travail pour l'obtention du diplôme d'ingénieur en aéronautique, nous sommes intéressés au domaine de la synthèse de diagramme de rayonnement des réseaux d'antennes linéaires avec l'objectif principal de donner un outil efficace pour l'optimisation de ces derniers.

Une telle simulation nécessite non seulement la connaissance des antennes et leur utilisation mais aussi les méthodes d'optimalisation pour résoudre les problèmes liés à leurs performances.

Plusieurs outils ont été développés pour la synthèse de diagramme de rayonnement des antennes telles que Fourier, Dolphe-Tchebycheff, Woodward- Lawson, Newton, gradient, etc. Compte tenu de la diversité des buts recherchés par les utilisateurs, on ne trouvera pas une méthode générale de synthèse applicable à tous les cas, mais plutôt un nombre important de méthodes propres à chaque type de problème.

Dans ces dernières décennies, de nouvelles méthodes ont été développées pour divers problèmes liées à l'industrie telle que les réseaux de neurones et les algorithmes génétiques (AG).

Les réseaux de neurones sont composés d'éléments simples (ou neurones) fonctionnant en parallèle. Ces éléments ont été fortement inspirés par le système nerveux biologique. Comme dans la nature, le fonctionnement du réseau (de neurone) est fortement influencé par la connections des éléments entre eux. On peut entraîner un réseau de neurone pour une tâche spécifique (reconnaissance de caractères par exemple) en ajustant les valeurs des connections (ou poids) entre les éléments (neurone).

L'objectif de notre étude est d'élaborer une technique de synthèse pour ce type de réseaux qui permet d'approcher un diagramme de rayonnement désiré et de trouver les pondérations complexes pour que la fonction réponde à des spécifications précises.

L'approche d'apprentissage des réseaux de neurones utilisée est basée sur la méthode de Dolphe-Tchebycheff et les algorithmes génétiques.

Le mémoire est divisé en quatre chapitres :

Le premier chapitre est une généralité sur les antennes on donne une idée sur le mécanisme de rayonnement des antennes, les différents paramètres d'une antenne et un certain nombre de types d'antennes, et puis on décrit les différents types de réseaux d'antenne rencontrés en pratique et leurs caractéristiques.

Les différentes techniques de synthèse des réseaux d'antennes à partir de spécification imposées sur le diagramme d'antenne sont présentées dans le deuxième chapitre.

Dans le troisième chapitre nous décrivons en détails les concepts fondamentaux et les notions liées aux réseaux neuronaux (neurone biologique, modèle mathématique, type de réseaux, les méthodes d'apprentissages). Et puis nous développons l'algorithme de rétro-propagation, qui est adopté comme outil d'apprentissage du réseau neuronal multicouche de type Feed-forward (MLP).

En fin, les résultats de simulation et leurs interprétations seront présentés dans le quatrième chapitre.

Finalement, une conclusion resumerà les principaux résultats auxquels nous avons abouti dans cette étude et quelques perspectives ultérieures seront tirées.

CHAPITRE I : GENERALITES SUR LES ANTENNES

I.1 Introduction

Dans ce premier chapitre, on donne une idée sur le mécanisme de rayonnement d'une antenne ainsi que les différents paramètres d'une antenne et les différentes méthodes de classification des différents types d'antennes, puis on décrira les différents types de réseaux d'antenne rencontrés en pratique et leur caractéristiques.

I.2 Historique des antennes

Les premiers systèmes de communication furent la télégraphie introduite en 1844, suivie par la téléphonie en 1878. Dans ces systèmes, les signaux sont transmis par des lignes de transmission.

La théorie des antennes intervient après des développements mathématiques de James Maxwell, en 1854, corroborés par les expériences de Hertz en 1887 et les premiers systèmes de radiocommunication de Marconi en 1901 (première communication trans-océanique sans fils entre l'Angleterre et le Canada sur une fréquence de 820 kHz).

En 1888 Heinrich Hertz utilisa pour la première fois des antennes pour démontrer l'existence des ondes électromagnétiques prédites par la théorie de Maxwell. Il utilisa des antennes doublet, tant pour la réception que pour l'émission. Il installa même le dipôle émetteur au foyer d'un réflecteur parabolique.

En 1916, Marconi réalisa des expériences avec des signaux de 2 à 3 m de longueur d'onde en utilisant des réflecteurs paraboliques. Dans la décade 1910-1920, de nouvelles techniques ont vu le jour, comme les aides à la navigation aérienne, les communications sous-marines, les systèmes de commande à distance.

I.3 Définition d'une antenne

Une antenne en émission est un élément servant de transition entre une onde guidée (dans une ligne coaxiale ou guide d'onde) et l'espace libre. De par sa structure, elle permet de détachement des champs électromagnétique de leur support métallique et leur rayonnement dans l'espace. Une antenne en réception effectue l'opération inverse (figure I.1).

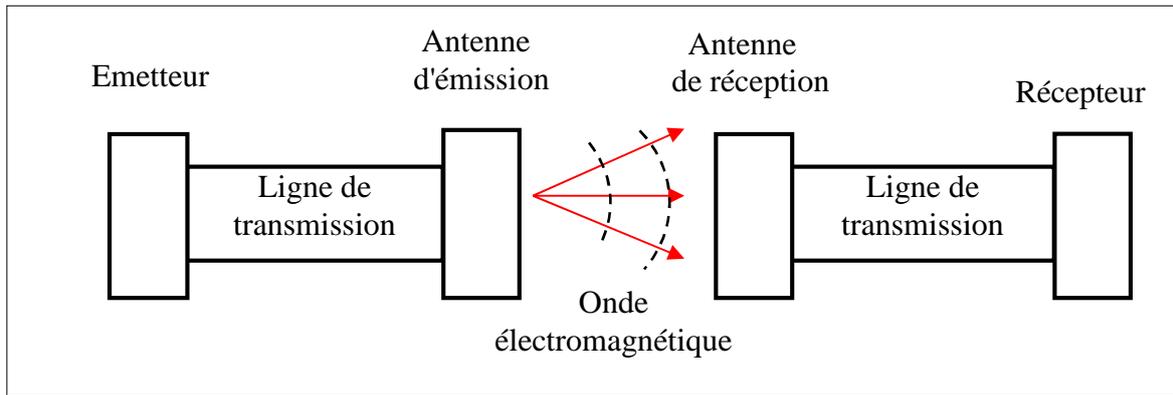


Fig. I.1 : Antennes en émission et en réception

I.4 Mécanisme de rayonnement d'une antenne

Un conducteur dans lequel passe un courant I produit dans son entourage un champ magnétique H .

Le sens du champ magnétique est conforme à la représentation donnée dans la figure I.2.

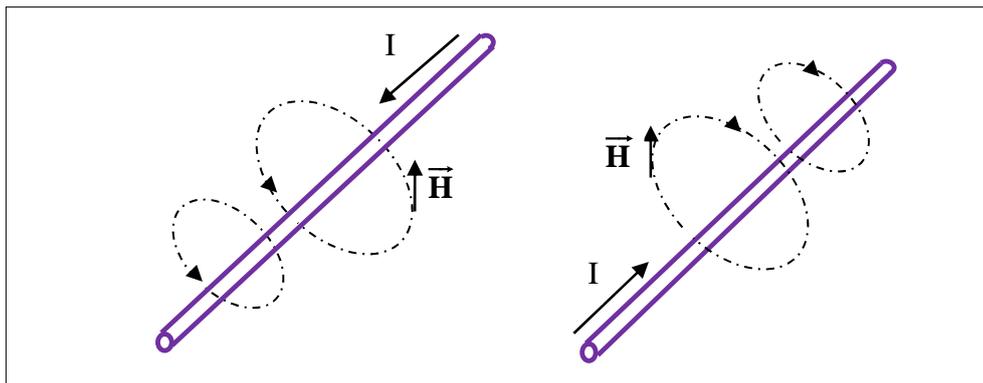


Fig. I.2 Champ magnétique crée par un courant I

D'autre part, un courant circulant dans un conducteur correspond à un déplacement de charges dans le conducteur. Nous pourrions donc considérer que l'effet provoqué par un courant est d'accumuler des charges positives à l'une des extrémités du conducteur et des charges négatives à l'autre, celles-ci donneront lieu à des lignes de champs électriques telles que représentées dans la figure I.3.

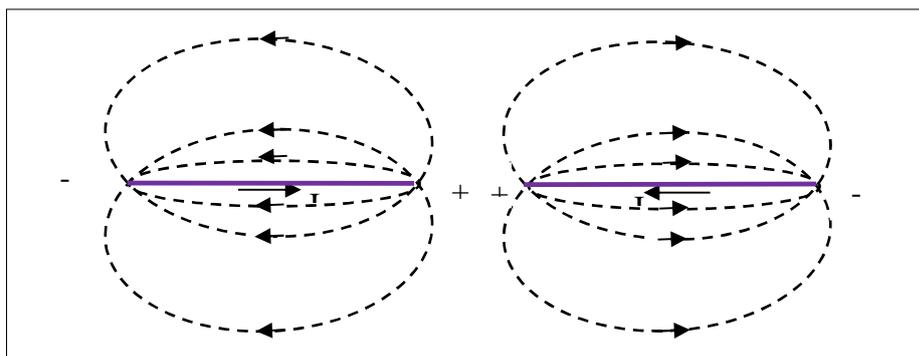


Fig. I. 3 champs électriques créent par un courant I dans un conducteur

Il existe donc des champs E et H perpendiculaires l'un de l'autre. Si les champs sont variables (courant variable), ceci donne naissance à une onde électromagnétique qui se propage perpendiculairement aux deux champs.

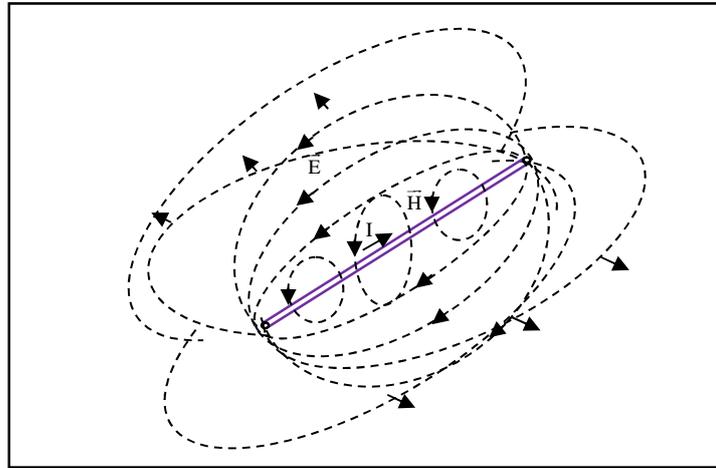


Fig. I. 4 : Onde électromagnétique émise à partir d'un conducteur alimenté en courant non continu

I.5 Paramètres d'une antenne

Une antenne peut être caractérisée par différents paramètres : sa résistance, son impédance, sa fréquence, sa directivité, son gain, son diagramme de rayonnement, sa largeur de faisceau, sa polarisation, sa surface effectif.

I.5.1 Résistance d'antenne

La résistance d'une antenne est un facteur essentiel à connaître parce qu'il permet d'adapter celle-ci à la résistance d'entrée de l'émetteur et assurer ainsi un transfert maximal d'énergie.

Une puissance électrique fournie à une antenne sera d'une part perdue en échauffement résultant de la résistance ohmique de l'antenne et, d'autre part, traduite en rayonnement électromagnétique. Nous pourrions donc considérer la résistance totale de l'antenne comme étant composée de deux résistances en série : la résistance ohmique R et la résistance de rayonnement R_r . une antenne sera d'autre plus efficace que le rapport R_r/R sera grand.

R_r étant défini comme le rapport entre la puissance totale rayonnée par l'antenne et le carré de la valeur efficace du courant à son entrée.

$$R_r = \frac{P_r}{I^2} \quad (\text{I.1})$$

R étant défini comme le rapport entre la puissance dissipée par effet de joule et le carré de la valeur efficace du courant à son entrée.

$$R_\Omega = \frac{P_\Omega}{I^2} \quad (\text{I.2})$$

I.5.2 Coefficient d'efficacité d'une antenne

Le coefficient d'efficacité d'une antenne est défini par le rapport :

$$\eta = \frac{R_r}{R_\Omega + R_r} \quad (\text{I.3})$$

I.5.3 Impédance d'une antenne

L'impédance d'une antenne se définit comme le rapport entre la tension et le courant à son entrée. En général, cette impédance est complexe : la partie réelle s'appelle résistance, la partie imaginaire réactance.

$$Z = \frac{V}{I} = R_r + jX \quad (\text{I.4})$$

I.5.4 Fréquence d'utilisation

Une antenne s'utilise en général avec des signaux autour d'une fréquence donnée pour laquelle l'antenne possède des capacités optimales pour émettre ou recevoir l'énergie électromagnétique correspondante dans l'espace environnant. La fréquence de résonance d'une antenne dépend d'abord de ses dimensions propres, mais aussi des éléments qui lui sont ajoutés. Par rapport à la fréquence de résonance centrale de l'antenne, un affaiblissement de 3dB détermine les fréquences minimum et maximum d'utilisation ; la différence entre ces deux fréquences correspond à la bande passante.

I.5.5 Directivité

La directivité (parfois appelée gain directionnel) d'une antenne est définie comme le rapport de la densité de puissance surfacique $P(r, \theta, \varphi)$ rayonnée par l'antenne et la densité de puissance surfacique P_{iso} rayonnée par l'antenne isotrope.

L'antenne isotrope est une antenne fictive émettant la même puissance dans toutes les directions.

$$D(\theta, \varphi) = \frac{P(r, \theta, \varphi)}{P_{\text{iso}}} \quad (\text{I.5})$$

Qui peut aussi s'écrire :

$$D(\theta, \varphi) = \frac{4\pi r^2 P(r, \theta, \varphi)}{P_r} \quad (\text{I.6})$$

P_r étant la puissance totale rayonnée par l'antenne et r la distance.

La directivité indique dans quelles directions la densité de puissance est meilleure ou moins bonne que celle de l'antenne isotrope.

I.5.6 Gain en puissance et rendement

Le gain se définit de la même manière que la directivité mais en tenant compte des pertes, c'est-à-dire en remplaçant la puissance rayonnée par la puissance fournie P_F à l'antenne :

$$G(\theta, \varphi) = \frac{4\pi r^2 P(r, \theta, \varphi)}{P_F} \quad (\text{I.7})$$

Le rendement d'une antenne est défini ainsi

$$\eta = \frac{G(r, \theta, \varphi)}{D(r, \theta, \varphi)} = \frac{P_r}{P_F} (< 1) \quad (\text{I.8})$$

Le gain en puissance dépend de la fréquence du signal émis. Il est très faible aux basses fréquences mais atteint des valeurs de 75% à 95% pour des fréquences supérieures à 1 MHz.

On utilise surtout le gain maximal $G = \max [G(\theta, \varphi)]$

Les plus grands gains sont obtenus avec des antennes paraboliques (20 à 60 dB).

On déduit la densité de puissance maximale d'une antenne par rapport à la puissance fournie P_F :

$$P(r, \theta, \varphi) = G \frac{P_F}{4\pi r^2} \quad (\text{W} / \text{m}^2) \quad (\text{I.9})$$

De fait, le gain en puissance d'une antenne est un paramètre plus intéressant que sa directivité; en effet, comme le rendement est inclus, on peut calculer la densité de puissance en introduisant directement la valeur de la puissance fournie P_F .

I.5.7 Diagramme de rayonnement

C'est la représentation du rapport des gains en puissance par rapport au gain max $G(\theta, \varphi) / G_{\max}$. Parfois, c'est tout simplement la représentation du gain $G(\theta, \varphi)$ lui-même.

On utilise aussi le diagramme champ en remplaçant le gain par le champ électrique

Différentes types de représentation de diagramme sont utilisées :

- Représentation en trois dimensions qui donne une idée de la répartition de la puissance rayonnée dans l'espace (fig. I.5)
- Représentation en coordonnées polaires dans des plans choisis (fig. I.6)
- Représentation en coordonnées cartésiennes (échelle linéaire)
- Représentation en coordonnées cartésiennes (échelle logarithmique) : cette échelle est habituellement souhaitable parce qu'une échelle logarithmique peut accentuer les détails des parties du diagramme qui ont des valeurs très basses (fig. I.7)

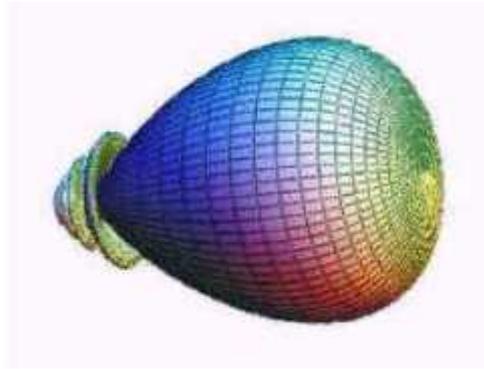


Fig. I.5 : Diagramme en trois dimensions

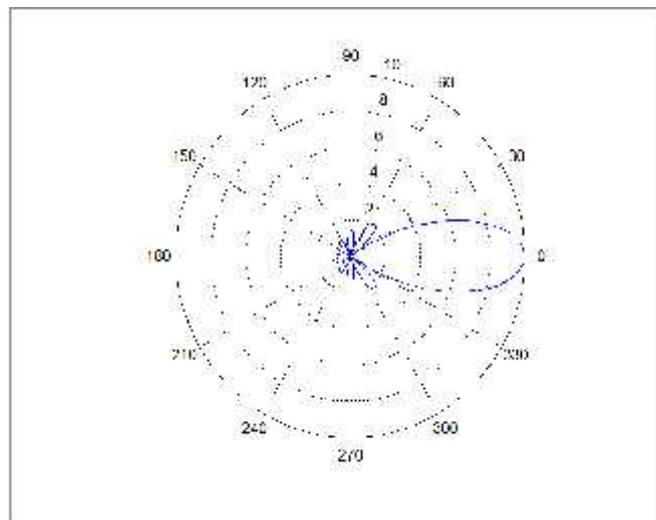


Fig. I.6 : Diagramme en coordonnées polaires

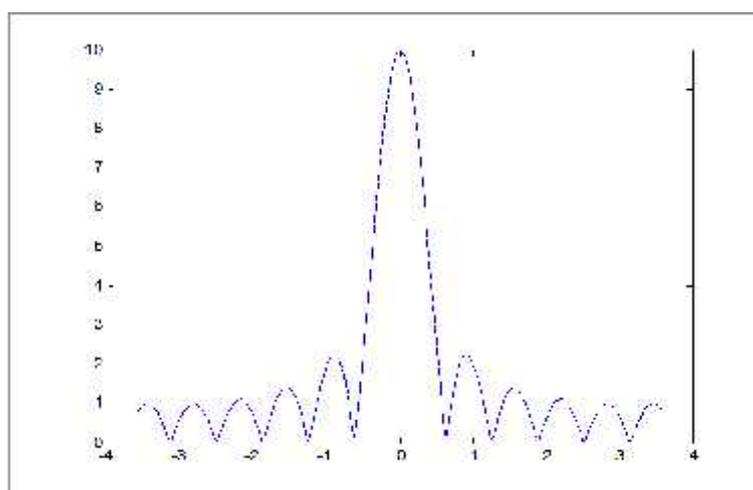


Fig. I.7 : Diagramme en coordonnées cartésiennes (échelle logarithmique)

I.5.8 Lobe principal, lobes secondaires et angle d'ouverture

Généralement dans un plan donné, le diagramme de rayonnement d'une antenne directive comprend un lobe principal et des lobes secondaires qui sont indésirables.

La plus grande partie de la puissance rayonnée est contenue dans un cône d'ouverture HPBW (Half Power Beam Wide) défini à mi-puissance, ce qui correspond à une chute de 3 dB par rapport à la puissance maximale ou à un champ max divisé par $\sqrt{2}$ (figure I.8).

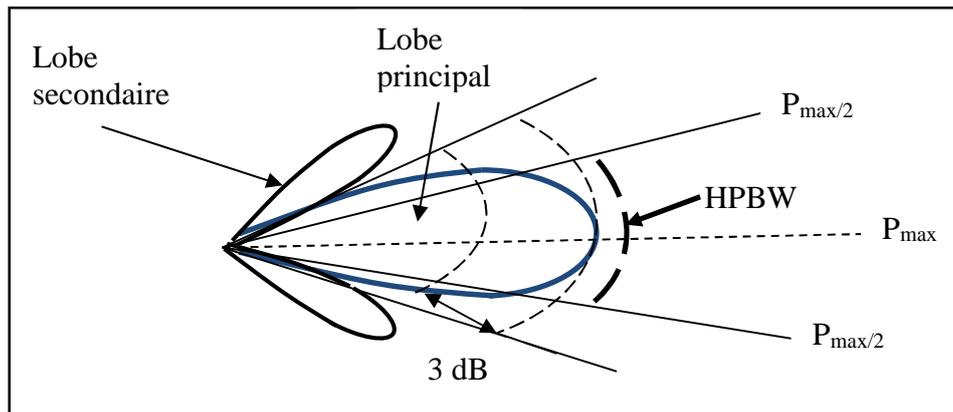


Fig. I.8 : Ouverture mi-puissance

I.5.9 Polarisation d'une antenne

La polarisation d'une antenne dans une direction donnée est définie comme « la polarisation de l'onde transmise (rayonnée) par l'antenne, c'est-à-dire de la direction du champ électrique émis par l'onde. Quand la direction n'est pas énoncée, la polarisation est prise pour être la polarisation dans la direction du gain maximum » Dans la pratique, la polarisation de l'énergie rayonnée varie avec la direction de l'antenne, de sorte que les différentes parties du diagramme puissent avoir différentes polarisations.

Les antennes peuvent émettre des ondes à polarisation (Figure I.9):

- Linéaire : le champ électrique conserve une direction constante
- Circulaire : l'extrémité du champ électrique décrit un cercle lors de sa propagation
- Elliptique : l'extrémité du champ électrique décrit une ellipse lors de sa propagation

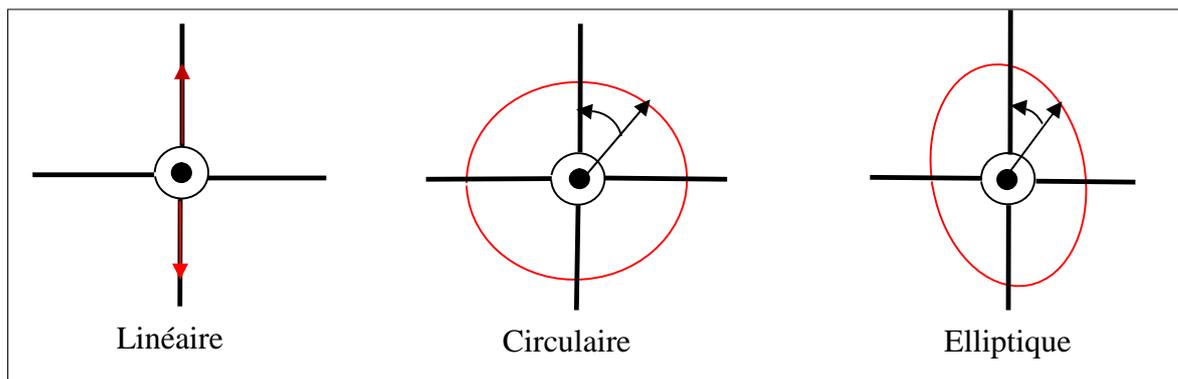


Fig. I.9 : Les différentes polarisations d'une antenne

I.5.10 Surface effective d'une antenne en réception

A chaque antenne, nous pouvons associer une surface effective. Celle-ci est employée pour calculer la puissance captée par une antenne. La surface effective (ou équivalente) A_e d'une antenne dans une direction donnée est définie comme « le rapport de la puissance disponible sur les bornes de l'antenne en réception P_{rec} à la puissance de l'onde plane incidente $P(r, \theta, \varphi)$ sur l'antenne dans cette direction :

$$A_e = \frac{P_{rec}}{P(r, \theta, \varphi)} \quad (\text{m}^2) \quad (\text{I.10})$$

La surface effective de n'importe quelle antenne est reliée à son gain max G par la relation importante :

$$A_e = \frac{\lambda^2}{4\pi} G \quad (\text{m}^2) \quad (\text{I.11})$$

Étant la longueur d'onde du signal émis par l'antenne.

Cette relation découle du théorème de réciprocité qui stipule qu'une antenne en émission conserve les mêmes caractéristiques en réception.

I.6 Equation de transmission (FRIIS)

L'équation de transmission relie la puissance captée P_{rec} en fonction des caractéristiques de la liaison émetteur-récepteur (figure I.10):

$$P_{rec} = P_F G_E G_R \left(\frac{\lambda}{4\pi r} \right)^2 \quad (\text{W}) \quad (\text{I.12})$$

Où

P_F : La puissance fournie à l'antenne d'émission

G_E : Gain de l'antenne d'émission

G_R : Gain de l'antenne de réception

r : la distance entre l'émetteur et le récepteur

Notons que cette équation ne tient pas compte des effets de désadaptation des antennes, ni des pertes atmosphériques.

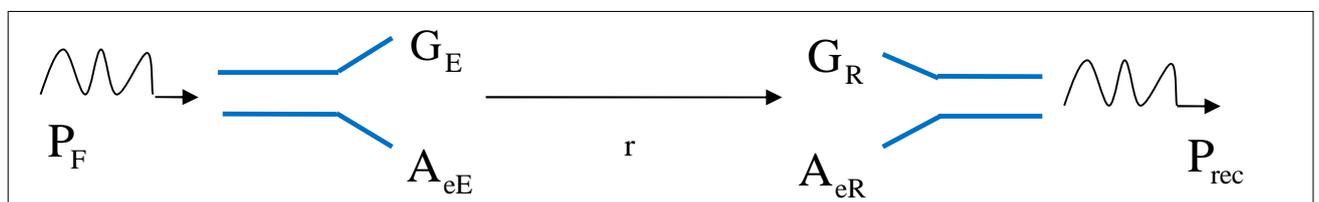


Fig. I.10 : Liaison radio

I.7 Classification de types d'antenne

Il existe un grand nombre de types d'antennes (figure I.11). Elles peuvent être classifiées de différentes manières.

