

# Optimisation pour la formation de faisceaux d'un Réseau Planaire d'Antennes Adaptatives par Réseaux de Neurones Artificiels

Georges Lemuel BALOGOG MANDACK

École Normale Supérieure d'Enseignement Technique (ENSET)  
Université de Douala, Cameroun

Juin 2019

## Résumé

La propagation en espace libre fait face aux problèmes de perturbations dus à un écosystème fortement congestionné. D'où la nécessité de concevoir des antennes intelligentes capables de s'adapter au milieu dans lequel elles se trouvent afin de maintenir les performances du système de communication, voire les améliorer. Dans un premier temps, il s'agit d'implémenter un algorithme de détermination des directions d'arrivée, puis de formation de faisceaux suivant la direction souhaitée. Ce travail présente les résultats d'optimisation en temps de calcul et de directivité du lobe de pointage (lobe principal) par une méthode heuristique : les réseaux de neurones artificiels (RNA) pour un réseau planaire d'antennes monopôle. Le critère de synthèse est la minimisation de l'erreur quadratique moyenne obtenue par la méthode de Dolph-Tchebychev et celle des RNA, notamment le perceptron multicouche (PMC). Les résultats obtenus sont pertinents et méritent d'être comparés par la suite aux autres méthodes heuristiques.

**Mots-clés** *Antennes intelligentes, Détermination des directions d'arrivée, Formation de faisceaux, Heuristique, RNA, Dolph-Tchebychev, PMC.*

## 1 Introduction

Les rapides développements de la recherche et de l'industrialisation ont permis à un large public d'accéder aux moyens modernes de communication. Les communications sans fil connaissent une évolution rapide avec une vitesse exponentielle ; mais le revers de la médaille est une pollution électromagnétique croissante. La transmission fiable des données revêt alors un enjeu plus que majeur ; ceci impose une maîtrise des techniques de plus en plus poussées au risque de saturation du spectre radio.

Jusqu'à présent, les techniques employées pour accroître la capacité du système étaient entre autres l'accès multiple et le découpage cellulaire [1]. Pour répondre à l'explosion des applications multimédias, la prochaine gé-

nération des réseaux de communication sans fil devra être en mesure de transmettre des signaux à très haut débit, avec différentes qualités de service, tout en permettant un déploiement rapide et une forte pénétration du marché. D'où la nécessité d'améliorer les performances des systèmes et d'augmenter leurs capacités spectrales.

Par ailleurs, pour faire face à l'augmentation imprévisible du nombre d'utilisateurs d'une part et à l'augmentation des débits de transmission d'autre part, les futurs réseaux de communications devront mettre en œuvre des techniques de plus en plus évoluées. Plusieurs approches sont possibles et l'une d'entre elles consiste à combiner les signaux reçus par les éléments d'une antenne réseau : ce sont les **antennes intelligentes**.

Un des principaux avantages de ces systèmes réside dans leur capacité à réagir automatiquement à un environnement complexe dont l'interférence est connue ou détectée. Ces systèmes reposent sur des antennes réseau [2, 3]. Des dispositifs pour calculer les angles d'arrivée et des outils numériques de synthèse permettent d'attribuer des poids (ou niveaux d'importance) aux éléments de l'antenne réseau afin d'optimiser le signal de sortie selon des techniques de contrôle prédéfinies pour la formation des voies et l'annulation d'interférents [4, 5].

### 1.1 Objectifs

L'objectif de notre étude est d'élaborer une technique numérique de synthèse pour la formation de faisceaux qui répondent aux spécifications imposées par le système adaptatif appliqué à un réseau planaire d'antennes monopôle, du fait de leur coût relativement bas. Une contrainte de temps de calcul nous étant imposée, nous avons développé un outil numérique basé sur les réseaux de neurones afin d'augmenter la vitesse de convergence.

### 1.2 Organisation du document

La Section 2 présente les méthodes de synthèse utilisées : celle de Dolph-Tchebychev et celle des RNA. Par

la suite, à la Section 3, les résultats obtenus sont présentés et discutés. Nous terminons par une conclusion à la Section 4.

## 2 Méthodologie

Avant toute chose, il est important de comprendre le mécanisme des réseaux d'antennes planaires.

### 2.1 Structure Planaire et Outils Logiciels

Une antenne intelligente planaire est vue comme un ensemble d'antennes élémentaires disposées dans le plan, pilotées par un processeur numérique de signaux (DSP) intégrant des algorithmes adaptatifs permettant d'adapter le pointage du diagramme de rayonnement en temps réel (Fig. 1b).