■ Formes ou géométries

Antennes filaires : dipôle, boucle, spirale

Antennes d'ouverture : cornet, fente

Antennes imprimées : patch, dipôle imprimé, spirale

■ Gain

Gain élevé : plat

□ Gain moyen : cornet

□ Gain faible : dipôle, boucle, fente, patch

■ Formes de faisceau

Omnidirectionnel : dipôle

Faisceau de crayon : plat

Faisceau de fan : rangée

■ Largeur de bande

Large bande : hélix

Bande étroite : patch

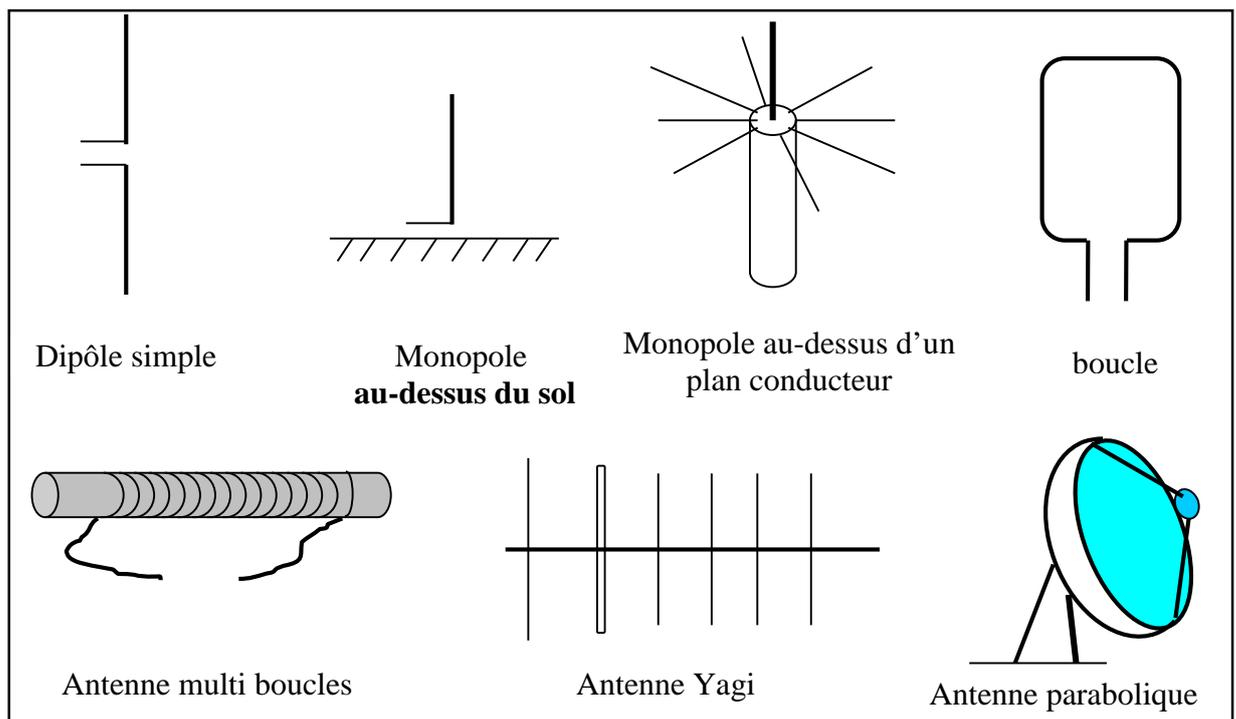


Fig. I.11 : Types d'antennes

I.8 Les réseaux d'antennes

Des antennes peuvent être arrangées dans l'espace, selon différentes configurations géométriques, pour obtenir des diagrammes très directifs. Ces configurations d'antennes s'appellent des réseaux.

L'avantage principal des réseaux d'antennes par rapport à une antenne simple est leurs capacités de balayage électronique, c'est-à-dire, que le lobe principal peut être orienté vers n'importe quelle direction en changeant la phase du courant d'excitation de chaque élément du réseau (antennes à réseaux déphasés). En outre, en agissant également sur les amplitudes des courants d'excitation, une grande variété de diagrammes de rayonnement et des caractéristiques de niveau des lobes secondaires peuvent être produites.

Il y a cinq paramètres principaux qui affectent les performances globales d'un réseau d'antennes :

- la géométrie du réseau (par exemple, linéaire, circulaire, ou planaire des éléments rayonnants),
- distances de séparation entre les éléments adjacents,
- amplitudes des courants d'excitation de chaque élément,
- phase de chaque élément rayonnant,
- diagramme de rayonnement de chaque élément.

I.8.1 Facteur de réseau

Le facteur de réseau est un outil qui nous sert à caractériser le retard successive pour chaque antenne par rapport à l'origine. chaque réseau peut être caractérisé par ce qu'on appelle facteur de réseau .

Le facteur de réseau est une fonction des paramètres suivants :

- l'arrangement géométrique des éléments rayonnants du réseau
- les courants d'excitation des éléments
- le nombre d'éléments
- la distance de la séparation d des éléments adjacents
- fréquences (ou longueur d'onde) d'opération

D'une façon générale, le diagramme d'un réseau est donné par le diagramme de rayonnement de l'élément simple multiplié par le facteur de réseau.

I.8.2 Réseaux linéaires uniformes

Parmi les réseaux les plus répandus, on trouve les réseaux uniformes linéaires, sphériques, cylindriques, circulaires.

Considérons maintenant un réseau de N éléments de sources isotropes représentés sur la figure I.12.

C'est un réseau linéaire puisque tous les éléments sont placés suivant une ligne et uniforme parce que les éléments identiques sont alimentés avec un courant de même amplitude mais avec un déphasage progressif ϕ . La distance de séparation entre les éléments adjacents est d . Dans ce cas-ci le facteur de réseau peut être exprimé comme la somme des contributions de chaque élément :

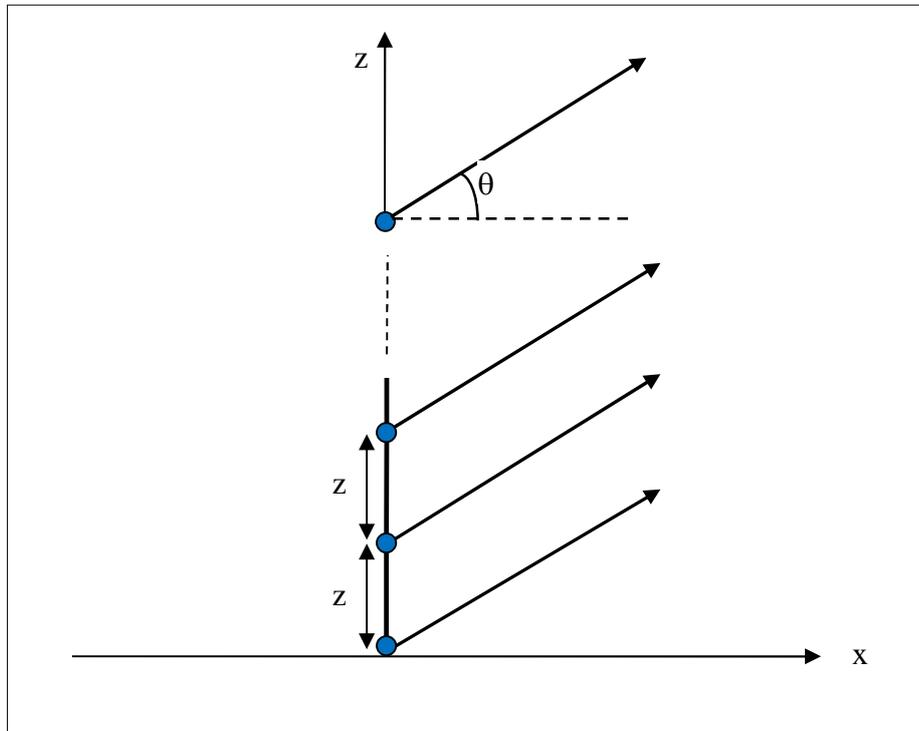


Fig. I.12: Configuration d'un réseau linéaire uniforme de N sources

$$F(\psi) = 1 + e^{j\psi} + e^{2j\psi} + \dots + e^{j(N-1)\psi} = \frac{\sin\left(\frac{N}{2}\psi\right)}{\sin\left(\frac{1}{2}\psi\right)} \tag{I.13}$$

ou $\psi = \frac{2\pi}{\lambda} \sin \theta + \phi$ (I.14)

L'examen de cette équation permet de faire ressortir les points suivants concernant le facteur de réseau d'un réseau linéaire uniforme :

- Le maximum principal (lobe principal) se produit quand $\psi = 0$, soit :

$$\theta_{\text{MAX}} = \arcsin\left(-\frac{\lambda\psi}{2\pi d}\right) \tag{I.15}$$

- Les nuls se produisent quand $\sin(N/2\psi) = 0$, soit :

$$(N/2)\psi = \pm n\pi \quad \text{pour} \quad n = 1, 2, 3, \dots$$

I.8.3 Réseaux plans

A la différence des réseaux linéaires qui peuvent seulement balayer dans un plan (ou), les réseaux plans peuvent balayer dans les deux plans (et).

Les réseaux plans offrent un gain plus élevé et des lobes secondaires plus bas aux dépens d'un nombre d'éléments plus grand.

Les principes de conception des réseaux plans sont semblables à ceux présentés pour les réseaux linéaires. Puisque les éléments sont placés dans deux dimensions (fig. I.13), le facteur de réseau d'un réseau plan peut être exprimé comme le produit des facteurs de réseau de deux réseaux linéaires : l'un situé le long de l'axe des abscisses et l'autre le long de l'axe des ordonnées :

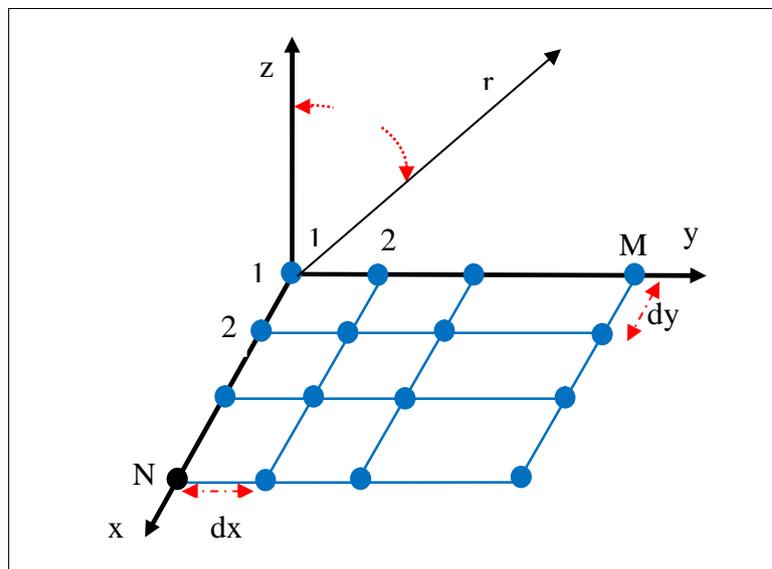


Fig. I.13: Réseau plan MxN

Le facteur de directivité du réseau est donné par :

$$F = \left(\frac{\sin(N\psi_x/2)}{N \sin(\psi_x/2)} \right) \left(\frac{\sin(M\psi_y/2)}{M \sin(\psi_y/2)} \right) \tag{I.16}$$

où

$$\psi_x = \frac{2\pi}{\lambda} d_x \cos \theta \cos \phi + \phi_x$$

$$\psi_y = \frac{2\pi}{\lambda} d_y \sin \theta \sin \phi + \phi_y$$

I.8.4 Réseaux circulaires

Dans une configuration de réseau circulaire, les éléments sont placés sur un cercle, comme, représenté sur la figure I.14.

Le facteur de réseau dans ce cas est donné par :

$$F(\theta, \phi) = \sum_{n=1}^N I_n e^{jKa(\sin\theta \cos(\phi - \phi_n) - \sin\theta_0 \cos(\phi_0 - \phi_n))} \tag{I.17}$$

Où θ_0 et ϕ_0 sont les angles du faisceau principal, a est le rayon du réseau circulaire et I_n les courants d'excitation des éléments.

La figure I.14 représente un réseau circulaire de 12 éléments avec un espacement de $(\lambda/2)$.

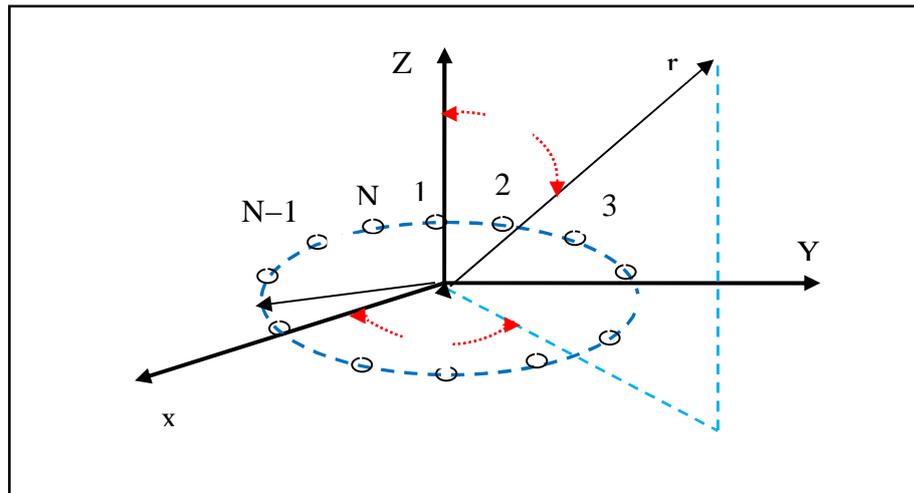


Fig. I.14: Réseau circulaire de N éléments

I.9 Influence des paramètres dans la conception des réseaux

I.9.1 Effet de la distance d

En général, on ne doit pas avoir un dédoublement du lobe principal dans la zone balayée.

Le nombre de lobes principaux dans l'espace réel dépend de la distance d entre les sources :

- Si $(d < \lambda/2)$, il y a un seul lobe principal
- Si $(d = \lambda/2)$, il y a plus d'un lobe principal. Les lobes en plus s'appellent lobes de réseau.
- Lorsque $(\lambda/2 < d < \lambda)$, l'apparition de lobes de réseau dépend du déphasage α . Pour l'éviter, on doit avoir ;

$$d < \frac{\lambda}{|1 + \cos \theta_{MAX}|} \tag{I.18}$$

I.9.2 Effet du nombre N de sources

Pour une distance d constante, l'augmentation de N augmente la directivité et le nombre de lobes secondaires et diminue donc l'ouverture du lobe principal (figure I.15).

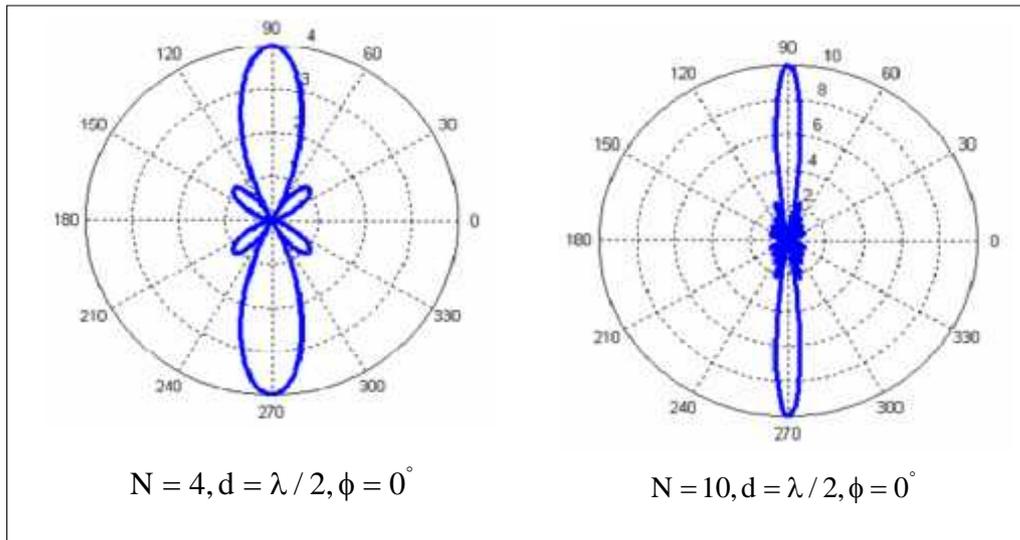


Fig. I.15 : Effet du nombre de sources

I.9.3 Effet du déphasage

Une phase variable permet d’orienter le lobe de broadside ($MAX=\pm 90^\circ$) à endfire ($MAX=+180^\circ$) (figure I.16). A partir d’un réseau donné, on peut donc faire varier, dynamiquement ou non, l’orientation privilégiée du gain en jouant sur les alimentations des divers éléments : c’est le principe des antennes à balayage électronique.

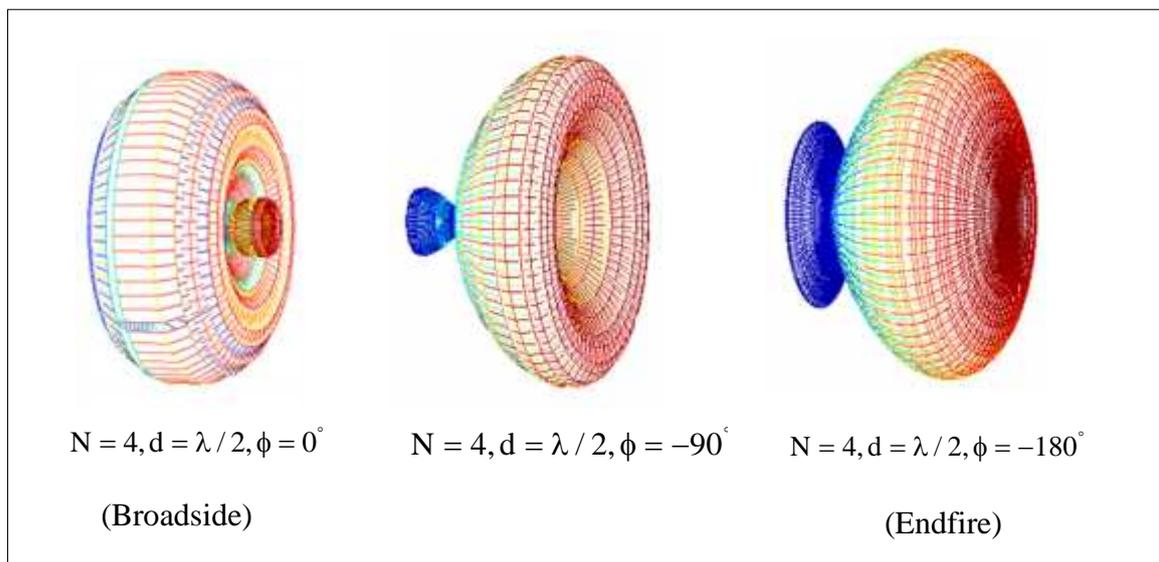


Fig. I.16 : Effet de la phase

I.10 Conclusion

Dans cette première partie nous avons développé certaines notions très importantes sur les antennes, comme le gain et le diagramme de rayonnement qui nous seront utiles dans la suite de notre travail.

CHAPITRE II : SYNTHÈSE DE RESEAUX D'ANTENNES

II.1 Introduction

Ce chapitre sera consacré à donner un aperçu des méthodes de synthèse de diagrammes de rayonnement utilisées à l'heure actuelle. On verra tout d'abord la notion de gabarit, nous décrirons ensuite quelques méthodes classiques qui seront utilisées dans notre travail, et finalement nous verrons les méthodes modernes qu'on est entrain de développer de plus en plus.

II.2 Position du problème

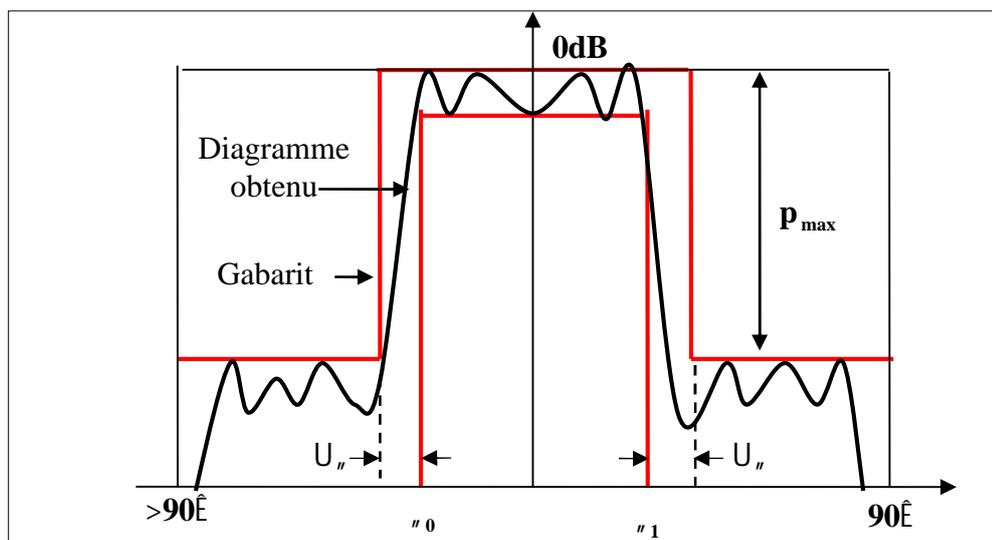
Les problèmes envisagés dans la synthèse sont tous du type : trouver le vecteur de pondération des courants d'excitation I satisfaisant un certain nombre de contraintes, ou si cela n'est pas possible, trouver le vecteur I dont la distance aux contraintes est la plus petite (minimisation d'une fonction de coût).

En ce qui concerne le diagramme de rayonnement du réseau, les contraintes peuvent être classées de manière générale en deux classes :

- contraintes sur tout le diagramme
- contraintes sur une partie du diagramme

II.2.1 Contraintes sur tout le diagramme

On impose
diagramme
respecter



au
de

simultanément plusieurs contraintes, comme le niveau de

Fig. II.1: Définition des spécifications sur le diagramme de rayonnement à partir d'un gabarit.

lobes secondaires, la largeur du lobe secondaire, le niveau d'oscillation du lobe principal.

Ces contraintes sont traduites par un gabarit (figure II.1)

II.2.2 Contraintes sur une partie du diagramme.

- Elimination d'un signal parasite : on veut annuler, sur une petite bande angulaire, l'émission ou la réception d'un signal. On relâchera si besoin les contraintes sur la largeur du lobe principal pour que le gabarit soit satisfait en donnant la priorité à la zone où on veut l'annulation.

L'exemple d'un tel gabarit satisfait est donné ci-dessous en figure II.2:

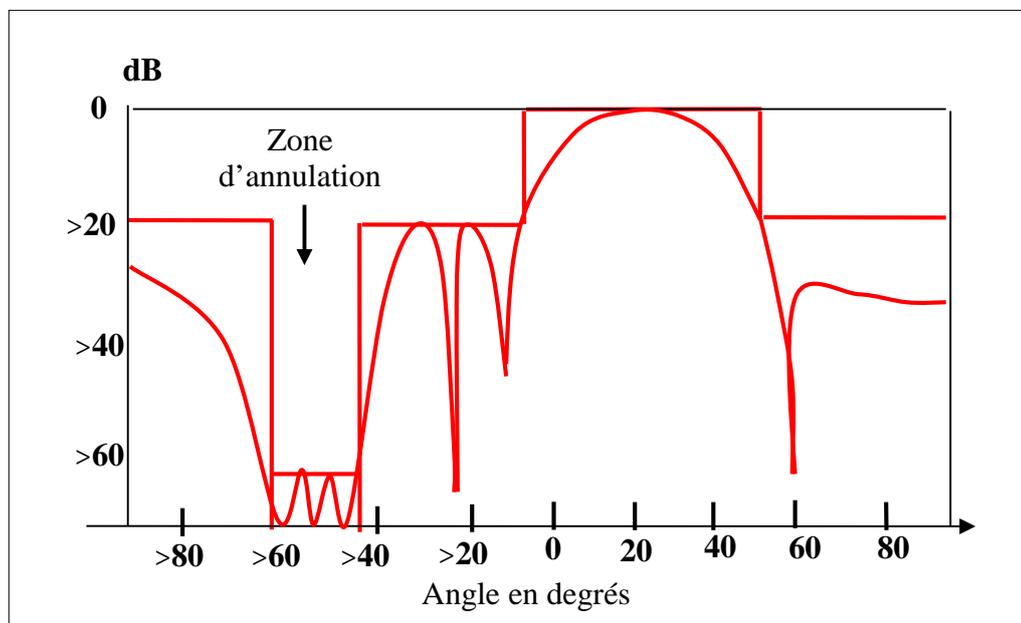


Fig. II.2 : Exemple de gabarit avec zone d'annulation

- lobes secondaires de bas niveau : on veut souvent que le niveau des lobes secondaires soit le plus bas possible. On mettra donc le paramètre p_{\max} à une valeur basse, quitte à relâcher les autres valeurs traduisant les contraintes pour que la contrainte p_{\max} soit satisfaite.
- largeur de lobe principal: on rappelle que la largeur du lobe principal est définie par la valeur des paramètres pour laquelle la fonction puissance $p(\theta, \varphi)$ est à -3dB au dessous de sa valeur maximale. Dans le cas où on veut fixer cette valeur, on choisira les paramètres du gabarit pour satisfaire cette contrainte. L'exemple d'un tel diagramme est représenté en figure II.3.

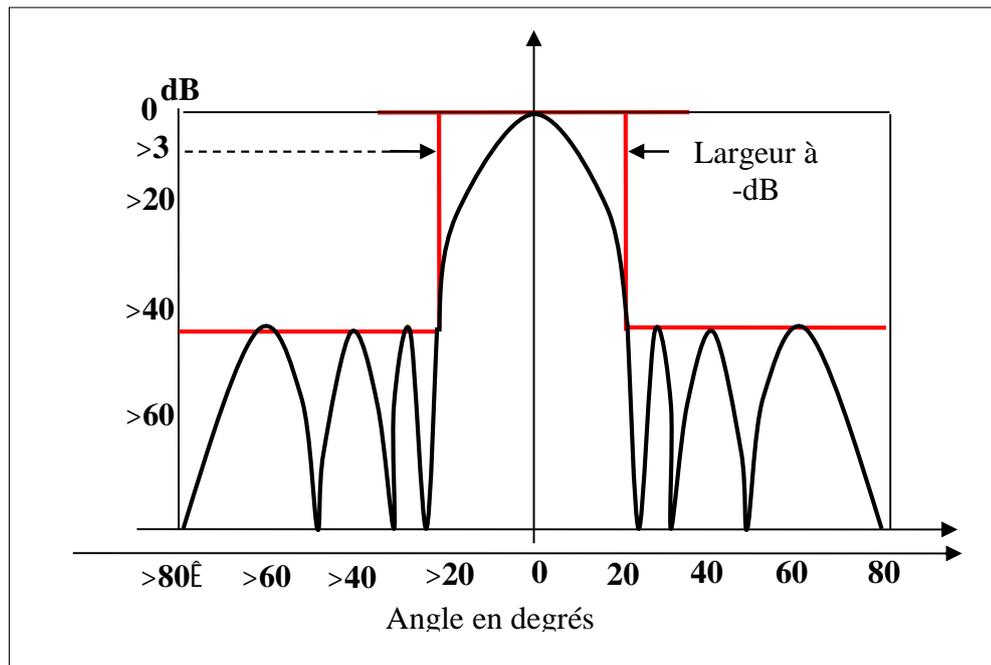


Fig. II.3 : Exemple de gabarit avec largeur de lobe principal fixée

II.2.3 Autre contraintes

D'autres considérations peuvent être envisagées lorsqu'il s'agit de faire la synthèse de diagrammes d'antennes, par exemple :

■ Temps de calcul et précision des résultats obtenus

La précision dépend du nombre d'itérations que l'on se fixe ainsi que de l'ordre d'approximation que l'on effectue à chaque itération. Le temps du calcul dépend du type de convergence de la méthode (linéaire, quadratique, etc...), de la formulation de la fonction objectif, et des valeurs initiales choisies

■ Prise en compte des contraintes de réalisation pratiques

1. Contraintes sur les paramètres de synthèse, par exemple des contraintes sur la dynamique du courant ou sur la position des éléments.
2. Possibilité d'introduire les pertes dans la formulation du problème (pertes diélectriques, métalliques, rayonnements).
3. Possibilité de tenir compte de couplage entre les éléments.

II.3 Paramètres pouvant être utilisés pour la synthèse

Il existe différents paramètres du réseau d'antennes sur lesquels on peut agir pour modifier le diagramme afin de respecter les spécifications du gabarit. On distingue principalement trois types :

■ Synthèse en amplitude seulement

Ce type permet de réaliser des lobes directifs symétriques avec la possibilité de jouer sur le niveau des lobes secondaires. Les techniques analytiques (Fourrier, Tchebycheff, ...) permettent de calculer les coefficients du réseau. Les applications de ce type de synthèse sont limitées.

■ Synthèse en amplitude et phase

Ce type permet de réaliser des lobes directifs avec des niveaux de lobes secondaires "fortement contrôlable". Cette technique est efficace pour les applications en réseau adaptatif, mais sa mise en pratique, nécessite un outil de synthèse en amplitude et en phase. Ce type de synthèse est lourd (temps de calcul important) et nécessite l'utilisation de circuits actifs de type, coûteux et compliqués.

■ Synthèse en phase

Ce type permet de réaliser des lobes directifs avec un niveau des lobes secondaires " moyennement contrôlable". Avec cette technique, on peut contrôler le niveau reçu dans la direction du rayonnement utile et interférent. Sa mise en pratique nécessite des "déphaseurs" contrôlés par un outil de synthèse de phase. Ce type de synthèse est un bon compromis qui permet de réaliser le formage du lobe directif avec un temps de calcul et un coût relativement faible.

II.3.1 Synoptique du logiciel de synthèse

Une synoptique de logiciel de synthèse de diagramme est représenté en figure II.4. Les étapes sont :

- Une partie initialisation, où sont définis l'étude et les objectifs désirés.
 - Une partie calcul.
 - Une partie traitement des résultats.
-

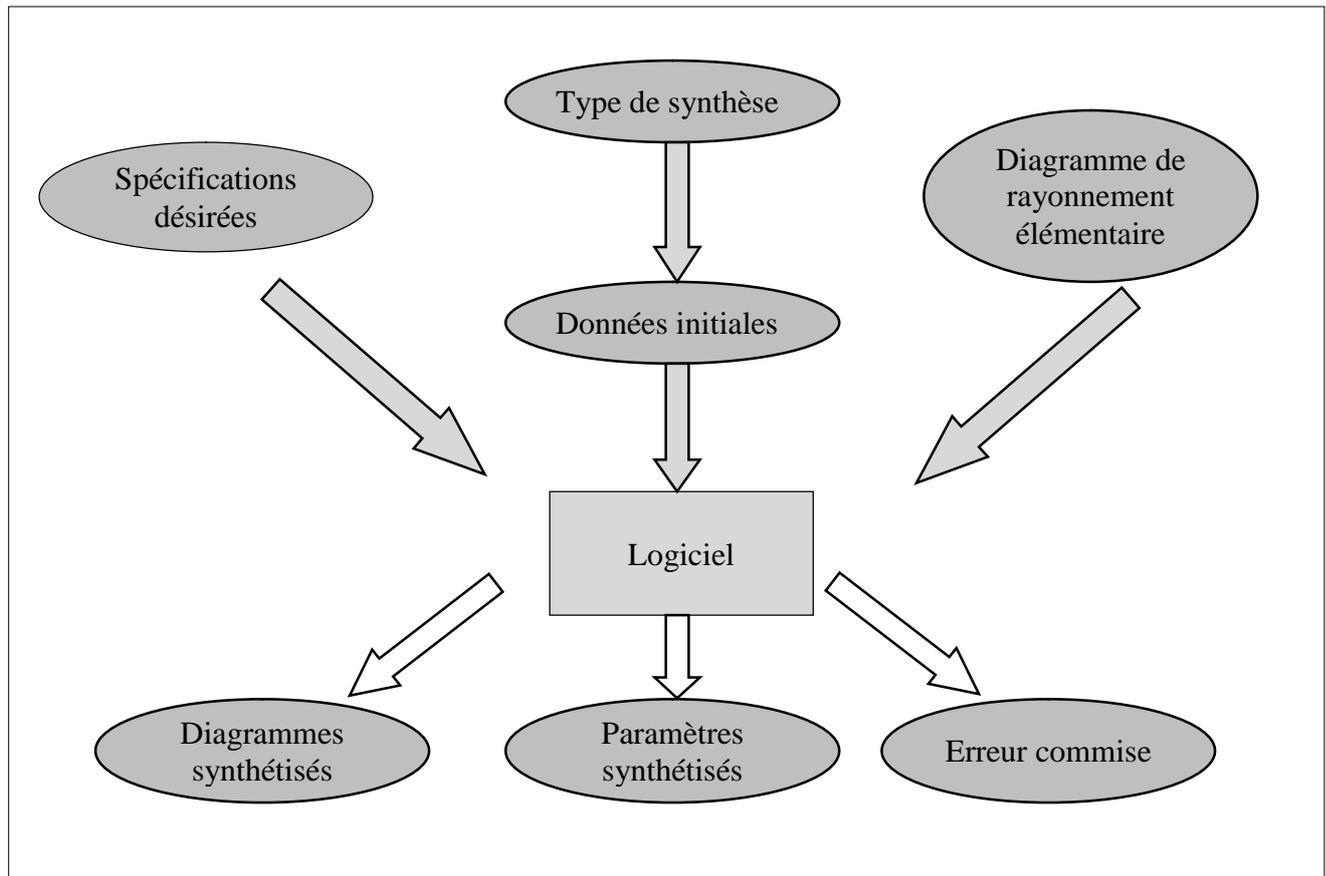


Fig. II.4 : Synoptique d'un logiciel de synthèse

■ Les spécifications

Ces spécifications sont définies à partir des fonctions poids qui contrôlent le niveau des lobes secondaires et de l'amplitude maximale des oscillations des lobes principaux formés, dans les différentes zones constituant le plan de formage (gabarit),

■ Le type de synthèse

La synthèse peut être effectuée en amplitude et (ou) en phase et dans certains cas en amplitude et en position d'éléments.

■ Les paramètres de synthèse initiaux

Une loi d'alimentation uniforme est généralement suffisante.

En sortie le logiciel de synthèse génère :

■ Les paramètres synthétisés

Ils varient en nature et en nombre suivant le type de synthèse spécifié. Les amplitudes et phases des excitations permettent de concevoir le répartiteur de puissance du réseau et d'écartier certaines tentatives de synthèse lorsque des éléments ne sont pas excités.

■ Le diagramme synthétisé

Il évalue qualitativement le diagramme de rayonnement du réseau, c'est souvent sur son unique aspect que l'on approuve ou rejette une tentative de synthèse.

■ Un compte rendu de l'erreur maximum commise

On en déduit le niveau des lobes secondaires et l'amplitude des oscillations dans le lobe principal. Cette erreur est également source d'acceptation ou de rejet de la tentative.

II.4 Méthodes classiques de synthèse de diagrammes de rayonnement

II.4.1 Les différentes méthodes

La systématisation des procédés de synthèse des groupements a entraîné le développement de nombreuses méthodes. On peut considérer deux types :

■ Méthode de Schelkunoff

Construit le diagramme en partant de la spécification de la direction des zéros dans le plan complexe z ou des directions de nuls dans l'espace réel.

■ Méthodes de formation de lobe (beam-shaping)

Ce sont des méthodes dans lesquelles on spécifie la forme du diagramme dans l'espace réel. On a l'habitude d'utiliser deux méthodes :

- Synthèse de Fourier : qui permet d'obtenir le diagramme avec une erreur quadratique moyenne par rapport aux spécifications. Cette méthode s'applique quand on connaît une expression mathématique du facteur de groupement désiré.
- Synthèse de Woodward : permet d'obtenir un diagramme qui coïncide avec les spécifications en un nombre fini de points. Elle est utile quand le diagramme spécifié ne peut s'exprimer mathématiquement, mais peut s'échantillonner.

Pour des diagrammes de rayonnement avec un lobe principal étroit et des lobes secondaires de niveau bas, on a l'habitude de spécifier les niveaux du lobe principal et des lobes secondaires et le nombre d'éléments du groupement. Pour réaliser cet objectif, il existe des méthodes variées :

- Synthèse de Tchebycheff: qui obtient le diagramme de rayonnement avec une ouverture minimale du lobe principal pour un niveau de lobe principal et secondaire donné.
- Synthèse de Taylor : elle est basée sur le même principe que la méthode de Tchebycheff, mais avec un rayonnement moindre dans les directions proches du lobe principal au prix d'un léger élargissement du lobe principal et d'une diminution de la directivité. Par ses avantages, cette méthode est très utilisée en pratique.

Groupements super-directifs il existe une série de méthodes pour synthétiser des groupements de directivité aussi élevée qu'on désire en théorie, au prix de problèmes considérables dans la réalisation pratique.

C'est possible, mais pas dans tous les cas, de calculer analytiquement les amplitudes et les phases des courants d'excitation des antennes du groupement pour obtenir le diagramme désiré . Mais ces méthodes, à cause de la complexité des calculs, ne sont utiles que dans les groupements statiques.

Nous allons décrire maintenant les deux méthodes de synthèse utilisées dans notre travail.

II.4.2 Méthode polynomiale de Schelkunoff

C'est une méthode analytique propice pour la synthèse de groupements dont les diagrammes de rayonnement possèdent des nuls dans des directions déterminées.

Le facteur de groupement de N éléments équidistants avec des amplitudes d'excitation non uniformes et des excitations de phase progressives est donné par :

$$F(\psi) = \sum_{n=1}^N a_n e^{j(n-1)\psi} \quad \text{où} \quad \psi = \phi + kd \sin \theta \quad (\text{II.1})$$

où a_n sont les amplitudes des excitations non uniformes de chaque élément du groupement.

En représentant :

$$z = x + jy = e^{j\psi} = e^{j(\phi + kd \sin \theta)} \quad (\text{II.2})$$

L'expression (II.1) peut se réécrire comme :

$$F(z) = \sum_{n=1}^N a_n z^{n-1} = a_1 + a_2 z + a_3 z^2 + \dots + a_N z^{N-1} \quad (\text{II.3})$$

L'expression (II.3) représente un polynôme de degré (N-1). Ce polynôme possède (N-1) racines et peut s'exprimer comme un produit de termes linéaires. Ainsi, l'expression (II.3) s'écrit comme :

$$F(z) = a_N (z - z_1)(z - z_2)(z - z_3) \dots (z - z_{N-1}) \quad (\text{II.4})$$

où z_1, z_2, \dots, z_{N-1} sont les racines du polynôme, lesquelles peuvent être complexes. Le module de l'expression (II.4) s'exprime comme :

$$|F(z)| = a_N |z - z_1| |z - z_2| |z - z_3| \dots |z - z_{N-1}| \quad (\text{II.5})$$

La variable complexe z peut s'écrire d'une autre forme comme :

$$z = |z| e^{j\psi} = z \angle \psi = 1 \angle \psi \quad (\text{II.6})$$

La méthode consiste donc à fixer les directions θ_i des nuls, ce qui permet de trouver ensuite les zéros z_i à l'aide de l'expression (II.2). Le développement de l'expression (II.4) permet d'obtenir ensuite les excitations a_i du groupement.

Dans la figure II.5, on voit l'effet de l'addition d'un zéro dans le polynôme d'un groupement.

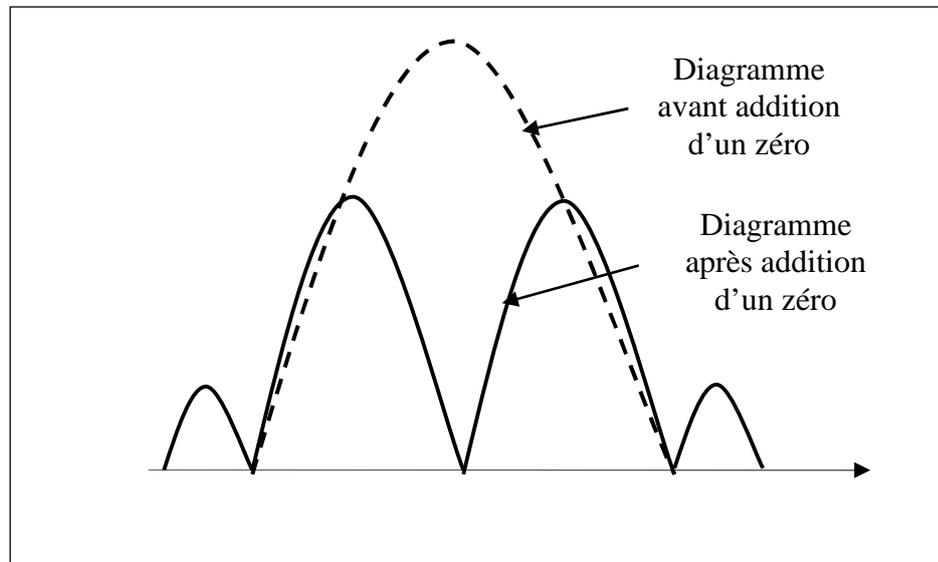


Fig. II.5 : Méthode à placement de zéros

II.4.3 Méthode de Dolph-Tchebycheff

La méthode de Dolph- Tchebycheff, très connue est une méthode de synthèse simple qui donne des résultats satisfaisants dans la plupart des cas. Nous utiliserons dans notre travail comme moyen d'apprentissage du réseau de neurones.

Grosso modo, cette méthode permet d'obtenir un lobe principal d'ouverture minimale, pour un niveau donné R des lobes secondaires par rapport au lobe principal. Les lobes secondaires ont tous ici le même niveau. La figure II.6 illustre un tel diagramme.

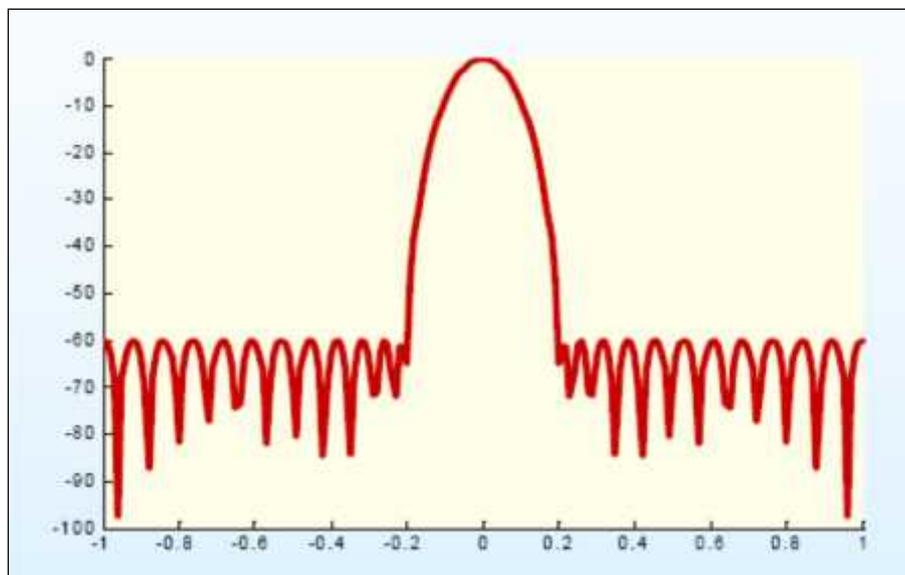


Fig. II.6 : Diagramme par la méthode Dolph (25 antennes et $R=60\text{dB}$)

L'ensemble des excitations des éléments d'un groupement d'antennes est relié aux polynômes de Tchebycheff, comme nous allons le voir.

La formule récursive des polynômes de Tchebycheff, est donnée par :

$$T_m(z) = 2 \cdot T_{m-1}(z) - T_{m-2}(z) \quad (\text{II.7})$$

Chaque polynôme peut se calculer en utilisant :

$$T_m(z) = \cos[m \operatorname{Arccos}(z)] \quad -1 \leq z \leq 1 \quad (\text{II.8a})$$

$$T_m(z) = \cosh[m \operatorname{Arccosh}(z)] \quad |z| \geq 1 \quad (\text{II.8b})$$

Le polynôme de Tchebycheff de degré 5 est représenté en figure II.7 à gauche.

On peut montrer qu'un diagramme de rayonnement d'un groupement linéaire d'antennes peut être décrit par un polynôme en $x = \cos \frac{\psi}{2}$ de degré égal au nombre de sources moins un, dont les coefficients dépendent de l'amplitude des sources. Il est alors possible d'obtenir une optimisation du niveau des lobes secondaires en calculant les amplitudes de manière à identifier le polynôme précédent avec le polynôme de Tchebycheff de degré $N-1$ en $z_0 x$, où :

x est la variable précédente.

z_0 : déterminé à partir du niveau $R = \frac{\text{Amplitude du lobe principal}}{\text{Amplitude du lobe secondaire}}$ des lobes secondaires par la

formule :

$$z_0 = \cosh \left[\frac{1}{N-1} \operatorname{Arccosh}(R) \right] \quad (\text{II.9})$$

La figure II.7 illustre ce principe connu sous le nom de méthode de Dolph.

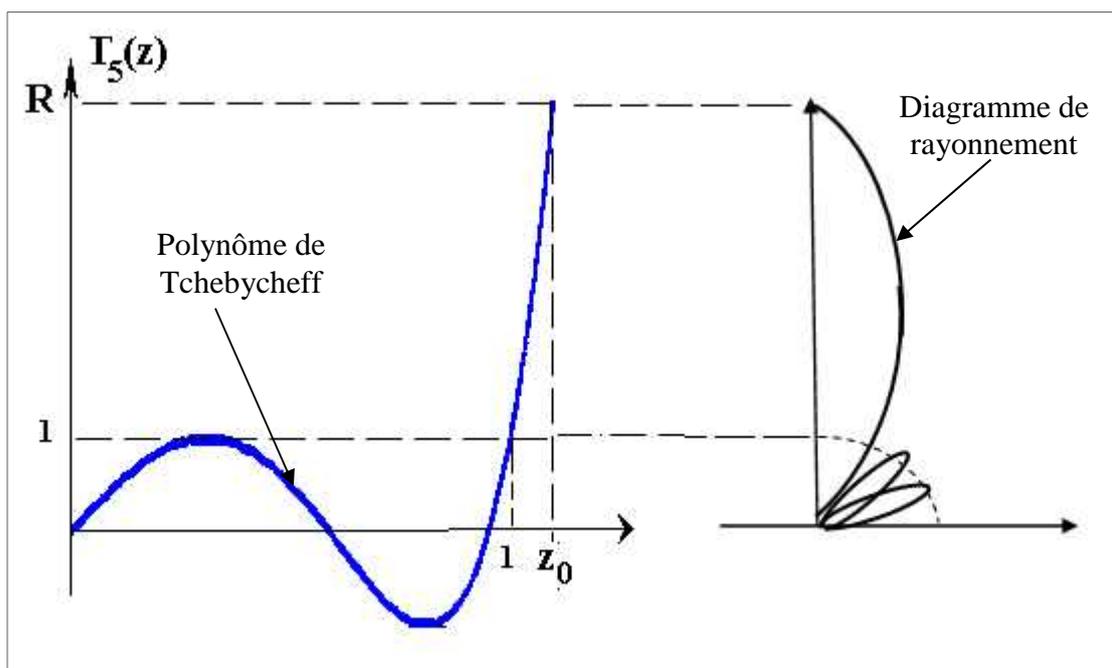


Fig. II.7 : relation entre polynômes de Tchebycheff et diagramme de rayonnement

II.5 Méthodes d'optimisation globale

II.5.1 Les différents types

Beaucoup de problèmes de synthèse en électromagnétisme tendent à être de type multimodales, multidimensionnelles et multivariables. D'un autre côté, ils consistent généralement en différents objectifs à la fois et ils se convertissent ainsi en problèmes d'optimisation multiobjectif, par exemple le fait d'imposer des nuls dans certaines directions et en même temps avoir un lobe principal étroit.

la plupart des techniques d'optimisation classiques d'approximation analytiques ne proposent pas des solutions flexibles pour les problèmes d'optimisation multiobjectif antérieurs. Les inconvénients des techniques classiques et analytiques et le développement rapide des techniques de calcul dans les années récentes ont impulsé l'usage d'algorithmes d'optimisation évolutifs basés sur des méthodologies d'intelligence artificielle.

L'optimisation globale se réfère à la recherche d'une solution x d'un ensemble de solutions possibles S . Optimisation est un terme qui a différentes connotations. Généralement, il s'utilise pour se référer à un processus qui permet à un système d'obtenir la meilleure sortie ou solution possible. L'objectif, lorsqu'on résout un problème d'optimisation, est de trouver une solution optimale avec un coût de calcul raisonnable. On peut classer les méthodes d'optimisation en deux catégories :

■ Méthodes de recherche locale

Elles obtiennent la meilleure solution au voisinage du point initial, donnant ainsi une forte dépendance avec le point de départ. La plupart de ces méthodes reposent sur la variation du gradient, ce qui implique différenciabilité, continuité, conditions difficiles à rencontrer dans la pratique. La majorité des techniques analytiques utilisent une méthode déterministe point par point comme c'est illustré dans la figure II.8 a

■ Méthodes de recherche globale

Elles sont capables d'explorer tout l'espace de recherche de manière exhaustive et trouver une solution globale, comme c'est illustré dans la figure II.8 b.

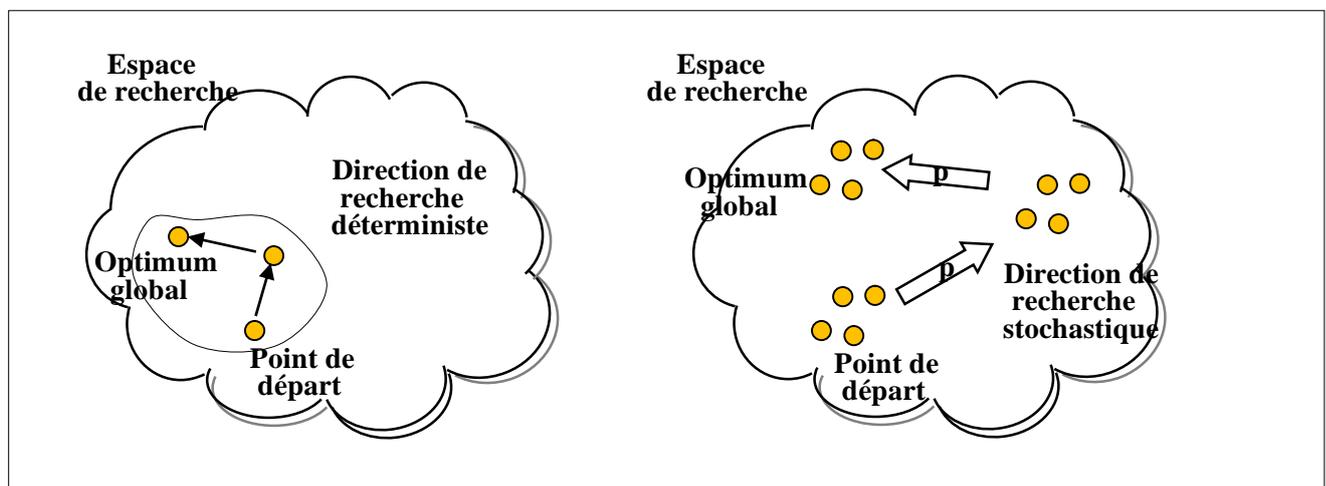


Fig II. 8: (a) Recherche locale (b) Recherche globale où p est la probabilité de transition

Une classification plus spécifique des méthodes d'optimisation est montrée de manière concrète en figure II. 9.