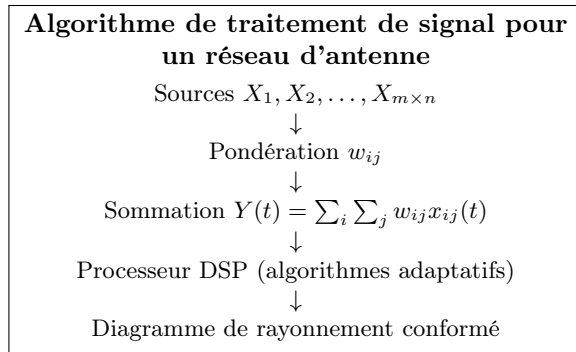
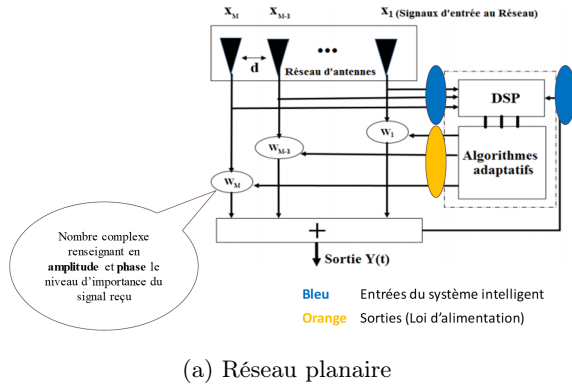


FIGURE 1 – Architecture et principe de fonctionnement d'une antenne intelligente planaire

Le signal en réception ou en émission est exprimé par :

$$Y(t) = \sum_i^m \sum_j^n w_{ij} x_{ij}(t) \quad (1)$$

où  $m \times n$  correspond à la matrice formée par le plan constitué des antennes en réseau respectivement sur les axes  $x$  et  $y$ .

Le diagramme de rayonnement des éléments pris individuellement sont additionnés. Les coefficients  $w_{ij}$  expriment le niveau d'importance accordé au signal provenant de l'élément rayonnant élémentaire  $ij$ . Ainsi, pour des valeurs de  $w_{ij}$  variant dans le temps, selon un formalisme bien défini, le système ainsi constitué est un réseau adaptatif capable de conformer le diagramme de rayonnement. Ledit formalisme est basé sur un calcul préalable de la direction d'arrivée (DOA, *Direction of Arrival*) ce qui est réalisable grâce à un processeur de traitement de signal (DSP).

### 2.2 Environnement Logiciel

Pour implémenter l'algorithme de formation de faisceaux (*beamforming*) construit autour d'un RNA de type perceptron multicouche, le logiciel retenu a été **MATLAB R2018a** du fait qu'il intègre la boîte à outils NNSTART [6] qui propose les fonctions appliquées pour les problèmes de régression dont fait partie le problème de la formation de faisceaux.

L'outil MATLAB est en effet construit autour de boîtes à outils logicielles utilisées dans de nombreux problèmes d'ingénierie et facilement accessible par la communauté étudiante dotée d'outils informatiques.

Pour apprécier les performances électriques et magnétiques du réseau construit, les simulations sont faites sur le logiciel **CST Microwave Studio** [7] car celui-ci, grâce à ses solveurs numériques, offre des résultats les plus proches du réel. L'enjeu est donc de lier logiciellement MATLAB et CST Microwave Studio afin de piloter le réseau d'antennes conçu sur CST à partir de données issues du RNA sur MATLAB (voir schéma Fig. 4).

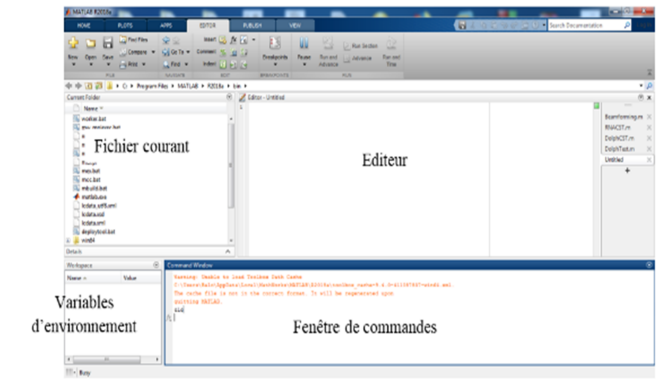


FIGURE 2 – Matlab

### 2.3 Méthode de Dolph-Tchebychev

Cette méthode est basée sur les propriétés des polynômes de Tchebychev, qui permettent d'obtenir le maximum de gain pour un niveau de lobes secondaires imposé. Cette méthode utilise le fait que la répartition optimale des amplitudes des sources est celle qui donne, pour

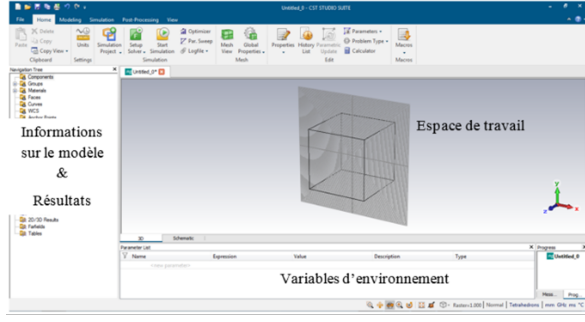


FIGURE 3 – CST



FIGURE 4 – Chaîne de co-simulation MATLAB–CST

l'expression du champ rayonné par un alignement de  $N$  sources, les polynômes de Tchebychev de degré  $(N - 1)$ .