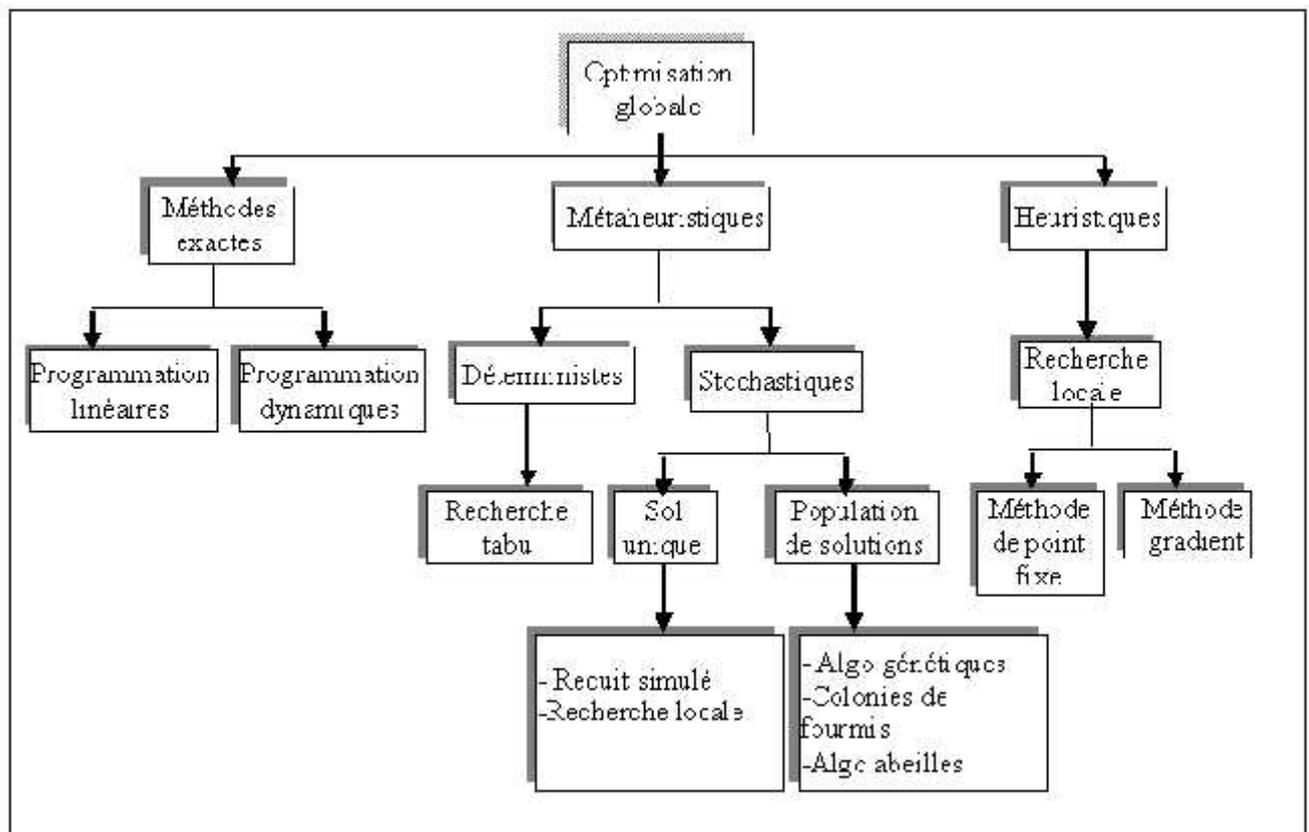


Fig II.9 : Classification des méthodes d'optimisation

II.5.2 Algorithmes g

Comme nous allons utiliser l'algorithme génétique afin de générer des données d'apprentissage pour les réseaux de neurones.

■ Le principe

Cette méthode d'optimisation fut utilisée par Panduro Mendoza pour la synthèse d'un groupement d'antennes avec l'objectif d'optimiser le lobe principal, contrôlant les poids complexes des éléments du groupement en se basant sur le principe de la sélection naturelle de Darwin.

Dans le schéma classique, on fait évoluer la population constituée d'un ensemble d'individus (chromosomes) grâce à une suite d'opérations de sélection, de croisement et de mutation.

A partir d'une population initiale et après un certain nombre d'itérations (générations), on obtient une population d'individus forts, c'est-à-dire de bonnes solutions se rapprochant de la solution optimale du problème considéré

D'une manière générale, les algorithmes génétiques requièrent :

- une représentation des solutions potentielles du problème (codification),
- d'initialiser aléatoirement une population,
- de classer les solutions en fonction de leur aptitude à travers une fonction d'évaluation (fitness),

- un choix des opérateurs (sélection, croisement, mutation) qui modifient la structure des solutions.

■ Structure de l'algorithme

L'algorithme génétique proprement dit commence par générer une population de solutions de forme aléatoire en utilisant un codage approprié et on entre dans la boucle (figure II.10).

- 1) On évalue la fonction d'évaluation (fitness) de chacun des chromosomes. Si on dépasse un certain nombre de générations, on sort de la boucle,
- 2) Sélection : on sélectionne les meilleures individus de la population en fonction de leur fitness pour former un ensemble d'individus forts,
- 3) Croisement : reproduction par croisement pour compléter la population,
- 4) Mutation : modification aléatoire d'un gène d'un individu,
- 5) Nouvelle génération : la nouvelle population se substitue à l'ancienne,
- 6) On retourne au point 1.

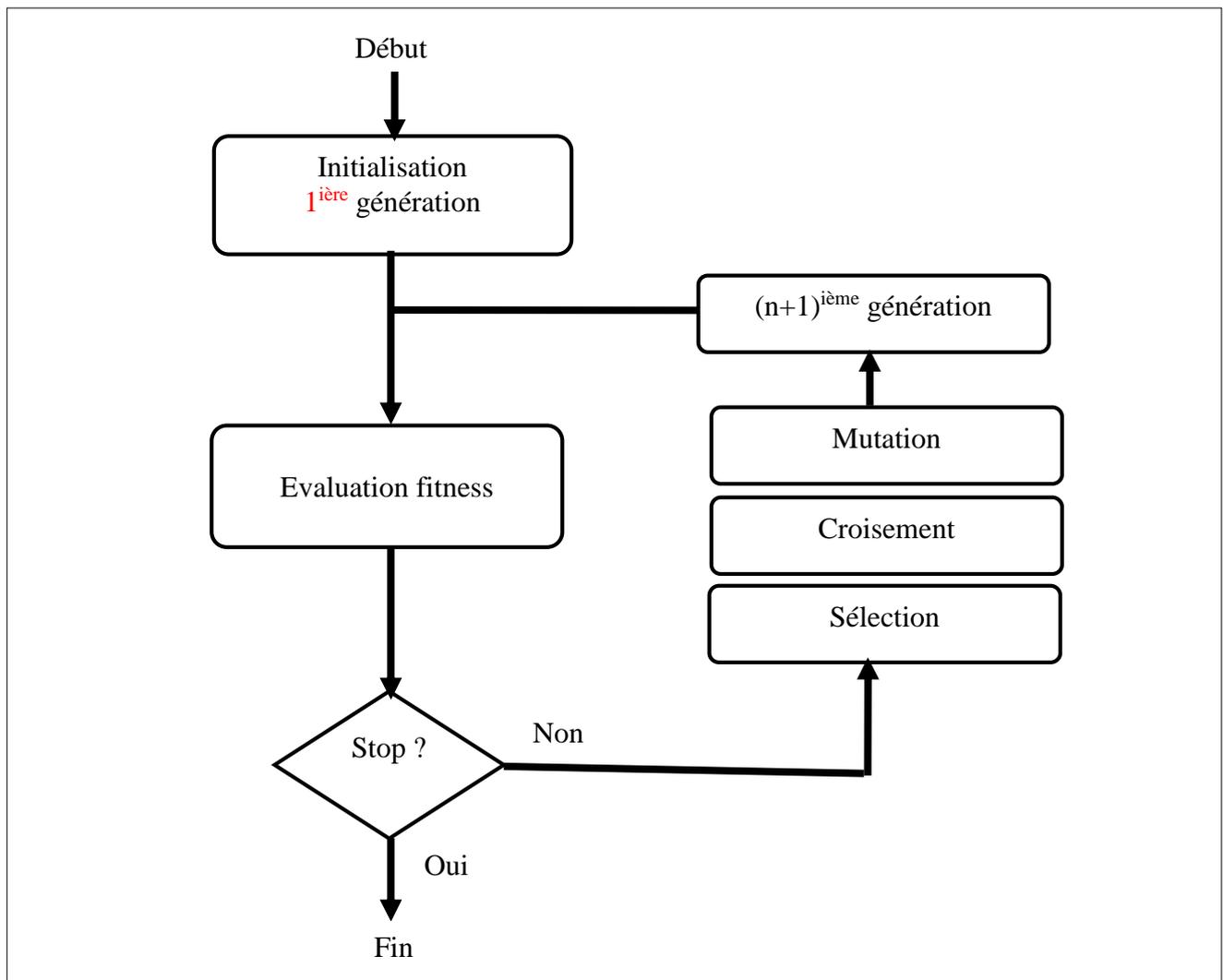


Fig II.10 : Structure générale des algorithmes génétiques

■ Principe de synthèse de réseau au placement de nuls par l'algorithme génétique

Dans notre travail, nous avons repris un travail antérieur de synthèse de diagramme par placement de nuls (méthode de Schelkunoff) avec l'algorithme génétique pour utiliser les résultats comme données d'apprentissage des réseaux de neurones.

Nous rappelons ici brièvement le principe de l'adaptation de l'algorithme génétique à la synthèse de réseau par placement de nuls.

Le lien entre l'algorithme génétique et le problème d'adaptation du réseau d'antenne est réalisé par la fonction fitness suivante :

Soit un réseau rectiligne et uniforme de N éléments. Le champ total est donné par :

$$F(\theta) = f(\theta) \sum_{i=1}^{N-1} a_i \exp(jk_0 i d \sin \theta \cos \varphi + b_i) \quad (\text{II.10})$$

et θ : angle de direction,

a_i : amplitude de l'alimentation du réseau d'antenne,

b_i : phase de l'alimentation de réseau d'antenne,

d : représente l'espacement entre les éléments du réseau,

$f(\theta)$: champ d'un élément du réseau d'antenne (il est identique pour l'ensemble des antennes du réseau),

k_0 : le nombre d'onde.

Le principe de l'adaptation du réseau d'antennes est de déterminer la pondération complexe d'alimentation qui place des zéros dans les directions des interférences, c'est-à-dire trouver les valeurs de a_i et b_i . L'approche utilisée à l'aide de l'algorithme génétique pour adapter le réseau d'antennes à son environnement est de fixer l'amplitude de la pondération et de rechercher uniquement la loi de phase optimale qui minimise les niveaux de rayonnement en directions des interférences.

Le lien entre l'algorithme génétique et le problème d'adaptation du réseau d'antennes est réalisé par la fonction fitness suivante :

$$\text{fitness} = 20 \log_{10} \left(\sum_{i=1}^{M+1} s_i f(\theta_i) \sum_{n=1}^N a_n \exp(jk_0 n d \sin \theta_n \cos \varphi_n + b_n) \right) \quad (\text{II.11})$$

M : le nombre d'interférence,

s_i : vecteur d'espace des sources.

Les variables à optimiser sont représentées par des gènes et l'ensemble des gènes constitue un individu. Par analogie avec notre problème, les gènes sont les b_n et l'individu est l'ensemble du vecteur phase des N éléments. Nous utiliserons un codage binaire de 16 bits afin d'augmenter la précision de calcul de la loi d'alimentation.

La première étape de l'algorithme génétique est de générer une population initiale sous forme d'une matrice binaire de L ligne et de C colonnes, telles que : L est le nombre d'individus et C le nombre de gènes dans l'individu et est égale au problème, c'est-à-dire N fois le nombre de bits du codage binaire utilisé. Nous évaluerons la force des individus de la population, en calculant la fitness de chaque individu (chaque ligne de la matrice initiale). Pour cela, on décode le chromosome correspondant à chaque individu. On utilisera la formule de décodage suivante :

$$b = \frac{p_{\max} - p_{\min}}{2^N} \sum_{i=0}^{N-1} 2^i p_i + p_{\min} \quad (\text{II.12})$$

p_{\min} et p_{\max} sont les bornes inférieures et supérieures de l'intervalle de variation des phases. Le vecteur b obtenu servira ensuite à calculer la fonction fitness de cet individu. A partir de cette étape, les opérations de l'algorithme génétique vont intervenir dans la reproduction par : la sélection, le

croisement et la mutation. L'AG effectue la sélection et le classement des meilleurs individus en se basant sur le principe de la sélection proportionnelle. Après le classement de la population, on génère une nouvelle population dite population enfants.

On répète les quatre opérations tant qu'un nombre de générations déterminé à l'avance n'est pas atteint.

II.5.3 La méthode du recuit simulé

Le recuit simulé est apparu en 1982, avec la publication dans la revue science d'un article de Kirpatrick ; Le recuit simulé fut obtenu par analogie avec le phénomène thermodynamique de recuit des métaux. Initialement, le métal est porté à très haute température, puis il est refroidi progressivement. La méthode du recuit simulé (RS) à l'originalité de pouvoir s'appliquer à une grande variété de domaines et en particulier aux télécommunications.

Le recuit simulé est une heuristique d'optimisation qui consiste en une recherche locale par perturbations. Ce processus donne la possibilité de s'éloigner occasionnellement d'un minimum local pour permettre ainsi un élargissement du champ de recherche de la solution idéale. Le recuit simulé est une procédure de recherche selon laquelle la topologie courante, retenue momentanément comme meilleure solution, est continuellement comparée à d'autres topologies qui lui sont très proches. Ces topologies voisines sont obtenues à la suite de petites perturbations sur la topologie courante. Lorsqu'une perturbation aboutit à une topologie meilleure que la solution courante, elle est sauvegardée comme solution courante. Cependant, il peut arriver que, suite à une perturbation, la topologie voisine obtenue soit conservée comme solution courante, même si elle n'est pas meilleure que la solution courante, à condition qu'elle respecte une certaine probabilité d'acceptation. Le fait d'accepter de temps à autre une solution dégradée permet d'éviter de s'enfermer trop tôt dans un minimum local.

II.6 Réseaux de neurones

Depuis une cinquantaine d'années, les chercheurs en informatique s'inspirent de la biologie dans le but de réaliser des constructions capables de résoudre divers problèmes complexes. Les réseaux de neurones s'inspirent fortement du fonctionnement du cerveau humain.

- les neurones sont des composants non-linéaires, par conséquent le réseau aussi : ceci est une propriété très importante car cela leur permet de traiter des problèmes physiques non linéaires
- ils sont capables d'apprendre et de généraliser : après une période d'apprentissage qui consiste à leur présenter des entrées ainsi que les sorties que l'on voudrait qu'ils trouvent, ils sont capables de produire des sorties "raisonnables" sur des cas qu'ils n'ont pas appris.

D'où l'idée de les utiliser pour synthétiser les diagrammes de rayonnement. C'est l'objet de notre travail.

II.7 Conclusion

Nous avons présenté dans la première partie de ce chapitre la notion de gabarit, puis nous avons décrit quelques méthodes de synthèse de diagramme, en insistant sur la méthode de Dolph-Chebyshev. Nous avons rappelé aussi le procédé de mise en œuvre de l'algorithme génétique (AG) dans la synthèse des réseaux d'antennes. Ces deux méthodes seront au cœur de notre simulation des réseaux de neurones.

CHAPITRE III : LES RESEAUX DE NEURONES

III.1 Introduction

Nous décrivons en détails dans ce chapitre les concepts fondamentaux et les notions liées aux réseaux neuronaux (neurone biologique, modèle mathématique, type de réseaux, méthodes et algorithme d'apprentissage...).

III.2 Réseaux de Neurones : Théories fondamentales

Depuis la création de la terre, l'homme n'a pas cessé des questions, sur le monde qu'il entoure, par exemple comment on peut faire la différence entre deux choses, comment on peut parler, calculer...et quelle est la partie qui fait tout ce travail ?la réponse pour toutes ces questions :c'est le cerveau, ce dernier est un dispositif très complexe, il peut réaliser des millions d'opération en une seconde, mais la question qui se pose :pourrait-il exister un jour des machines ou des circuits électriques qui peuvent remplacer notre cerveau?

Depuis quelques années, des chercheurs de part le monde entier relèvent ce défi, or en 1943 les deux américains de Chicago, le neurophysiologue Warren Mcculloch et logicien Walter Pitts en inventent le premier neurone formel qui portera leurs noms (neurone de Mcculloch-Pitts) et sert à calculer certain fonctions logiques.

Quelques années plus tard en 1949, Hebb souligna l'importance du couplage synoptique dans les processus d'apprentissages, cette dernière repose sur le fait que si deux neurones sont connectés entre eux et activés en même temps, les connexions qui les relient doivent êtres renforcées, dans le cas contraire, elles ne sont pas modifiées.

En 1958 Frank Rosenbblat le premier réseau de neurone artificiel qui conçoit le fameux 'perceptron',ce modèle à réussi à calculer certaines fonctions logiques, et l'utiliser aussi dans le domaine de la reconnaissance des formes simples, et cela jusqu'à la fin des années 60 car les deux mathématiciens Marvin Minsky et Seymour Papert ont prouvé les limitations du système, ils ont publié un livre dans lequel ils décrivent une série de problème de reconnaissance des formes que le perceptron ne peut pas résoudre et cette dernière est la cause pour stopper toutes les recherches sur la technologie des réseaux.

C'est l'évolution des systèmes informatiques et leur disponibilité qui permettent plus facilement la simulation des réseaux de neurones au début des années 80. De plus, des nouvelles découvertes dans

ce domaine contribuent au retour de cette approche comme par exemple, le modèle de Hopfield (1982) ou les recherches du groupe Rumelhart, Hinton et Williams (1986) qui sont arrivés à découvrir l'algorithme de la rétropropagation et à démontrer que ces règles d'apprentissage donnent la possibilité aux réseaux de neurones d'apprendre des fonctions tel le XOR, qui était impossible pour le perceptron.

Un ordinateur conventionnel possède une seule unité de traitement par laquelle transitent toutes les données. Sa structure est donc très différente de celle du cerveau. Un réseau neuronal présente, pour sa part, plus d'analogies avec le cerveau en ce qu'il comporte un grand nombre d'unités de traitement simples qui traitent chacune une petite partie des données mais de manière parallèle. Les ordinateurs conventionnels se sont révélés extrêmement efficaces pour effectuer des calculs à grande vitesse. Mais, pour des tâches difficiles à accomplir, telles que la reconnaissance des formes, en particulier, la reconnaissance de l'écriture manuscrite, la reconnaissance d'un visage particulier, ou l'expression de ce visage, il devient plus difficile de rédiger un ensemble d'instructions permettant de l'identifier.

Les capacités d'apprentissage du cerveau résultent essentiellement des fonctionnements collectifs et simultanés des neurones qui le composent, organisés en réseaux fortement interconnectés. La création de systèmes de neurones exige la connaissance approfondie de plusieurs domaines scientifiques. Des spécialisations au niveau médical, principalement en neurophysiologie sont requises pour l'apport d'informations sur le fonctionnement du cerveau humain. Des notions de mathématiques ainsi que de la physique sont impératives pour la création de formules permettant la modélisation du cerveau.

III.3 Le neurone biologique

III.3.1 Description

La physiologie du cerveau montre que celui-ci est constitué de cellules (les neurones) interconnectées, ce qui est schématisé dans la figure III.1.

Les neurones reçoivent les signaux (impulsions électriques) par des extensions très ramifiées de leur corps cellulaire (les dendrites) et envoient l'information par de longs prolongements (les axones). Les impulsions électriques sont régénérées pendant le parcours le long de l'axone. La durée de chaque impulsion est de l'ordre d'une milliseconde et son amplitude d'environ 100 mvolts.

Les contacts entre deux neurones, de l'axone à une dendrite, se font par l'intermédiaire des synapses. Lorsqu'un potentiel d'action atteint la terminaison d'un axone, des neuromédiateurs sont libérés et se lient à des récepteurs post-synaptiques présents sur les dendrites. L'effet peut être excitateur ou inhibiteur.

Chaque neurone intègre en permanence jusqu'à un millier de signaux synaptiques. Ces signaux n'opèrent pas de manière linéaire (effet de seuil).

On peut décomposer le neurone en trois régions principales : le corps cellulaire, l'axone et l'arborisation dendritique.

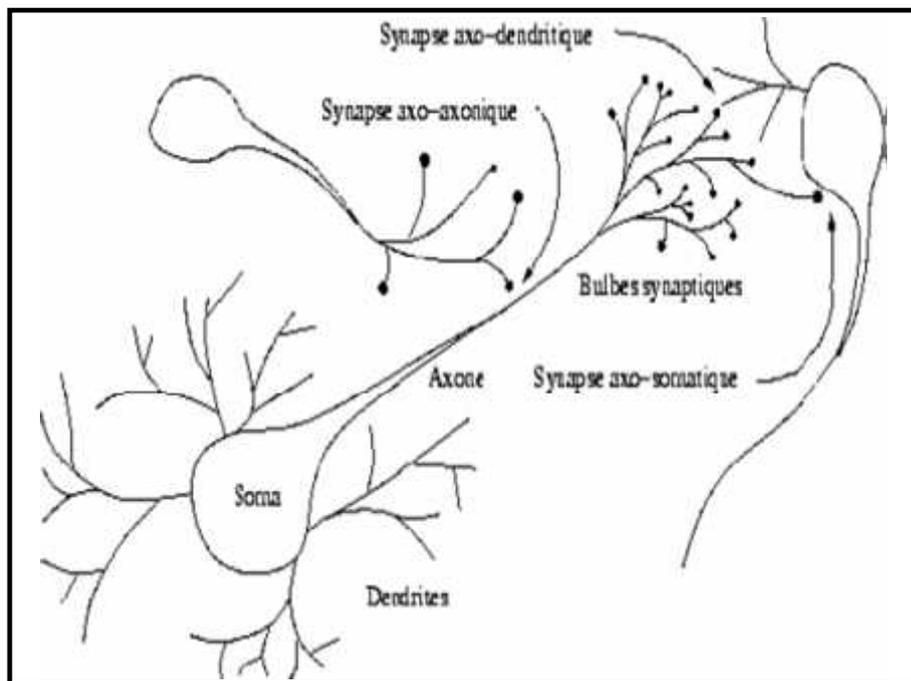


Fig. III.1 : Les neurones schématisés

En ce qui concerne les neurones, on a les informations suivantes :

- le cerveau contient environ 100 milliards de neurones.
- on ne dénombre que quelques dizaines de catégories distinctes de neurones.
- aucune catégorie de neurones n'est propre à l'homme.
- la vitesse de propagation des influx nerveux est de l'ordre de 100m/s. C'est à dire bien inférieure à la vitesse de transmission de l'information dans un circuit électronique.
- on compte de quelques centaines à plusieurs dizaines de milliers de contacts synaptiques par neurone. Le nombre total de connexions est estimé à environ 10^{15} .
- La structure du cerveau provient en partie des contacts avec l'environnement, l'apprentissage est donc indispensable à son développement.
- le nombre de neurones décroît après la naissance. Cependant, cette affirmation semble remise en question.

III.3.2 Fonctionnement

La réponse du neurone biologique est en général non linéaire, mais les neurophysiologistes ont trouvé que pour quelques neurones biologiques, le signal de sortie est proportionnel à la combinaison linéaire de tous les signaux d'entrées fournis par les autres neurones du système nerveux.

Il faut considérer (pour simplifier) le neurone comme une entité polarisée, c'est-à-dire que l'information ne se transmet que dans un seul sens : des dendrites vers l'axone. Pour rentrer un peu dans le détail, le neurone va donc recevoir des informations, venant d'autres neurones, grâce à ses dendrites. Il va ensuite y avoir sommation, au niveau du corps cellulaire, de toutes ces informations et via un potentiel d'action (un signal électrique) le résultat de l'analyse va transiter le long de l'axone jusqu'aux terminaisons synaptiques. A cet endroit, lors de l'arrivée du signal, des vésicules synaptiques vont venir fusionner avec la membrane cellulaire, ce qui va permettre la libération des neurotransmetteurs (médiateurs chimiques) dans la fente synaptique. Le signal électrique ne pouvant pas passer la synapse (dans le cas d'une synapse chimique), les neurotransmetteurs permettent donc le passage des informations, d'un neurone à un autre.

Au niveau post-synaptique, sur la membrane dendritique, se trouvent des récepteurs pour les neurotransmetteurs. Suivant le type de neurotransmetteur et le type des récepteurs, l'excitabilité du neurone suivant va augmenter ou diminuer, ce qui fera se propager ou non l'information.

Les synapses possèdent une sorte de "mémoire" qui leur permet d'ajuster leur fonctionnement. En fonction de leur "histoire", c'est-à-dire de leur activation répétée ou non entre deux neurones, les connexions synaptiques vont donc se modifier.

On pourra résumer cette modélisation par le tableau III.1, qui permet de voir clairement la transition entre le neurone biologique et le neurone formel.

Neurone biologique	Neurone artificiel
Synapses	Poids de connexions
Axones	Signal de sortie
Dendrite	Signal d'entrée
Cellule	Fonction d'activation

Tableau III.1 : Analogie entre neurones biologique et artificiel

III.4 Le neurone formel (Artificiel)

III.4.1 Le modèle mathématique

Le modèle de neurone formel présenté ici, du à Mac Culloch et Pitts, est un modèle mathématique très simple dérivé d'une analyse (elle aussi assez simple) de la réalité biologique. Il est considéré comme un automate relié à des automates voisins, par des connexions représentées par des poids réels, qui reçoit à son entrée des signaux délivrés par les neurones auxquels il est connecté, et il fournit à la sortie un signal qui dépend à la fois de ses signaux d'entrée et des poids de ses différentes connexions (somme pondéré A_i), (Figure III.2).

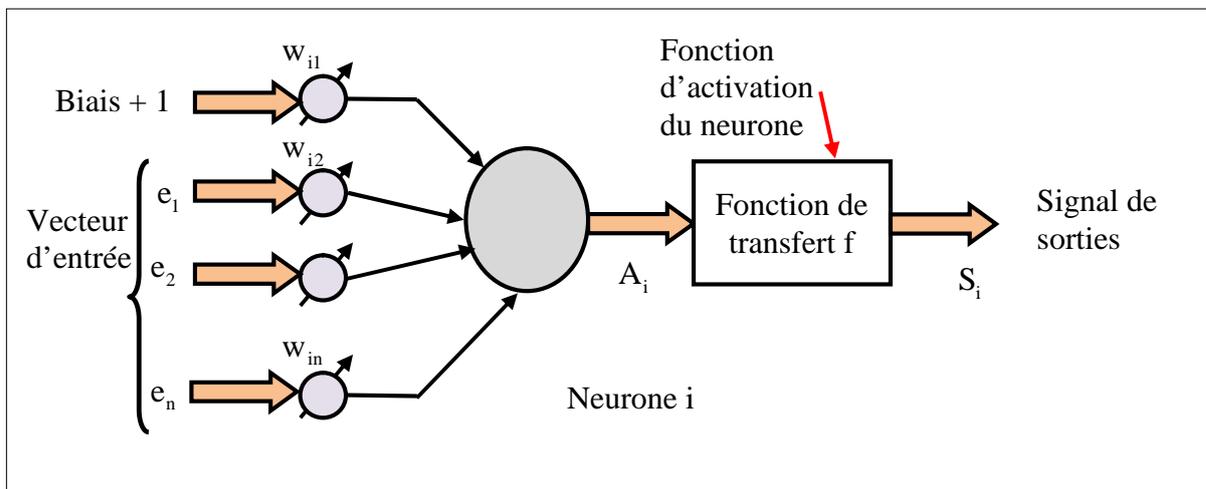


Fig. III.2: Schéma fonctionnel du neurone formel

On peut donc, d'après la figure III.2, définir un neurone formel par les paramètres suivants :

1. Un ensemble des connexions (ou synapses) dont chacune se caractérise par un poids réel. Le signal e_j se trouvant à l'entrée de la synapse j qui est connectée au neurone i . Ce signal est multiplié par le poids de la synapse w_{ij} . w_{ij} est donc le poids de la connexion dirigée du neurone j vers le neurone i . Si le poids est positif on aura donc un effet excitateur, s'il est négatif alors l'effet est inhibiteur.

2. La somme pondérée A_i des signaux d'entrée e_j du neurone i qui sont en même temps les signaux de sortie des neurones de la couche amont auquel ce neurone est connecté. La somme A_i est donnée par l'expression suivante :

$$A_i = \sum_{j=1}^n w_{ij} e_j + \theta_i \quad (\text{III.1})$$

3. Le seuil θ_i propre au neurone i qui est un nombre réel et qui représente la limite à partir de laquelle le neurone s'activera.

4. La fonction de transfert f qui limite en général l'amplitude de la sortie $S_i = f(A_i)$ du neurone entre $[-1, 1]$. Elle existe sous différentes formes, telles que : fonction non linéaire, fonction à seuil binaire, fonction linéaire à seuil, etc.

III.4.2 Fonction de transfert

En général, la fonction de transfert f d'un neurone représente l'état d'activation de ce neurone. Elle peut prendre les formes principales suivantes :

1. Fonction à seuil binaire (hard limiter) ou fonction signe (Signum function) (Figure III.3).

$$f(x) = \begin{cases} +1 & \text{si } x > 0 \\ -1 & \text{si } x < 0 \end{cases}$$

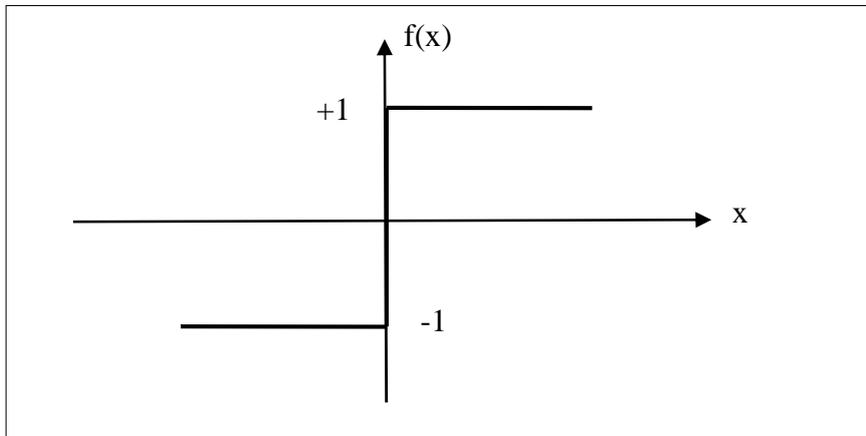


Fig. III.3: Fonction signe

2. Fonction sigmoïde (sigmoid limiter), (Figure III.4).

$$f(x) = \tanh\left(\frac{\alpha x}{2}\right) = \frac{1 - e^{-\alpha x}}{1 + e^{-\alpha x}} \quad (\alpha > 0)$$

où α est le gain de la fonction sigmoïde et il caractérise la forme sigmoïde de la fonction.

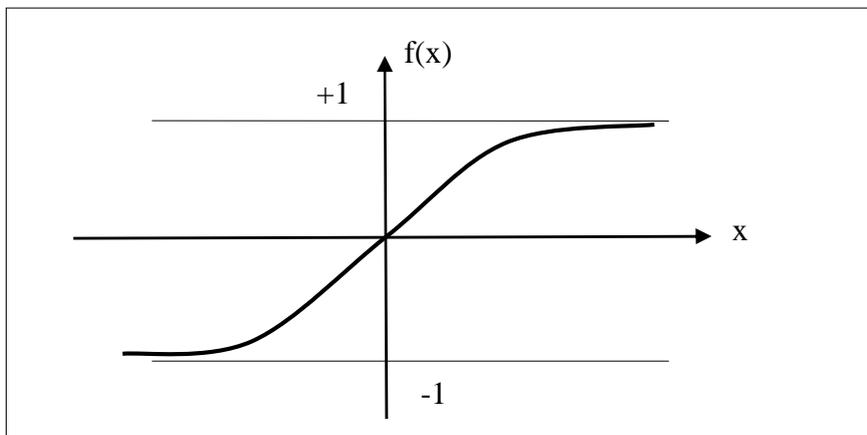


Fig. III.4: Fonction sigmoïde

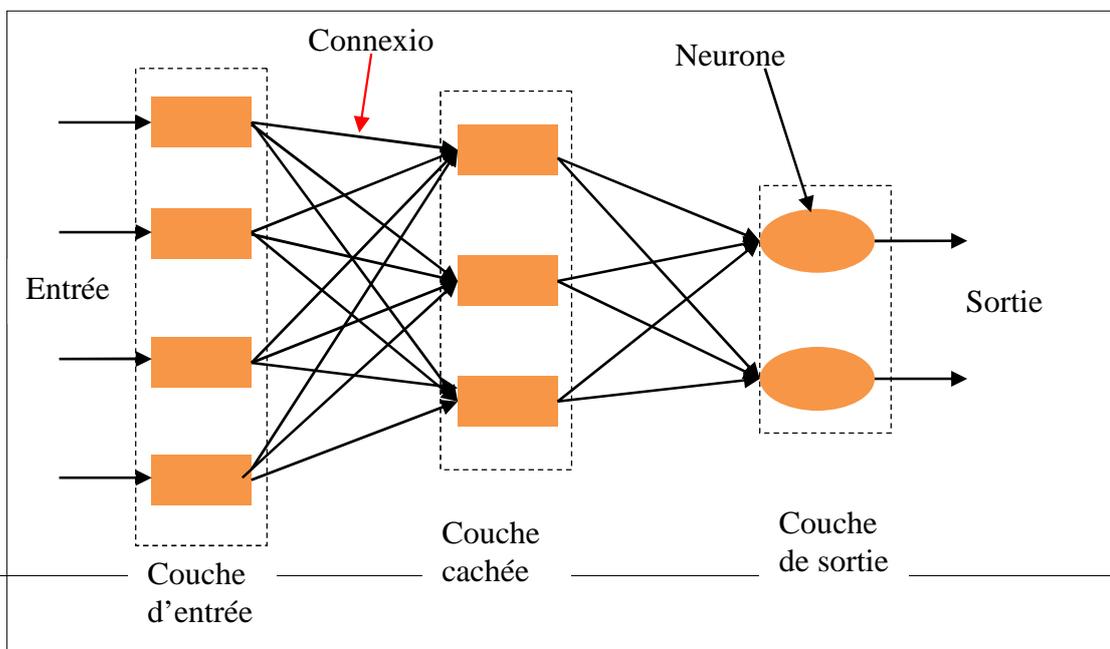
III.5 Réseaux de neurones artificiels

Un réseau de neurones artificiels est un ensemble de neurones formels associés en couches (ou sous-groupes) et fonctionnant en parallèle. Les réseaux de neurones artificiels ont la capacité de stocker de la connaissance empirique et de la rendre disponible à l'usage. Les habiletés de traitement (et donc la connaissance) du réseau vont être stockées dans les poids synaptiques, obtenus par des processus d'adaptation ou d'apprentissage. En ce sens, les réseaux de neurones artificiels ressemblent donc au cerveau car, non seulement, la connaissance est acquise à travers un apprentissage mais de plus, cette connaissance est stockée dans les connexions entre les entités, soit dans les poids synaptiques. La structure des connexions entre les différents neurones détermine la topologie du réseau. On peut classer les réseaux de neurones artificiels dans trois grandes catégories :

1. Réseau neuronal multicouche de type Feed- Forward (réseau non bouclé).
2. Réseau récurrent (réseau bouclé)
3. Réseau cellulaire.

III.5.1 Réseaux neuronaux multicouches de type Feed- Forward

Appelés aussi "réseaux de type Perceptron", ce sont des réseaux dans lesquels l'information se propage couche en couche sans que le retour en arrière soit possible. On trouve le perceptron monocouche et le perceptron multicouche. Le premier est un réseau simple, puisqu'il ne se compose que d'une couche d'entrée et d'une couche de sortie, sa principale limite est qu'il ne peut résoudre que des problèmes linéairement séparables. Le perceptron multicouche MLP (Multilayer perceptron) (Figure III.5) est une extension du précédent, avec une ou plusieurs couches cachées entre l'entrée et la sortie. Chaque neurone dans une couche est connecté à tous les neurones de la couche précédente et de la couche suivante (à l'exception des couches d'entrée et de sortie) et il n'y a pas de connexions entre les neurones d'une même couche. Les fonctions d'activation utilisées dans ce type de réseaux sont principalement les fonctions à seuil ou sigmoïdes. Il peut résoudre des problèmes non linéairement séparables et des problèmes logiques plus compliqués.



III.5.2 Réseaux récurrents

Appelés aussi réseaux bouclés ou dynamiques (Figure III.6), ce sont des réseaux dans lesquels il y a retour en arrière de l'information. Des connexions apparaissent entre la sortie du réseau et les neurones qui se trouvent dans des couches amonts. Ces réseaux sont souvent utilisés pour identifier des systèmes dynamiques non linéaires.

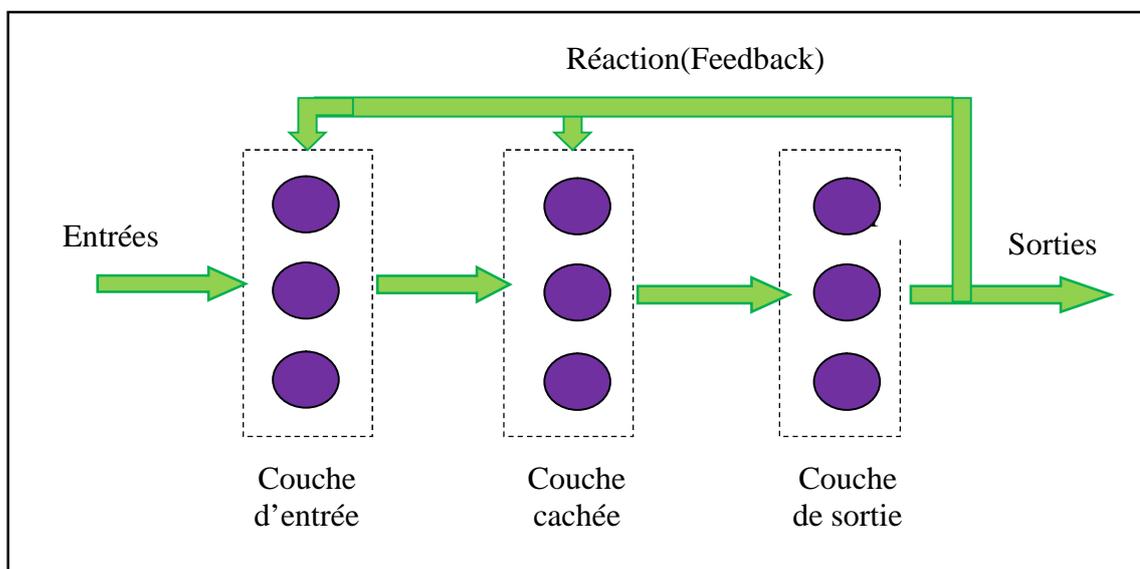


Fig. III.6: Structure d'un réseau neuronal récurrent

III.5.3 Réseaux cellulaires

Dans un réseau cellulaire (Figure III.7), les neurones sont entièrement connectés dans un plan où chaque neurone est relié à tous les neurones qui les entourent. Une connexion entre deux neurones peut être bidirectionnelle.

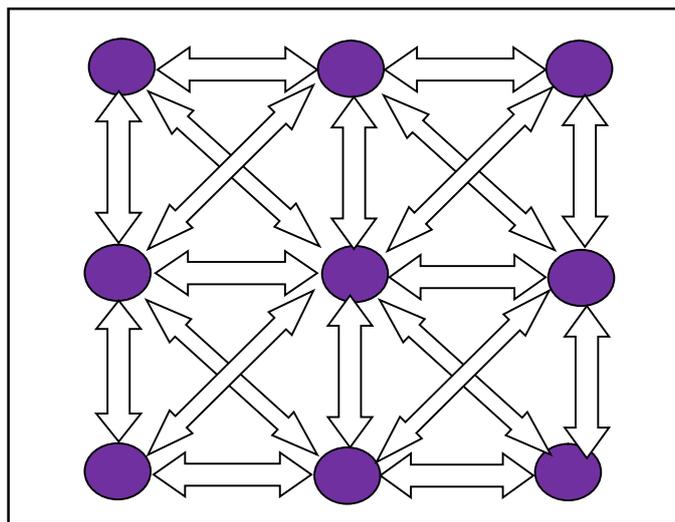


Fig. III.7: Réseau cellulaire

III.6 Les méthodes d'apprentissages des réseaux de neurones

Parmi les propriétés les plus intéressantes d'un réseau neuronal, la capacité d'un réseau à apprendre de son environnement et d'améliorer sa performance par un phénomène appelé phénomène d'apprentissage. Cette amélioration prend place avec le temps où les paramètres du réseau subissent des modifications suivant des règles appelées règles d'apprentissage, jusqu'à ce que la sortie désirée soit à peu près obtenue. Mais afin de pouvoir générer l'apprentissage du réseau, il est nécessaire de créer une base de données dans un domaine bien défini appelée la base d'apprentissage et qui contiendra toutes les informations à mémoriser dans le réseau. On distingue en général deux modes d'apprentissage :

III.6.1 Apprentissage non supervisé

L'apprentissage est qualifié de non supervisé lorsque seules les valeurs d'entrée sont disponibles. Dans ce cas, les exemples présentés à l'entrée provoquent une auto-adaptation du réseau afin de produire des valeurs de sortie qui soient proches en réponse à des valeurs d'entrée similaires (de même nature). (Figure III.8).

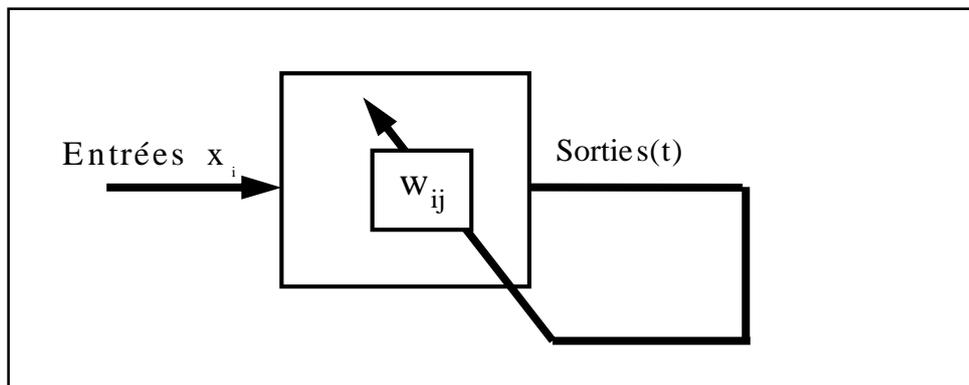
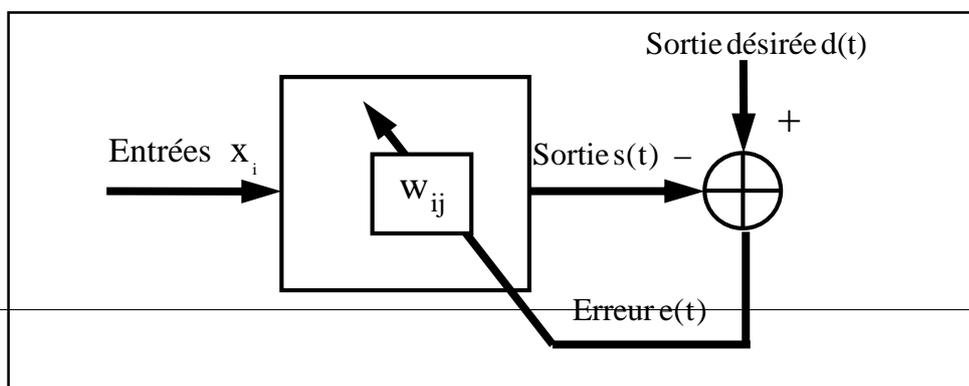


Fig. III.8 : Apprentissage non supervisé

III.6.2 Apprentissage supervisé

Dans ce type d'apprentissage (figure III.9), le réseau s'adapte par comparaison entre le résultat qu'il a calculé, en fonction des entrées fournies, et la réponse attendue en sortie, le résultat de cette comparaison c'est à dire de cette différence est le signal d'erreur. Ainsi les paramètres du réseau vont se modifier d'une manière itérative en fonction du signal d'erreur déjà calculé jusqu'à ce que les sorties

soient peu obtenues.



désirées près

Fig . III.9 : Apprentissage supervisé

III.7 Le perceptron multicouche

Le réseau perceptron multicouche de type Feed-Forward est le plus approprié dans notre étude pour sa capacité à résoudre les problèmes non-linéaires. Nous allons le décrire ici.

Le perceptron multicouche est un réseau comportant L couches, chaque neurone d'une couche étant totalement connecté aux neurones de la couche suivante.

Son architecture est donnée dans la figure III.10.

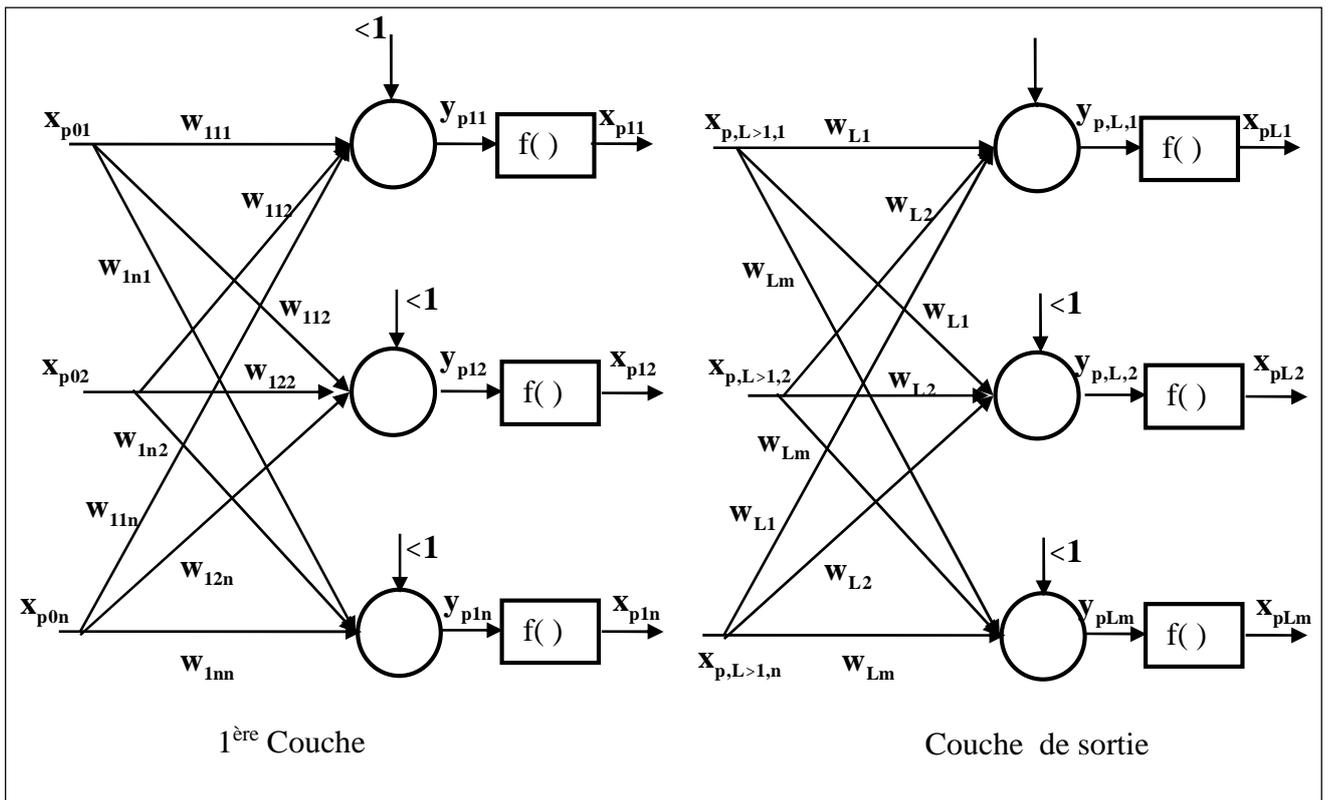


Fig . III.10 : Perceptron multicouche

L'apprentissage est supervisé, c'est-à-dire que l'on présente au réseau, en même temps, une forme et son modèle. Chaque neurone k est un automate linéaire généralisé dont la fonction de transfert f est supposée sigmoïdale, dans ce cas là la dérivabilité joue un rôle important.

L'apprentissage dans ce type de structure consiste à appliquer des couples (entrées, sorties désirées) à l'entrée du réseau.

Une sortie réelle est calculée pour chaque neurone de la $j^{\text{ème}}$ couche. Ce calcul est effectué de proche en proche la couche d'entrée vers la couche de sortie, celle-ci est appelée « propagation d'avant ». Ensuite l'erreur est calculée puis propagée dans le réseau, donnant lieu à une modification des poids.

On considère un réseau comportant une couche d'entrée à n neurones, une couche de sortie à m neurones et il comporte une à plusieurs couches cachées.

Supposons qu'on dispose d'un ensemble d'apprentissage composé de k paires de vecteurs :

$$(x_1, o_1), (x_2, o_2), \dots, (x_k, o_k) \quad (\text{III.2})$$

avec :

$$x_p = (x_{p01}, x_{p02}, \dots, x_{p0n})^T \quad \text{vecteur d'entrée} \quad (\text{III.3})$$

$$o_p = (o_{p01}, o_{p02}, \dots, o_{p0m})^T \quad \text{vecteur des sorties désiré} \quad (\text{III.4})$$

$$y_p = (y_{p11}, y_{p12}, \dots, y_{p1m})^T \quad \text{vecteur des sorties réel du réseau} \quad (\text{III.5})$$

L'entrée totale du k nœud pour la couche j est :

$$y_{pjk} = \sum_{i=0}^n w_{jki} x_{p,j-i,i} \quad (\text{III.6})$$

La sortie de ce nœud sera :

$$x_{pjk} = f(y_{pjk}) \quad (\text{III.7})$$

où :

w_{jki} : la connexion entre le neurone k de la couche $(j-1)$ et le neurone i de la couche j .

y_{pkj} : l'entrée totale du neurone k pour l'échantillon p de la couche j .

$w_{jk0} = \theta_{jk}$ le poids fictif du neurone k de la couche j correspondant à un biais dont l'entrée est fixée à 1.

III.8 Mise en œuvre des réseaux neuronaux

La démarche consiste en quatre étapes principales :

Etape 1 : fixer le nombre de couches cachées

Mis à part les couches d'entrée et de sortie, l'analyste doit décider du nombre de couches intermédiaires ou cachées. Sans couche cachée, le réseau n'offre que de faibles possibilités d'adaptation ; avec une couche cachée, il est capable, avec un nombre suffisant de neurones, d'approximer toute fonction continue. Une seconde couche cachée prend en compte les discontinuités éventuelles.

Etape 2 : déterminer le nombre de neurones par couches cachées

Chaque neurone supplémentaire permet de prendre en compte des profils spécifiques des neurones d'entrée. Un nombre plus important permet donc de mieux coller aux données présentées mais diminue la capacité de généralisation du réseau. Il n'existe plus de règle générale mais des règles empiriques.

Etape 3 : choisir la fonction d'activation

Nous considérerons la fonction sigmoïde pour le passage de la couche d'entrée à la couche cachée. Le passage de cette dernière à la couche de sortie sera soit linéaire, soit selon nos types de variables.

Etape 4 : choisir l'apprentissage

L'apprentissage par rétro-propagation nécessite la détermination du paramètre d'ajustement des poids synaptiques à chaque itération.

La détermination du critère d'arrêt est aussi cruciale dans la mesure où la convergence peut passer par des minima locaux.

Il existe trois types de réseaux neuronaux multicouches de type Feedforward, chacun étant basé sur une base de fonctions d'activation différente et possédant une procédure d'initialisation adaptée aux paramètres de la fonction d'activation choisie.

III.9 Algorithmes d'apprentissage

Nous nous intéresserons aux algorithmes utilisés dans le réseau neuronal multicouche de type Feedforward (MLP). Les algorithmes d'apprentissage qui peuvent être utilisés pour ce type de modèle neuronal sont les suivants:

- Algorithme de la rétro-propagation (GBP : Gradient Backpropagation).
 - La rétro-propagation du gradient avec momentum.
 - Algorithme de la rétro-propagation du gradient à convergence accélérée par l'adaptation du coefficient d'apprentissage.
 - Algorithme du gradient conjugué CG.
-

- Algorithme du gradient conjugué modifié MCG.
- Méthode de Newton.
- Méthodes quasi-newtoniennes.

III.9.1 Retropropagation du gradient

L'algorithme de rétro-propagation a été développé en particulier par Rumelhart et Parkenet le Cun en 1985. Cet algorithme repose sur la minimisation de l'erreur quadratique entre les sorties calculées et celles souhaitées.

Le terme rétro-propagation du gradient provient du fait que l'erreur calculée en sortie est transmise en sens inverse vers l'entrée.

L'erreur commise sur le $k^{\text{ième}}$ nœud de sortie est :

$$\delta_{pk} = O_{pk} - x_{plk} \quad (\text{III.8})$$

Par conséquent l'erreur totale (pour tous les nœuds) est :

$$E_p = \frac{1}{2} \sum_{k=1}^m \delta_{pk}^2 = \frac{1}{2} \sum_{k=1}^m (O_{pk} - x_{plk})^2 \quad (\text{III.9})$$

Pour minimiser E_p , on calcule son gradient par rapport à chaque poids w , puis on modifie les poids dans le sens inverse du gradient.

La mise à jour des poids diffère de la couche de sortie et les couches cachées.

■ Mise à jour des poids de la couche de sortie

$$w_{lkj}(t+1) = w_{lkj}(t) + \mu \cdot e_{p,l,k} \cdot x_{p,l-1,k} \quad (\text{III.10})$$

$$e_{p,l,k} = (O_{pk} - x_{plk}) \cdot f'(y_{plk}) \quad (\text{III.11})$$

où μ est le pas d'apprentissage $0 < \mu < 1$

■ Mise à jour des poids des couches cachées

$$w_{lkj}(t+1) = w_{lkj}(t) + \mu \cdot e_{p,l-1,j} \cdot x_{p,l-2,i} \quad (\text{III.12})$$

$$e_{p,l-1,j} = f'(y_{p,l-1,k}) \sum_{k=1}^m e_{p,l,k} w_{l,k,j} \quad (\text{III.13})$$

■ Résumé de l'algorithme de rétro-propagation

1. Appliquer un vecteur d'entrée $x_p = (x_{p01}, x_{p02}, \dots, x_{p0n})^T$ aux nœuds d'entrées puis initialiser les poids du réseau;
2. Exécuter l'échantillon d'apprentissage à travers le réseau;

3. Calculer les termes d'erreur de signal de la couche de sortie et les couches cachées en utilisant l'équation (III.12) et l'équation (III.13) respectivement;
4. Mise à jour les poids de la couche de sortie et couches cachées en utilisant l'équation (III.10) et l'équation (III.11) respectivement;
5. Répéter ce processus jusqu'à ce que l'erreur E_p devienne acceptable (aller à étape 2).

III.9.2 Accélération de l'algorithme avec le momentum

La convergence du réseau par rétro-propagation est un problème crucial car il requiert de nombreuses itérations. Pour pallier à ce problème, un paramètre est souvent rajouté pour accélérer la convergence. Ce paramètre est appelé « le momentum ».

Le momentum est un moyen efficace pour accélérer l'apprentissage et aussi pour pouvoir sortir des minimums locaux.

La règle de mise à jour des poids devient alors :

$$w_{j,ki}(t+1) = w_{j,ki}(t) + e_{p,j,k} \cdot x_{p,j-1,i}(t) + \Omega(w_{j,ki}(t) - w_{j,ki}(t-1))$$

(III.14)

Ω : est la constante du momentum.

III.9.3 Méthodes d'optimisation du second ordre

Les méthodes d'optimisation du second ordre sont des méthodes itératives de descente du gradient qui consistent à remplacer la fonction de coût par son approximation quadratique au voisinage de point courant.

$$Q(s) = J_k(w_k)g(w_k)^T s + s^T \frac{1}{2} G(w_k)s$$

(III.15)

J : la fonction de coût.

$$s = w_{k+1} - w_k$$

$$g(w_k) = \nabla_w J : \text{le gradient}$$

$$G(w_k) = \nabla_w^2 J : \text{le hessien}$$

On distingue de nombreuses méthodes :

- La méthode de Newton consiste à calculer $w(k+1)$ de manière à minimiser $Q(s)$ de l'équation (III.15). La formule itérative devient :

$$w_{k+1} = w_k - [G(w_k)]^{-1} g(w_k)$$

(III.16)

- Le principe des méthodes quasi-newtoniennes consiste en une généralisation de la méthode de Newton de l'équation. Le Hessien (ou le gradient) est approximé par une méthode itérative. Les améliorations successives ont conduit à différents algorithmes :
 - Davidon-Fletcher-Powell (DFP)
 - Broyden-Fletcher-Goldfarb-Shanno (BFGS)
 - Levenberg-Marquardt (LM)

Une méthode « quasi-newton » est efficace dans le voisinage de la solution minimale pour des problèmes de grandes dimensions.

III.10 Boîte à outils Matlab Réseaux de Neurones-(Neural Network toolbox)

Le logiciel Matlab contient une boîte à outils de réseaux de neurones comportant de nombreuses fonctions pour la création de réseaux de neurones, d'apprentissage et de simulation.

III.10.1 Création d'un réseau

La fonction de création d'un réseau est spécifique au modèle de réseau utilisé (`newc`, `newlvq`, etc).

Pour les réseaux multicouches, la création du réseau est commandée par la fonction `newff` :

réseau = `newff`(PR, [S₁ S₂.....S_{N1}] , {TF₁ TF₂.....TF_{N1}} , BTF , BLF , PF) ;

Avec :

- PR : Plage des variations des entrées.
- S_i : nombre des neurones dans la couche i, pour N₁ couches.
- TF_i : fonction d'activation dans la couche i, par défaut la fonction d'activation est 'tansig',
- BTF : l'algorithme d'apprentissage par paquets du réseau, la fonction BTF peut être :
 - `trainlm` : apprentissage par l'algorithme de Levenberg-Marquardt
 - `trainbfg` : apprentissage par l'algorithme BFGS.
 - `trainoss` : apprentissage par l'algorithme « one-step BFGS Method »
 - `trainbr` : version de `trainlm` avec modération automatique des poids.
 - `trainrp` : apprentissage par l'algorithme RPROP.
 - `trainscg` : apprentissage par scaled conjugate gradient (SCG)
 - `traincgf` : apprentissage par la méthode du gradient conjugué+FLETCHER-REEVES.
 - `traincp` : apprentissage par la méthode du gradient conjugué+POLAK-RIBIERE.
- BLF : l'algorithme d'apprentissage incrémental du réseau, la fonction BLF peut être :
 - `Learngd` : L'algorithme d'apprentissage sera la descente de gradient à taux d'apprentissage fixe.
 - `Learngdm` : version de `learngd` avec moment.
- PF : fonction du coût.

- mae : erreur absolue moyenne
- mse : erreur quadratique moyenne
- msereg : version de mse avec modérations des poids
- sse : somme des carrés des erreurs

III.10.2 Apprentissage

Il existe 2 types d'apprentissage :

- Incrémental : fonction adapt.
- Par paquets : fonction train.
 - Apprentissage incrémental (en-ligne, on-line) : les poids sont modifiés à chaque présentation d'une entrée.
 - Apprentissage par paquets (hors-ligne, off-line, batch mode) : les poids sont modifiés uniquement après présentation de toutes les entrées.

III.10.3 Simulation (ou activation) d'un réseau

$A = \text{sim}(\text{net}, p)$;

où net est le pointeur retourné par une fonction de création de réseau.

III.11 Application des réseaux de neurones

Les réseaux de neurones, se trouvent à l'intersection de différents domaines (informatique, électroniques, science cognitive, neurobiologie et même philosophie), ont de nombreuses applications possibles. En voici quelque exemple.

- Approximation de fonctions : les fonctions trop compliquées peuvent être approximées, grâce au réseau, par une somme de fonctions plus simples comme des polynômes ou des sigmoïdes.
- Optimisation de trajectoire : en intégrant les paramètres tels que le vent, les conditions extérieures,...on peut, par exemple, déterminer quelle est la meilleure trajectoire pour un avion, une fusée...
- Reconnaissance : un réseau peut servir à reconnaître des caractères. Cela est déjà utilisé à la poste pour lire les codes postaux, ou même dans certaines banques pour lire les chèques.il est possible de retrouver la prononciation des mots à partir d'un texte.
- Prévission : on utilise de plus en plus les réseaux pour faire des prévisions en marketing (prédiction de comportement, de possibilité de vente d'un produit,..) ou pour le trafic routier...mais les prévisions en météo ou en bourse sont trop compliquées à réaliser.
- Contrôle : on peut contrôler les produits dans une industrie.
- Robotique : certains robots sont dotés des réseaux de neurones. Des entreprises japonaises se ventent déjà à de leur utilisation, même pour des produits électroménagers ou informatiques.
- Télécommunications et informatique : analyse du signal, compression de données...

- Industrie : contrôle qualité, diagnostic de panne, corrélations entre les données fournies par différents capteurs, analyses de signature manuscrite...
- Les réseaux de neurones ont encore beaucoup d'autres applications possibles. En général, cela se résume en un problème d'optimisation.

III.12 Conclusion

Dans ce chapitre, nous avons présenté les idées de base sur l'optimisation par réseaux de neurones. Nous avons décrit:

- Le neurone biologique et artificiel.
 - Les trois grandes catégories de réseaux de neurones artificiels (Réseau neuronal multicouche de type Feedforward, Réseau récurrent et Réseau cellulaire).
 - Les différentes méthodes d'apprentissage qui peuvent être utilisés pour les réseaux neuronaux multicouches de type Feedforward.
 - Les principaux algorithmes d'apprentissage qui peuvent être utilisés pour les réseaux neuronaux multicouches de type Feedforward.
-

CHAPITRE IV : SIMULATIONS ET COMMENTAIRES

IV.1 Introduction

Dans ce chapitre, nous allons décrire les démarches que nous avons utilisées et les résultats obtenus afin d'optimiser le diagramme de rayonnement des réseaux d'antennes par les réseaux de neurones pour répondre à différentes exigences.

IV.2 Application de réseaux de neurones à la synthèse de réseau d'antennes linéaires

Nous allons appliquer le modèle neuronal au domaine de la synthèse des antennes linéaires dans le but de remplacer deux méthodes : les méthodes de Dolph-Tchebycheff et l'algorithme génétique par une autre plus performante au niveau de vitesse de convergence. La modélisation se fait par un choix judicieux des valeurs des pondérations de la base d'apprentissage.

IV.2.1 Procédure de développement d'un réseau de neurones

Les procédures de développement des réseaux de neurones après l'examen des propriétés, peuvent être divisées en plusieurs étapes :

■ Collecte et analyse des données

L'objectif de cette étape est de recueillir des données, à la fois pour développer le réseau de neurones et pour le tester. Dans notre cas elles sont constituées par les pondérations des réseaux de Dolph-Chebychev et les réseaux génétiques. Dans ce dernier cas, nous utiliserons le logiciel développé par un étudiant de l'institut.

Il est préférable d'effectuer une analyse des données de manière à déterminer les caractéristiques discriminantes pour détecter ou différencier les données. Ces caractéristiques constituent l'entrée du réseau de neurones.

Cette détermination des caractéristiques a des conséquences sur le temps de développement (temps d'apprentissage). Une étude statistique sur les données peut permettre d'écarter celles qui sont aberrantes et redondantes. Dans le cas d'un problème de classification, il appartient à l'expérimentateur de déterminer le nombre de classes auxquelles ses données appartiennent et de déterminer pour chaque donnée la classe à laquelle elle appartient.

■ Choix d'un réseau de neurones

Il existe un grand nombre de types de réseaux de neurones, avec pour chacun des avantages et des inconvénients. Le choix d'un réseau peut dépendre :

- de la tâche à effectuer,
- de la nature des données,
- d'éventuelles contraintes d'utilisation.

Pour notre problème, on était intéressé par l'utilisation d'un réseau de neurones artificiels multicouche (perceptron) au vu de sa simplicité et de son usage dans la reconnaissance des formes (Pattern recognition), traitement des images, analyse des antennes, détection d'angle d'arrivée et capacité de résoudre des équations non- linéaires,.....

Comme il n'existe pas de méthode pour créer un réseau de neurones optimal, nous avons pu choisir les paramètres (nombre des couches cachées, nombre de neurones dans chaque couche, ...) convenables par de nombreux essais.

■ **Logiciel Matlab**

MATLAB est un logiciel interactif permettant d'effectuer des calculs numériques complexes particulièrement utiles dans le domaine de l'ingénierie. Ce logiciel fournit un langage et des outils de développement de haut niveau nous permettant de développer et d'analyser rapidement nos algorithmes et nos applications. Il permet l'apprentissage et une visualisation de l'erreur commise du réseau de neurones.

■ **Base d'apprentissage et mise en forme des données pour un réseau de neurones**

Afin de pouvoir générer l'apprentissage du réseau, il est nécessaire de créer une base d'apprentissage. Comme l'apprentissage est supervisé, cette base doit contenir à la fois l'entrée du réseau et la sortie souhaitée. Lorsque la phase d'apprentissage est terminée, on procède de la manière suivante :

1. Test du réseau.
2. Si le test fournit de bons résultats, alors le réseau sera prêt à calculer la sortie correspondant à une entrée donnée. Si non, il faut régler les paramètres du système, pour augmenter les performances du réseau, et répéter la phase précédente.

De manière générale, les bases de données doivent subir une phase de prétraitement (Preprocessing) et une phase de post-traitement (Postprocessing) afin, d'être adaptées aux entrées et aux sorties du réseau de neurones et de rendre la formation de réseau neurologique plus efficace.

- Le prétraitement des données

Un prétraitement consiste à éliminer les discontinuités artificielles dans l'espace de fonction d'entrée et de ramener les entrées de problème à un ensemble d'information approprié.

Ensuite il faut faire une normalisation appropriée, qui tienne compte de l'amplitude des valeurs acceptées par le réseau.

IV.2.2 Algorithme et paramètres d'apprentissage

Tous les modèles de réseaux de neurones requièrent un apprentissage. Plusieurs types d'apprentissages peuvent être adaptés à un même type de réseau de neurones. Les critères de choix sont souvent la rapidité de convergence ou les performances de généralisation.

Le critère d'arrêt de l'apprentissage est souvent calculé à partir d'une fonction de coût, caractérisant l'écart entre les valeurs de sortie obtenues et les valeurs de références (réponses souhaitées pour chaque exemple présenté).

Les possibilités offertes par le logiciel (Matlab) sur les méthodes d'apprentissage sont assez étendues (voir annexe). Celle retenue est l'apprentissage de Levenberg-Marquardt, technique reposant sur une minimisation d'un critère d'erreur de type moindre carré.

IV.2.3 Validation et résultats de simulation

Une fois le réseau de neurones entraîné (après apprentissage), il est nécessaire de le tester sur une base de données différente de celles utilisées pour l'apprentissage. Ce test permet à la fois d'apprécier les performances du système neuronal et de détecter le type de données qui pose problème. Si les performances ne sont pas satisfaisantes, il faudra soit modifier l'architecture du réseau, soit modifier la base d'apprentissage (caractéristiques discriminantes ou représentativité des données de chaque classe). L'évolution de fonction d'erreur pendant la phase d'apprentissage se visualise par l'intermédiaire d'un graphe. Il illustre la convergence du réseau de neurones par rapport au fichier d'apprentissage.

IV.3 Programmation de la méthode de Dolph-Tchebycheff

Une grande partie de notre travail a été de programmer la méthode de synthèse de Dolph-Tchebycheff pour l'utiliser comme méthode d'apprentissage. Cette méthode a été décrite dans le chapitre II.

Nous l'avons programmé en utilisant l'outil « Symbolic Math Toolbox » de logiciel Matlab.

La commande "sym" permet d'exprimer des variables et des expressions symboliques.

L'organigramme de la méthode est représenté en figure IV.1. Il permet de calculer les coefficients de pondération du réseau quelque soit le nombre d'antennes.

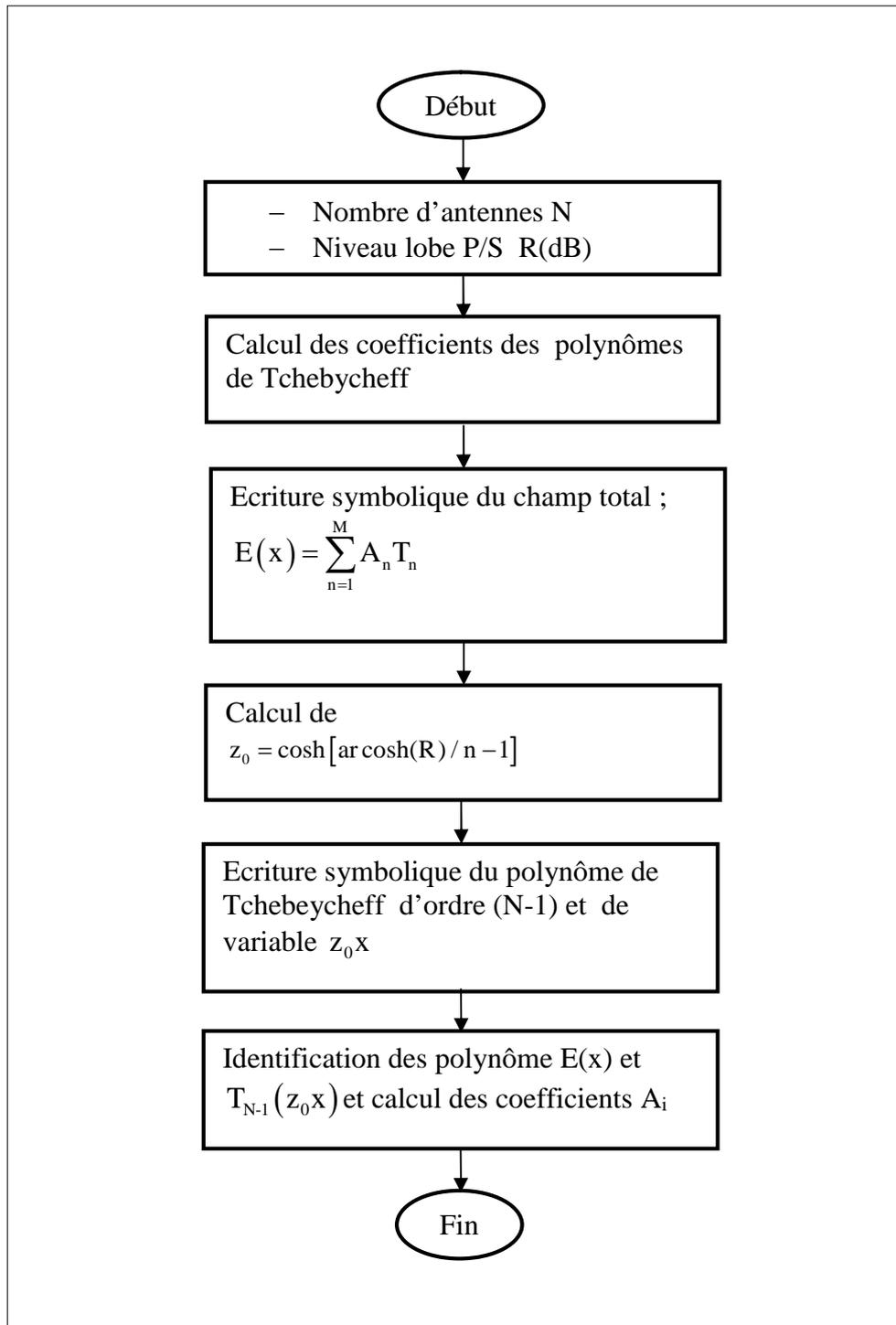


Fig IV.1: Organigramme de synthèse par la méthode Dolph- Tchebycheff

Nous avons représenté en figures IV.2, IV.3 et IV.4 quelques exemples de diagrammes obtenus par l'algorithme de Dolph. Les coefficients des pondérations respectifs sont donnés dans les tableaux IV.1, IV.2 et IV.3. On remarque que les lobes secondaires sont sur le même niveau.

Excitations synthétisées par Dolph-Chébychev	
	20dB
1	0.5799
2	0.6603
3	0.8751
4	1.0000
5	1.0000
6	0.8751
7	0.6603
8	0.5799

Tableau IV.1 Coefficients pour N=8 et R=20 dB

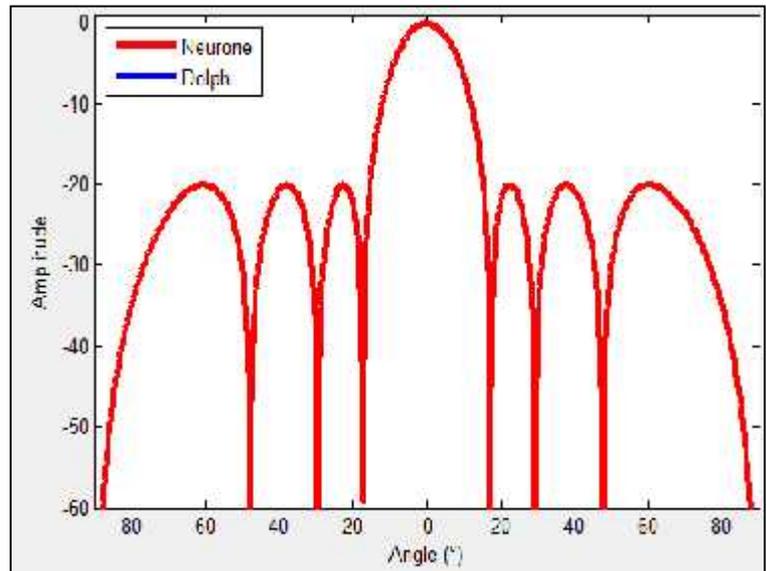


Fig IV. 2 : Diagramme pour N=8 et R=20 dB

Excitations synthétisées par Dolph-Chébychev	
	30dB
1	0.2622
2	0.5187
3	0.8120
4	1.0000
5	1.0000
6	0.8120
7	0.5187
8	0.2622

Tableau IV.2 Coefficients pour N=8 et R=30 dB

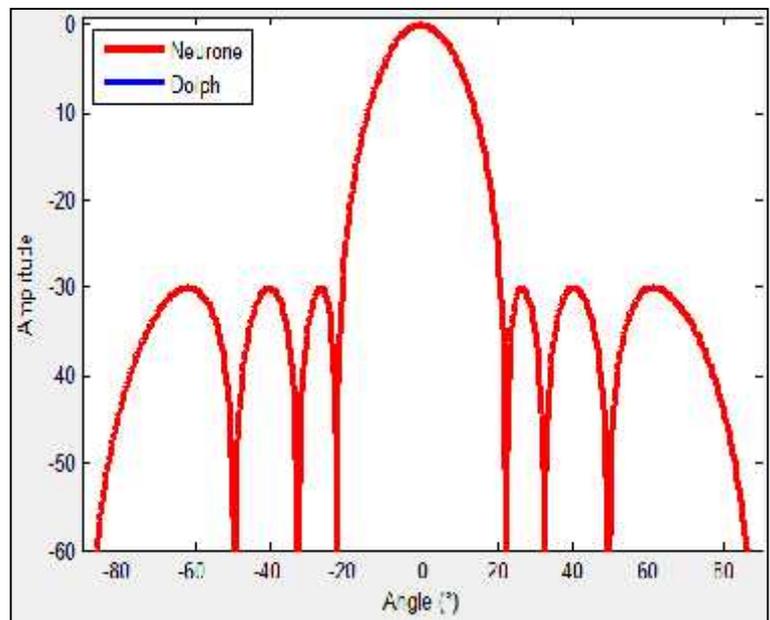


Fig IV.3 : Diagramme pour N=8 et R=30 dB

Excitations synthétisées par Dolph-Chébychev	
	40dB
1	0.1461
2	0.4179
3	0.7594
4	1.0000
5	1.0000
6	0.7594
7	0.4179
8	0.1461

Tableau IV.3 Coefficients pour N=8 et R=40 dB

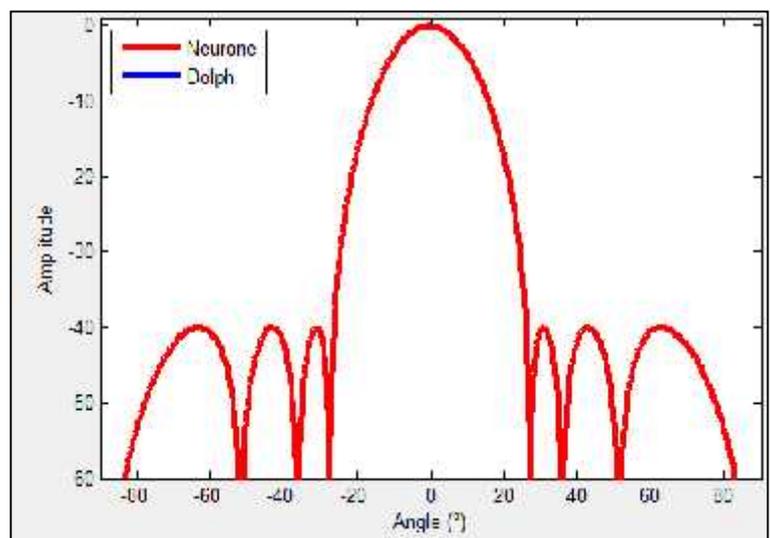


Fig IV.4 : Diagramme pour N=8 et R=40 dB

IV.4 Simulation des réseaux de neurones avec apprentissage par Dolph

■ Phase de création du réseau

Nous utilisons la fonction « newff » de l'outil de Matlab pour créer un réseau de neurone de type Feed-Forward .

La syntaxe est la suivante `net = newff(P,T,[N],{'tansig','tansig'})`

P : Entrées,

T : Sorties,

[N] : Nombre de neurones de la couche cachée,

{'tansig','tansig'} : Fonction de transfert de la couche cachée et Fonction de transfert de la couche de sortie.

Le code généré par cette fonction est donné en annexe. Il comprend les caractéristiques de l'architecture du réseau, des fonctions, des paramètres et des poids.

■ Phase d'apprentissage

Une fois que l'architecture du réseau a été décidée, la phase d'apprentissage permet de calculer les poids synaptiques menant à chaque neurone formel. Ceci est illustré dans la figure IV.5.

Nous avons choisi l'algorithme d'apprentissage de Levenberg-Marquardt (`trainlm`) qui a donné les meilleurs résultats.

Cet algorithme consiste à présenter au réseau des exemples d'apprentissage, c'est-à-dire des jeux d'activités des neurones d'entrée ainsi que ceux des activités des neurones de sortie. Il examine l'écart entre la sortie du réseau et la sortie souhaitée et on modifie les poids synaptiques des connexions jusqu'à ce que le réseau produise une sortie très proche de celle souhaitée.

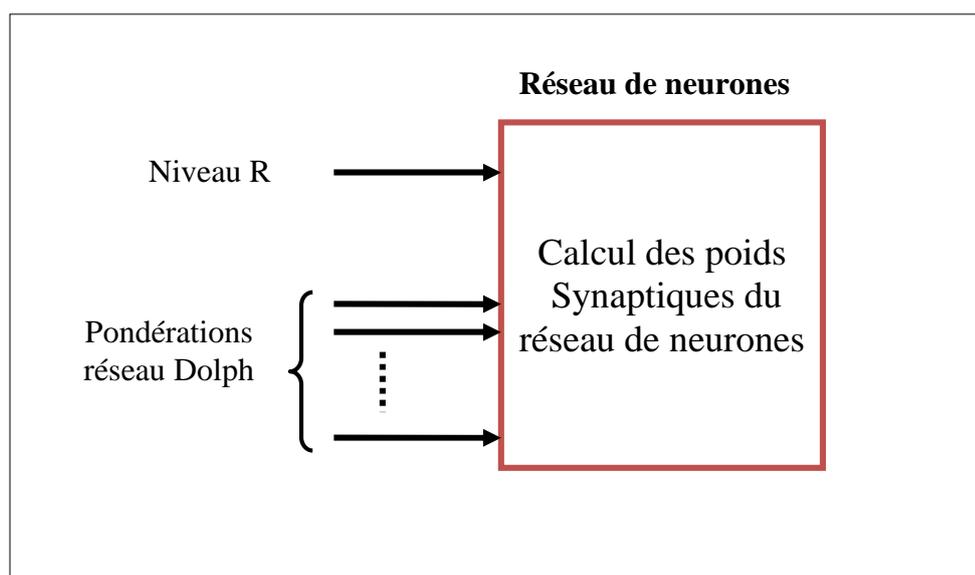


Fig IV.5: Phase d'apprentissage

La figure IV.6 montre l'évolution de l'erreur.

L'objectif essentiel ici, est de trouver le meilleur apprentissage qui permet de donner un bon modèle. Pour cela, plusieurs essais sont nécessaires, en agissant sur les paramètres influant sur l'apprentissage. Ces paramètres sont les suivants :

- le nombre de neurones dans la couche cachée,
- les fonctions d'activation,
- le pas d'apprentissage.

Après plusieurs essais, un réseau multicouche a été retenu avec la topologie suivante :

- 1 neurone dans la couche d'entrée.
- 7 neurones dans la couche cachée
- $N/2$ neurones dans la couche de sortie représentant la loi d'amplitude pour un réseau symétrique de N (pair) éléments rayonnants.

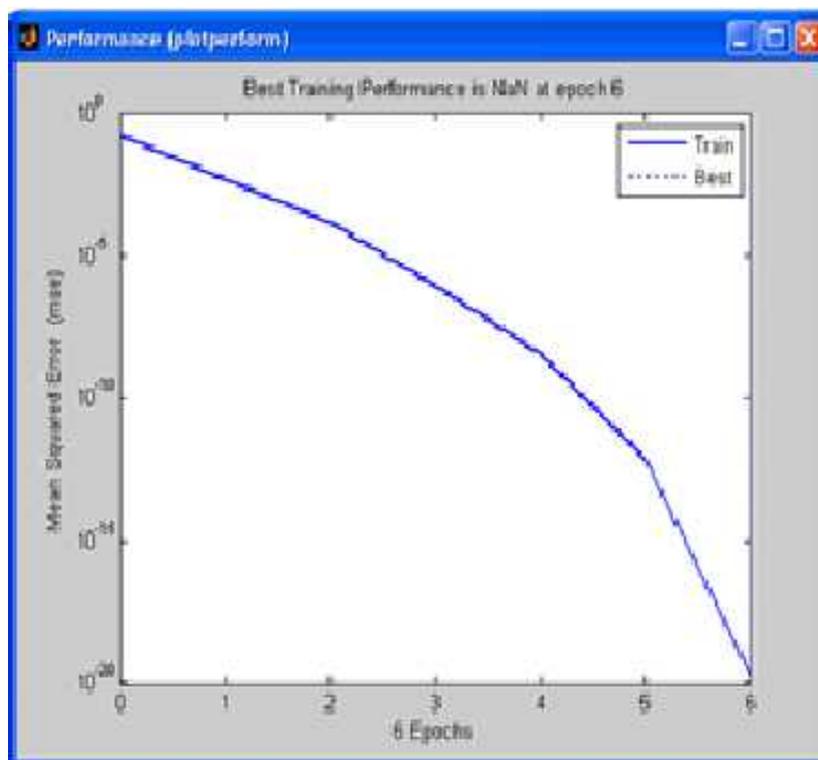


Fig IV.6 : Evolution de la fonction d'erreur

■ Phase de test

Cette phase consiste à vérifier si le réseau de neurones fonctionne convenablement. Pour cela, on lui présente des valeurs quelconques de niveau R à son entrée, puis à l'aide de la fonction « sym », on calcule les pondérations. Ceci est illustré dans la figure IV.7.

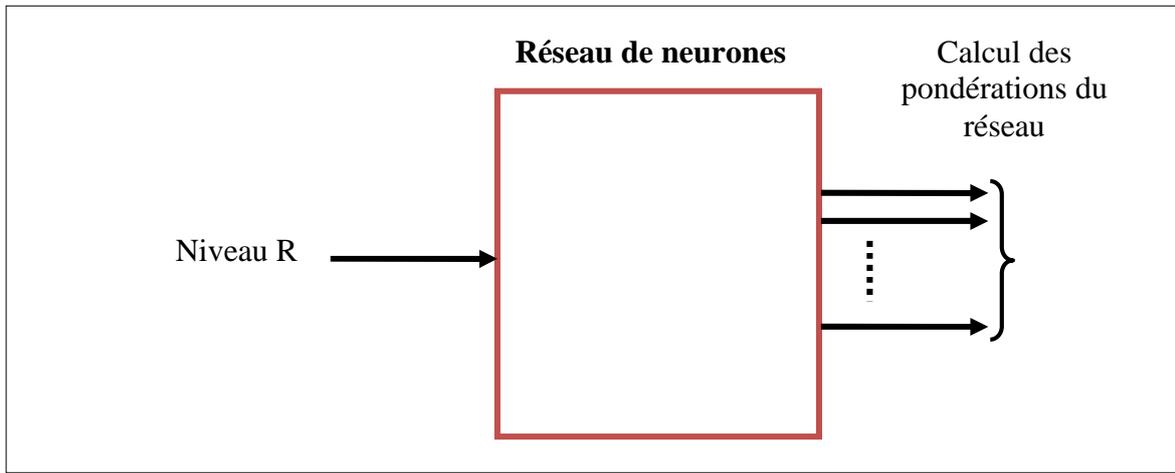


Fig IV.7: Phase de test

■ **Simulation d'un réseau de 8 antennes**

Comme le diagramme est symétrique, pour 8 antennes, il suffit de calculer 4 coefficients.

Les résultats de simulations sont représentés dans les figures suivantes.

Dans la figure IV.8, nous avons représenté en pointillés, la valeur des coefficients obtenus par la méthode de Dolph pour différentes valeurs de R (NLPS) pour un nombre d'antennes égal à 8. Les cercles représentent les pondérations obtenues lors de la phase d'apprentissage pour les valeurs R=20, 30, 40 et 50dB. On remarque que ces cercles sont situés sur les lignes en pointillés, ce qui signifie que l'erreur pendant la phase d'apprentissage est faible. Les étoiles représentent les pondérations obtenues pendant les tests (R=25, 35, 45 et 55). On remarque que le réseau de neurone fonctionne très bien, puisque les pondérations sont identiques à ceux obtenus par Dolph.

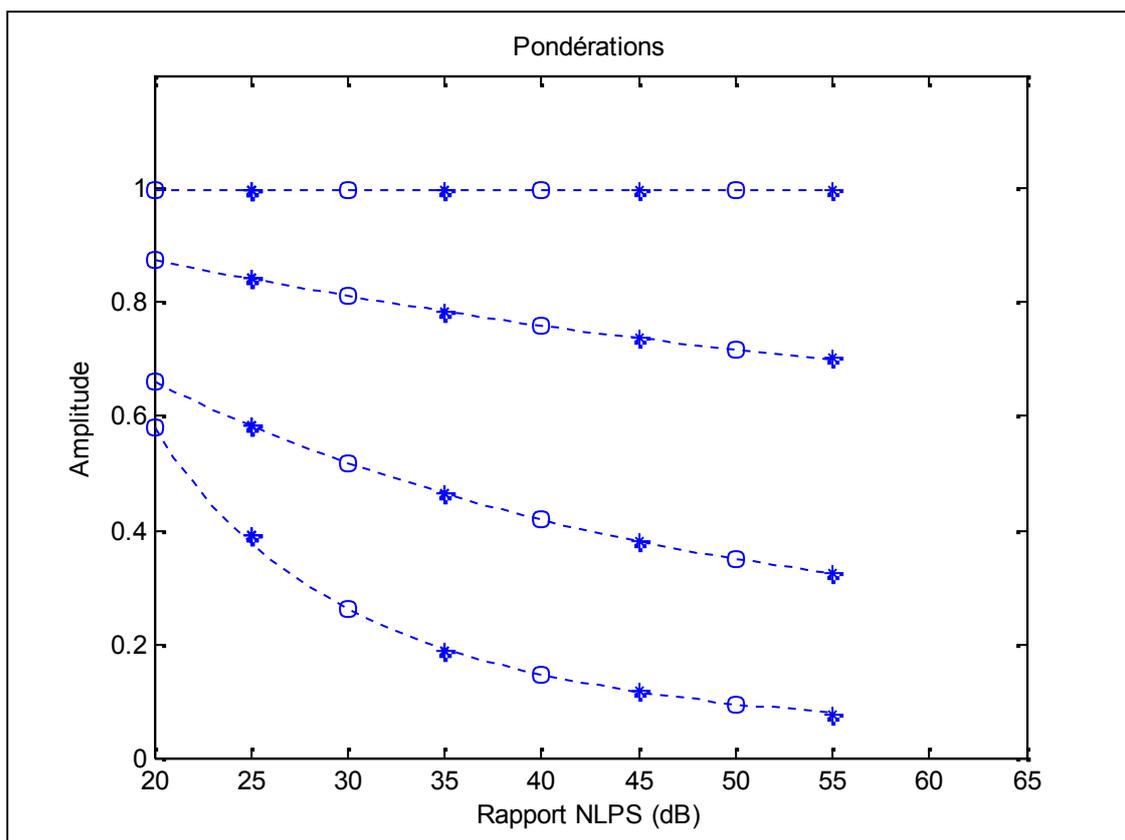


Fig IV.8 : Test pour R=25, 35, 45 et 55dB ;N=8

Les figures de IV.9 à IV.12 montrent les diagrammes obtenus par la méthode de Dolph (en bleu) et le réseau de neurones (en rouge) pour différentes valeurs de R. On remarque leur similitude sauf pour la valeur de R=55dB. Ceci peut s'expliquer par le fait qu'on est assez éloigné de la valeur extrême de 50 dB qui a servi dans l'apprentissage.

Les tableaux IV.4 à IV.7 donnent les coefficients obtenus avec les deux méthodes.

Excitations synthétisées par (R=25dB)		
	Dolph	RN
1	0.3778	0.3916
2	0.5843	0.5850
3	0.8424	0.8426
4	1.0000	1.0000
5	1.0000	1.0000
6	0.8424	0.8426
7	0.5843	0.5850
8	0.3778	0.3916

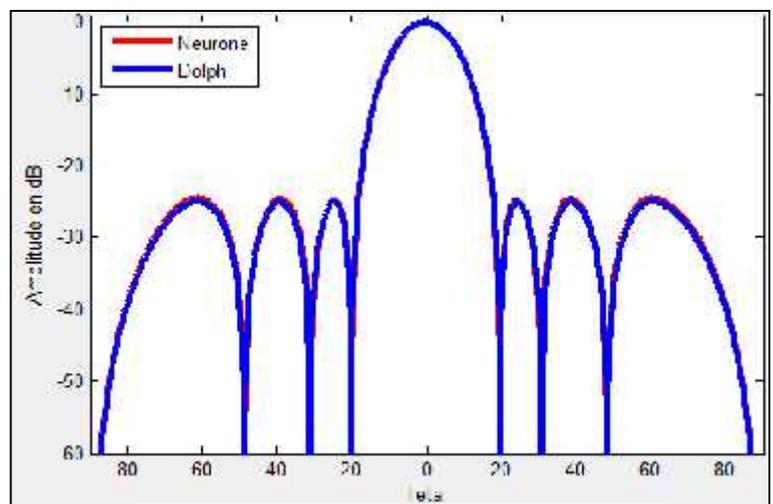


Fig IV.9: Test pour R=25dB N=8

Tableau IV.4 : Pondérations Dolph et RN
R=25dB N=8

Excitations synthétisées par (R=35dB)		
	Dolph	RN
1	0.1915	0.1885
2	0.4636	0.4637
3	0.7843	0.7843
4	1.0000	1.0000
5	1.0000	1.0000
6	0.7843	0.7843
7	0.4636	0.4637
8	0.1915	0.1885

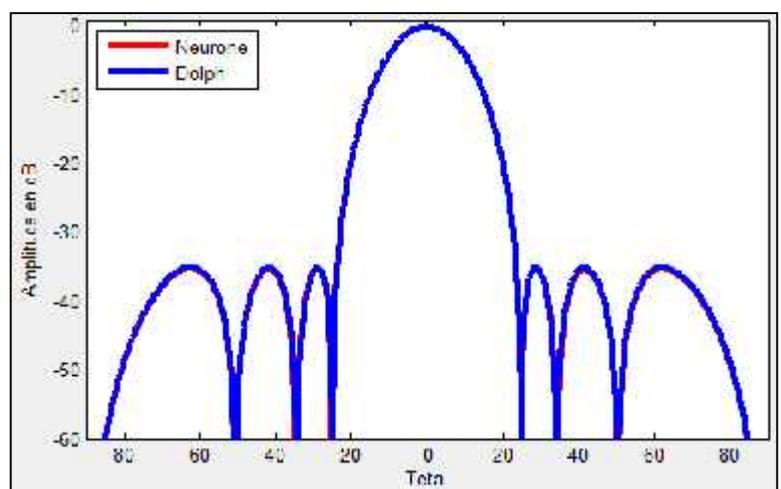


Fig IV. 10: Test pour R=35dB N=8

Tableau IV.5 : Pondérations Dolph et RN
R=35dB N=8

Excitations synthétisées par (R=45dB)		
	Dolph	RN
1	0.1157	0.1176
2	0.3803	0.3800
3	0.7375	0.7374
4	1.0000	1.0000
5	1.0000	1.0000
6	0.7375	0.7374
7	0.3803	0.3800
8	0.1157	0.1176

Tableau IV.6 : Pondérations Dolph et RN
R=45dB N=8

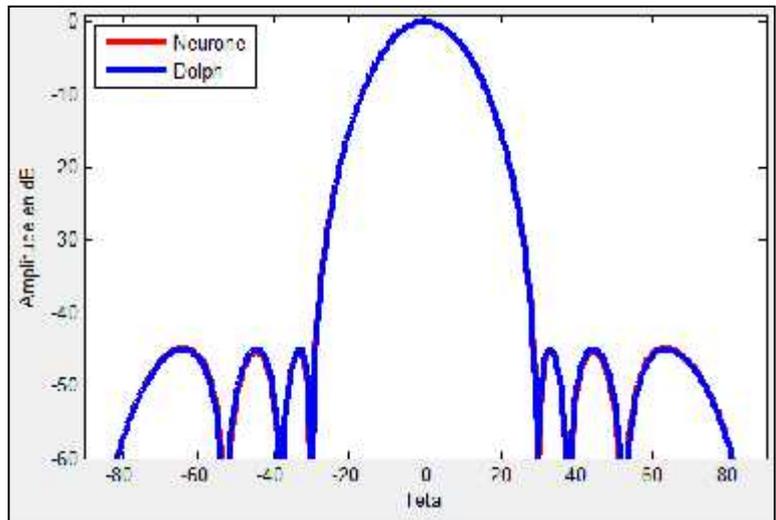


Fig IV.11: Test pour R=45dB N=8

Excitations synthétisées par (R=55dB)		
	Dolph	RN
1	0.0795	0.0737
2	0.3240	0.3254
3	0.7014	0.7020
4	1.0000	1.0000
5	1.0000	1.0000
6	0.7014	0.7020
7	0.3240	0.3254
8	0.0795	0.0737

Tableau IV.7 : Pondérations Dolph et RN
R=55dB N=8

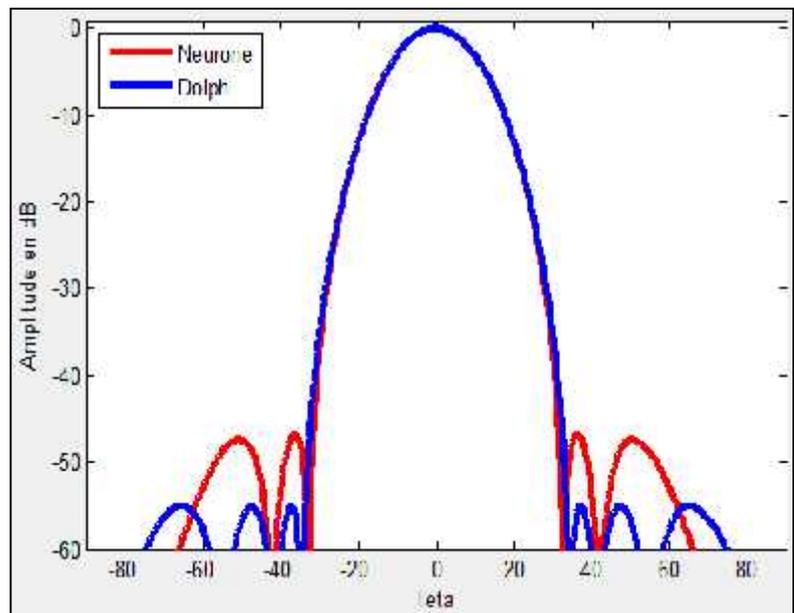


Fig IV. 12: Test pour R=55dB N=8

■ **Simulation pour un réseau de 16 antennes**

Nous avons repris les mêmes simulations antérieures avec un réseau de 16 antennes en calculant 8 coefficients.

Les résultats sont illustrés dans les figures de IV.13 à IV.17 et les tableaux IV.8 à IV.11.

On remarque là aussi que le réseau de neurone a bien fonctionné sauf pour la valeur de R =55dB..

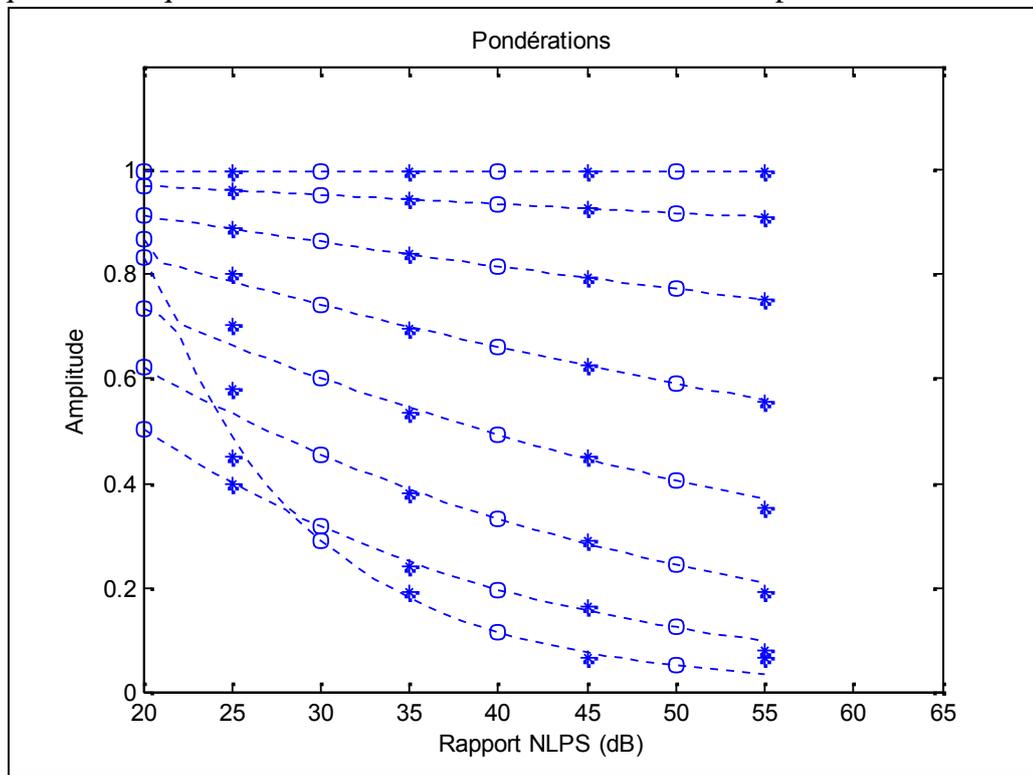


Fig IV. 13 : Test pour R=25,35, 45, et 55 dB ; N=16

Excitations synthétisées par (R=25dB)		
	Dolph	RN
1	0.4018	0.4020
2	0.4907	0.4511
3	0.5334	0.5832
4	0.6651	0.7054
5	0.7867	0.8012
6	0.8884	0.8889
7	0.9617	0.9616
8	1.0000	1.0000
9	1.0000	1.0000
10	0.9617	0.9616
11	0.8884	0.8889
12	0.7867	0.8012
13	0.6651	0.7054
14	0.5334	0.5832
15	0.4907	0.4511
16	0.4018	0.4020

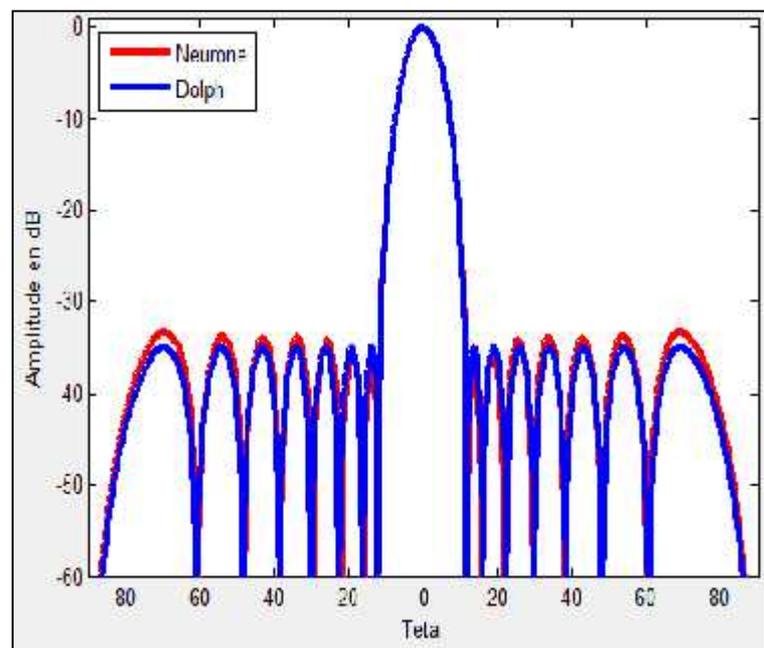


Tableau IV.8 : Pondérations Dolph et RN
R=25dB N=16

Fig IV.14: Test pour R=25dB N=16

Excitations synthétisées par (R=35dB)

	Dolph	RN
1	0.1791	0.1921
2	0.2496	0.2363
3	0.3888	0.3748
4	0.5442	0.5337
5	0.7005	0.6966
6	0.8396	0.8393
7	0.9440	0.9440
8	1.0000	1.0000
9	1.0000	1.0000
10	0.9440	0.9440
11	0.8396	0.8393
12	0.7005	0.6966
13	0.5442	0.5337
14	0.3888	0.3748
15	0.2496	0.2363
16	0.1791	0.1921

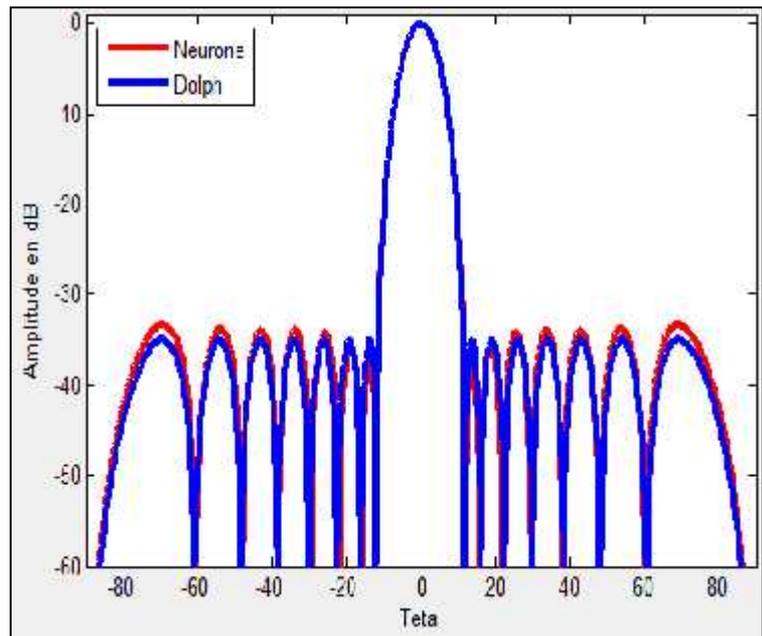


Tableau IV.9 : Pondérations Dolph et RN
R=35dB N=16

Fig IV.15: Test pour R=35dB N=16

Excitations synthétisées par (R=45dB)		
	Dolph	RN
1	0.0744	0.0704
2	0.1549	0.1559
3	0.2841	0.2843
4	0.4466	0.4463
5	0.6249	0.6238
6	0.7941	0.7931
7	0.9269	0.9265
8	1.0000	1.0000
9	1.0000	1.0000
10	0.9269	0.9265
11	0.7941	0.7931
12	0.6249	0.6238
13	0.4466	0.4463
14	0.2841	0.2843
15	0.1549	0.1559
16	0.0744	0.0704

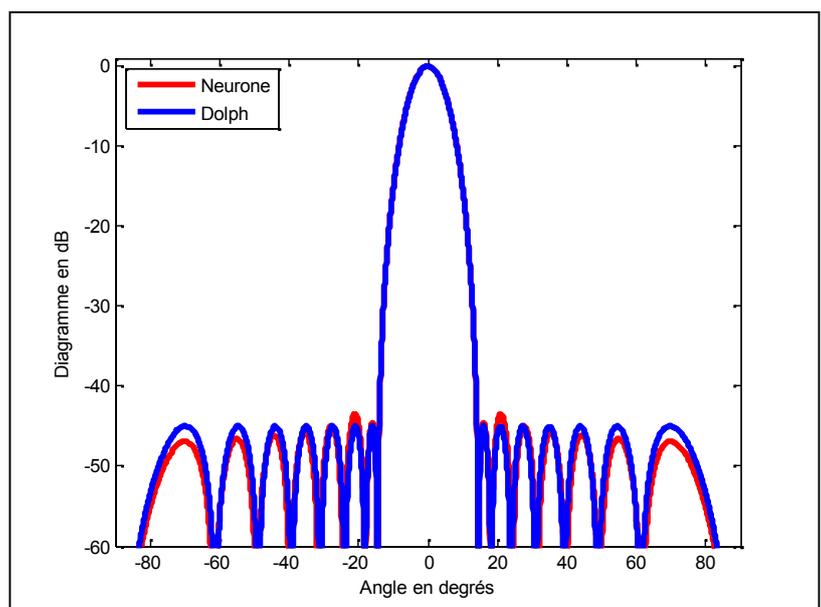


Fig IV10: Pondérations Dolph et RN
R=45dB N=16

Fig IV.16: Test pour R=45dB N=16

Excitations synthétisées par (R=55dB)		
	Dolph	RN
1	0.0343	0.0430
2	0.0981	0.0995
3	0.2104	0.2152
4	0.3696	0.3760
5	0.5602	0.5679
6	0.7529	0.7586
7	0.9109	0.9131
8	1.0000	1.0000
9	1.0000	1.0000
10	0.9109	0.9131
11	0.7529	0.7586
12	0.5602	0.5679
13	0.3696	0.3760
14	0.2104	0.2152
15	0.0981	0.0430
16	0.0343	0.0995

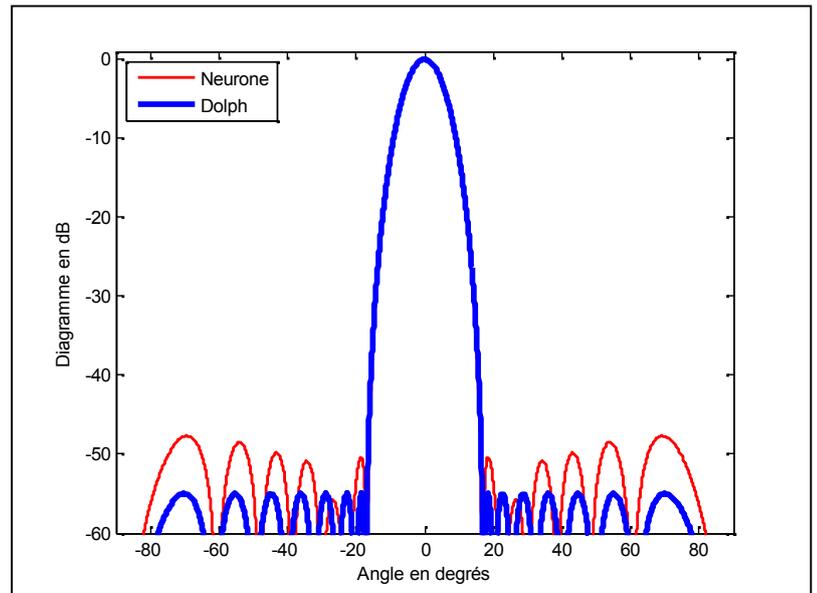


Tableau IV.11 : Pondérations Dolph et RN
R=55dB N=16

Fig IV.17: Test pour R=55dB N=16

IV.5 Simulation des réseaux de neurones avec apprentissage par algorithmes génétiques

Nous avons effectué les mêmes simulations antérieures, mais cette fois-ci pour une synthèse avec la méthode de Schelkunoff expliquée dans le chapitre II.

Les données d'apprentissage sont constituées de la direction de nul imposée et de deux « zéros » obtenus par l'algorithme génétique afin de satisfaire un niveau des lobes secondaires minimaux.

Nous avons représenté, dans les figures IV.18 à IV.20, le résultat de simulation pour les directions de nuls 17,3°, 23,5°, et 33,5°. Les coefficients sont donnés dans les tableaux IV.12 à IV.14.

La position des zéros obtenus par le réseau de neurone (étoiles) par rapport aux zéros obtenus par l'AG sont représentées en figure IV.21.

On voit sur ces figures, la similitude des diagrammes obtenus par l'algorithme AG (bleu) et le réseau de neurones (rouge). On remarque, là aussi que les diagrammes sont presque identiques, ce qui prouve la puissance du réseau de neurone.

Il faut noter un point important : la réponse du réseau de neurones est presque instantanée, alors que celle du réseau génétique est beaucoup plus lente.

Nuls synthétisées par : (Nul=17,3°)		
	AG	RN
1	1.2603	1.2851
2	2.2501	2.2114

Tableau IV.12 : Pondérations AG et RN
Nul = 17,3°N=16

Nuls synthétisées par (Nul=23,5°)		
	AG	RN
1	1.2788	1.4052
2	1.9673	1.9344

Tableau IV.13 : Pondérations AG et RN
Nul = 23,5°N=16

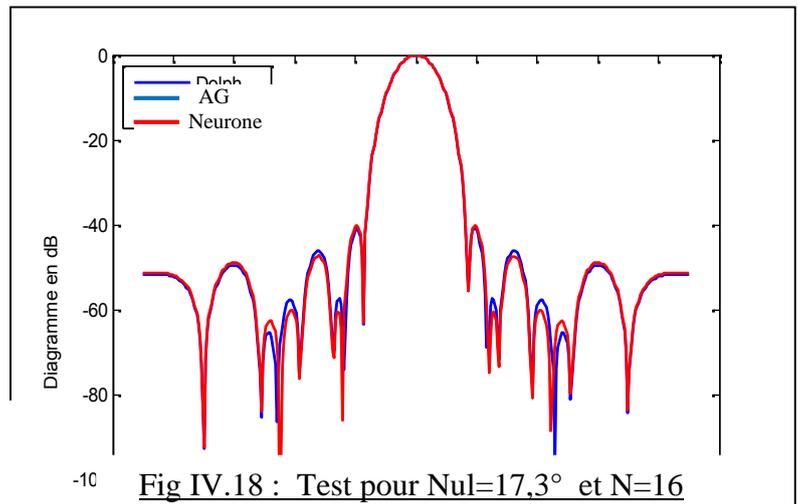


Fig IV.18 : Test pour Nul=17,3° et N=16

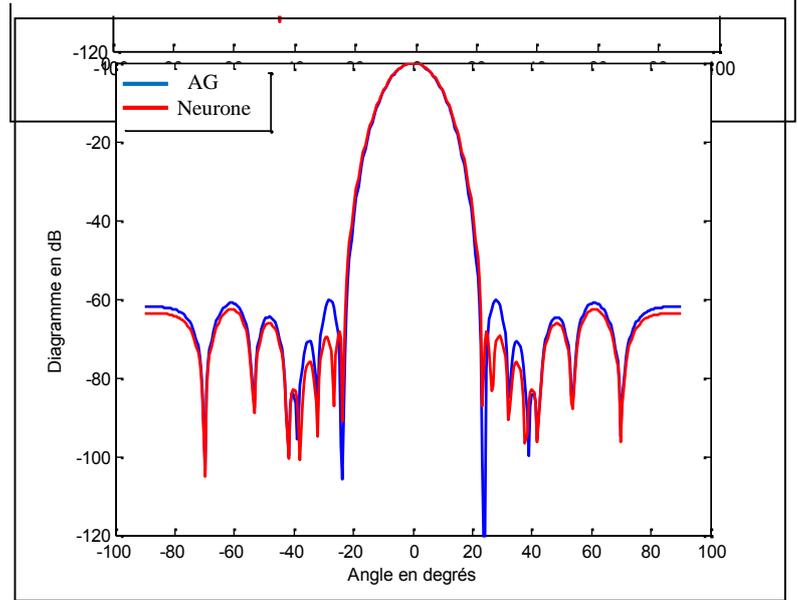


Fig IV.19 : Test pour Nul=23,5° et N=16

Nuls synthétisés par : (Nul=33,5°)		
	AG	RN
1	1.2173	1.1902
2	1.3280	1.2958

Tableau IV.14 : Pondérations AG et RN
Nul = 33,5° N=16

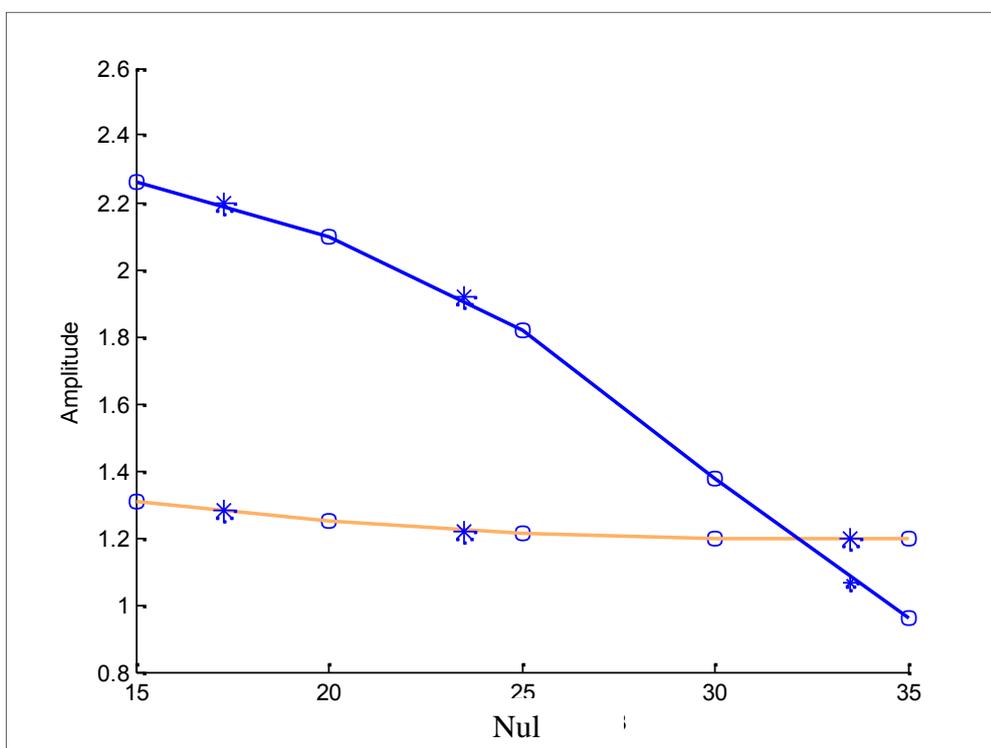
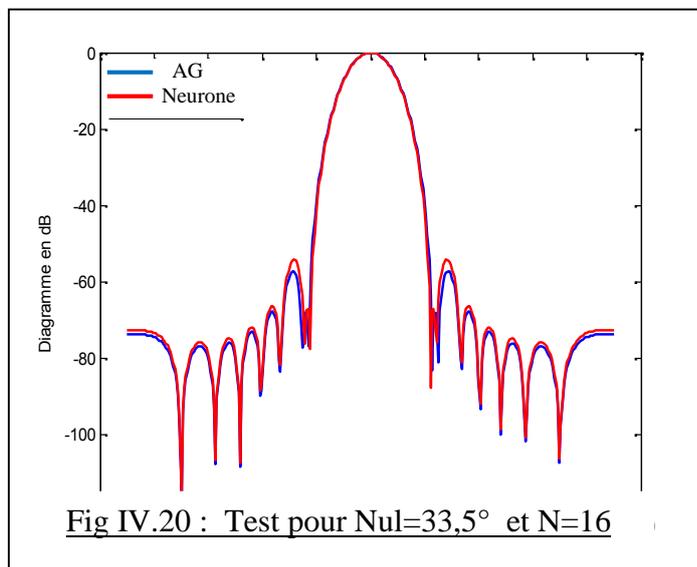


Fig IV. 21 : Test pour Nul=17,5° ,23,5°et 33,5° ; N=16

IV.6 Conclusion

Nous avons montré à travers les différentes simulations que nous avons effectuées, la puissance des réseaux de neurones pour résoudre des problèmes très complexes tel que la synthèse de réseaux d'antennes en calculant les pondérations à appliquer aux courants d'excitation du réseau.

Il faut évidemment disposer de données d'apprentissage en fonction du problème étudié. Le choix de ces données d'apprentissage est crucial pour obtenir de bons résultats. Il faut que leur répartition soit bien étudiée. Lorsque cela est vrai, l'algorithme des réseaux de neurones offre des solutions très proches des autres algorithmes en un temps de calcul presque instantané, ce qui permet d'utiliser sans difficulté les réseaux de neurones en temps réel.

CONCLUSION GENERALE

Dans le cadre de notre formation d'ingénieur, nous avons présenté un mémoire ayant pour thème la synthèse d'un réseau d'antennes par réseau de neurones.

Nous avons utilisé les réseaux de neurones multicouches de type Feedforward, et en particulier, le perceptron multicouche MLP, car ce type est adapté dans notre travail. Nous avons utilisé comme données d'apprentissage, les pondérations obtenues par la méthode Dolph-Tchébycheff dans le cas de la synthèse d'un réseau avec niveau de lobes secondaires fixé, et l'algorithme génétique dans le cas de placement de zéros. Après différents essais, nous avons opté pour l'algorithme d'apprentissage de Levenberg-Marquardt et un réseau à une couche cachée.

Les tests effectués nous ont démontré que les réseaux de neurones constituent des approximateurs universels capables de modéliser des systèmes complexes tels que les réseaux d'antennes avec la précision voulue à partir d'un jeu de données entrées /sorties et un processus d'optimisation.

Leur capacité d'apprentissage permet de simplifier la synthèse d'un diagramme dans la mesure où l'on obtient celui-ci par un simple réglage de paramètres.

L'approche neuronale réduit aussi à l'extrême le temps de calcul lors de la phase d'utilisation ou de généralisation.

Cependant, l'inconvénient des réseaux de neurones est le manque de règle dans le choix de paramètres comme le nombre de couches cachées et le nombre de neurones de ces couches.

Dans des études futures, il serait intéressant d'augmenter le nombre de paramètres d'entrée du réseau de neurones comme par exemple l'ouverture du faisceau, le nombre d'antennes pour avoir un outil de synthèse plus performant.

ANNEXES

A. Fonctions de la boîte à outils du réseau de neurones de Matlab

- New Networks
 - newc Create a competitive layer.
 - newcf Create a cascade-forward backpropagation network.
 - newelm Create an Elman backpropagation network.
 - newff Create a feed-forward backpropagation network.
 - newfftd Create a feed-forward input-delay backprop network.
 - newgrnn Design a generalized regression neural network.
 - newhop Create a Hopfield recurrent network.
 - newlin Create a linear layer.
 - newlind Design a linear layer.
 - newlvq Create a learning vector quantization.
 - networknewp Create a perceptron.
 - newpnn Design a probabilistic neural network.
 - newrb Design a radial basis network.
 - newrbe Design an exact radial basis network.
 - newsom Create a self-organizing map.

- **NEWFF** Create a feed-forward backpropagation network.

Syntaxe : net = newff

net = newff(PR,[S1 S2...SNI],{TF1 TF2...TFNI},BTF,BLF,PF)

Description

NET = NEWFF creates a new network with a dialog box.

NEWFF(PR,[S1 S2...SNI],{TF1 TF2...TFNI},BTF,BLF,PF) takes,

PR R x 2 matrix of min and max values for R input elements Si Size of ith layer, for NI layers

TFi Transfer function of ith layer, default = 'tansig'

BTF Backpropagation network training function, default = 'traingdx'

BLF Backpropagation weight/bias learning function, default = 'learnqdm'

PF Performance function, default = 'mse'

and returns an N layer feed-forward backprop network.

The transfer functions TFi can be any differentiable transfer function such as TANSIG, LOGSIG, or PURELIN.

The training function BTF can be any of the backprop training unctions such as TRAINLM, TRAINBFG, TRAINRP, TRAINGD, etc.

B. Fonction performance

net.performFcn

- mae Mean absolute error-performance function.
- mse Mean squared error-performance function.
- msereg Mean squared error w/reg performance function.
- sse Sum squared error-performance function.

C. Fonction d'apprentissage

net.trainFcn

- Training Functions
- trainbfg BFGS quasi-Newton backpropagation.
- trainbr Bayesian regularization.
- traincgb Powell-Beale conjugate gradient backpropagation.
- traincgf Fletcher-Powell conjugate gradient backpropagation.
- traincgp Polak-Ribiere conjugate gradient backpropagation.
- traingd Gradient descent backpropagation.
- traingda Gradient descent with adaptive lr backpropagation.
- traingdm Gradient descent with momentum backpropagation.
- traingdx Gradient descent with momentum and adaptive lr backprop.
- trainlm Levenberg-Marquardt backpropagation.
- trainoss One-step secant backpropagation.
- trainrp Resilient backpropagation (Rprop).
- trainscg Scaled conjugate gradient backpropagation.
- trainb Batch training with weight and bias learning rules.
- trainc Cyclical order incremental training with learning functions.
- trainr Random order incremental training with learning functions.

TRAIN Train a neural network.

Syntaxe

[net,tr,Y,E,Pf,Af] = train(NET,P,T,Pi,Ai,VV,TV)

Description

TRAIN trains a network NET according to NET.trainFcn and NET.trainParam.

TRAIN(NET,P,T,Pi,Ai) takes,
NET - Network.

P - Network inputs.

T - Network targets, default = zeros.

Pi - Initial input delay conditions, default = zeros.
 Ai - Initial layer delay conditions, default = zeros.
 VV - Structure of validation vectors, default = [].
 TV - Structure of test vectors, default = [].
 and returns,
 NET - New network.
 TR - Training record (epoch and perf).
 Y - Network outputs.
 E - Network errors.
 Pf - Final input delay conditions.
 Af - Final layer delay conditions.

D. Fonction de simulation

SIM Simulate a neural network.

Syntaxe

[Y,Pf,Af,E,perf] = sim(net,P,Pi,Ai,T)
 [Y,Pf,Af,E,perf] = sim(net,{Q TS},Pi,Ai,T)
 [Y,Pf,Af,E,perf] = sim(net,Q,Pi,Ai,T)

Description

SIM simulates neural networks.
 [Y,Pf,Af,E,perf] = SIM(net,P,Pi,Ai,T) takes,
 NET - Network.
 P - Network inputs.
 Pi - Initial input delay conditions, default = zeros.
 Ai - Initial layer delay conditions, default = zeros.
 T - Network targets, default = zeros.
 and returns:
 Y - Network outputs.
 Pf - Final input delay conditions.
 Af - Final layer delay conditions.
 E - Network errors.
 perf - Network performance.
 Note that arguments Pi, Ai, Pf, and Af are optional and need only be used for networks that have input or layer delays.

E. Code généré par l'exécution de la fonction newff

net =

Neural Network object:

architecture:

numInputs: 1
 numLayers: 2
 biasConnect: [1; 1]
 inputConnect: [1; 0]
 layerConnect: [1 0; 1 0]
 outputConnect: [0 1]
 numOutputs: 1 (read-only)
 numInputDelays: 0 (read-only)
 numLayerDelays: 1 (read-only)

subobject structures:

inputs: {1x1 cell} of inputs
 layers: {2x1 cell} of layers
 outputs: {1x2 cell} containing 1 output
 biases: {2x1 cell} containing 2 biases

inputWeights: {2x1 cell} containing 1 input weight
layerWeights: {2x2 cell} containing 2 layer weights

functions:

 adaptFcn: 'trains'
 divideFcn: 'dividerand'
 gradientFcn: 'calcjxfp'
 initFcn: 'initlay'
 performFcn: 'mse'
 plotFcns: {'plotperform','plottrainstate'}
 trainFcn: 'trainlm'

parameters:

 adaptParam: .passes
 divideParam: .trainRatio, .valRatio, .testRatio
 gradientParam: (none)
 initParam: (none)
 performParam: (none)
 trainParam: .show, .showWindow, .showCommandLine, .epochs,
 .time, .goal, .max_fail, .mem_reduc,
 .min_grad, .mu, .mu_dec, .mu_inc,
 .mu_max

weight and bias values:

 IW: {2x1 cell} containing 1 input weight matrix
 LW: {2x2 cell} containing 2 layer weight matrices
 b: {2x1 cell} containing 2 bias vectors

other:

 name: "
 userdata: (user information)

Bibliographie

Thèse de l'étudiant RAHNI Lyes promo 2008 / 2009 «Optimisation d'un réseau d'antennes par l'algorithme génétique», encadré Par Mr. MEGUELATI ismail.

Thèse de poromo 2005/2006 «Etude des techniques de suppression des effets de lobes secondaire avec un réseau d'antenne linéaire», encadré Par Mme Doudou.

Thèse de docteur de l'université de limoge FADLALLAH Najib Année : 2005 «Contribution à l'optimisation de la synthèse du lobe de rayonnement pour une antenne intelligente. Application à la conception de réseaux à déphasage».

Thèse de les étudiants OUNISSI oussama et ALI OTHMANE salim promo juin 2009« commande d'un système dynamique NL (pendule inverse) a l'aide d'un contrôleur AG-neuronal», encadré Par Mr. BENOUARED abdelhalim.

Thèse de les étudiants BOUALBANI hamid et CHABANE mohammed promo2009/2007 «application de réseaux de neurones à l'estimation de la vitesse d'un moteur à synchrone », encadré Par Mr. FERDJOUNI. A .

Thèse de FADLALLAH Najib, RAMMAL mohammed, GHAYOULA ridha et GHARSALLAH ali setit 2007 « synthèse de diagramme de rayonnement de réseau d'antenne linéaires par la méthode de Dolph -Tchebycheff».

Thèse de Debbat.F et Bendimerad.F.T setit 2005 « les algorithmes d'optimisation globale : application réseau intelligents d'antennes».

Thèse de Mr. Ouahib GUENOUNOU le 22 avril 2009 « Méthodologie de conception de contrôleurs intelligents par l'approche génétique- application à un bioprocédé».