Ce polynôme présente toujours un maximum de niveau important qui correspond au maximum du lobe principal du diagramme de rayonnement, et une succession de maxima et de minima d'amplitudes égales, qui correspondent ici aux lobes secondaires [1]. Avec cette méthode, tous les lobes secondaires du diagramme ont un même niveau, et présentent toujours le maximum de gain compatible avec le niveau de lobes secondaires que l'on s'est imposé [1].

Pratiquement, le calcul de la distribution d'amplitude se fait par un changement d'échelle de façon à traduire la condition d'optimisation introduite en fixant le paramètre  $R_0$  (rapport entre le niveau du lobe principal et celui des lobes secondaires) tel que :

$$R_0 = T_{N-1}(x_0) = \cosh[(N - 1) \cdot \operatorname{arccosh}(x_0)] \quad (2)$$

avec

$$x_0 = \cosh \left[ \frac{\operatorname{arccosh}(R_0)}{N - 1} \right] \quad (3)$$

## 2.4 RNA : Perceptron Multicouche

Parmi les types de réseaux de neurones les plus utilisés, on trouve le MLP (*MultiLayer Perceptron*) avec son algorithme d'apprentissage, la rétropropagation des erreurs. Le perceptron multicouche est un réseau orienté de neurones artificiels en couches, où l'information voyage dans un seul sens, de la couche d'entrée vers la couche de sortie.

La couche d'entrée présente toujours une couche virtuelle associée aux entrées du système. Le RNA-PMC (Fig. 5) utilisé comporte :



FIGURE 5 – Réseau de neurone sur Matlab

- **Deux (2) neurones en entrée** : valeurs angulaires d'élévation  $\theta$  et d'azimut  $\phi$
- **Trois (3) couches cachées** avec fonction d'activation sigmoïde
- **Quarante (40) neurones en sortie** : valeurs d'amplitudes ( $A_i$ ) et phases ( $\phi_i$ ) des 20 éléments du réseau d'antennes

### 2.4.1 Mise en Œuvre des Réseaux PMC

La mise en œuvre des réseaux de neurones comporte à la fois une partie conception, dont l'objectif est de permettre de choisir la meilleure architecture possible, et une partie de calcul numérique, pour réaliser l'apprentissage d'un réseau de neurones.

Dans le cas général, un MLP peut posséder un nombre de couches quelconque et un nombre de neurones par couche également quelconque ; mais en vue de perfectionner le fonctionnement du MLP d'un côté et minimiser au maximum le temps de calcul d'autre part, il est recommandé de chercher une architecture optimale du point de vue nombre de couches et nombre de neurones par couche.

À partir d'une architecture de réseau de neurones donnée et des exemples disponibles (la base d'apprentissage), les poids optimaux sont déterminés par l'algorithme de la rétropropagation des erreurs, pour que la sortie du modèle s'approche le plus possible du fonctionnement désiré.

### 2.4.2 Apprentissage des Réseaux PMC

L'apprentissage neuronal fait appel à des exemples de comportement. Soit une base d'apprentissage constituée de  $N$  exemples, chacun étant constitué d'un vecteur  $\mathbf{x}(n)$  appliqué aux entrées du réseau, du vecteur  $\mathbf{d}(n)$  des valeurs désirées correspondantes pour les sorties, et du vecteur  $\mathbf{y}(n)$  correspondant à la sortie du réseau pour l'entrée  $\mathbf{x}(n)$ .

On suppose aussi que le réseau de neurones possède un nombre  $r$  de neurones de sortie. L'apprentissage d'un réseau de neurones est défini comme un problème d'optimisation qui consiste à trouver les coefficients du réseau minimisant une fonction d'erreur globale (fonction coût).

La définition de cette fonction de coût est primordiale, car celle-ci sert à mesurer l'erreur entre les sorties désirées du modèle et les sorties du réseau observées. La fonction dite **fonction d'erreur quadratique** est définie pour chaque échantillon (exemple d'apprentissage)  $n$  ( $n \in \mathbb{N}$ ) :

$$e_j = \frac{1}{2} \sum_{j=1}^r [d_j(n) - y_j(n)]^2 \quad (4)$$

Pour tout l'ensemble d'apprentissage  $N$ , on peut définir la fonction de coût (appelée aussi l'**erreur quadratique moyenne** EQM ou MSE, *Mean Squared Error*) :

$$E(n) = \frac{1}{N} \sum_{n=1}^N \left\{ \frac{1}{2} \sum_{j=1}^r [d_j(n) - y_j(n)]^2 \right\} \quad (5)$$

soit :

$$E(n) = \frac{1}{N} \sum_{n=1}^N e(n) \quad (6)$$

Le principe de l'algorithme d'apprentissage (la rétropropagation) est de calculer la contribution des poids du réseau à cette erreur.

### 2.4.3 Algorithme de Rétropropagation de l'Erreur

L'apprentissage du MLP est attaché à l'algorithme de la rétropropagation des erreurs. Cet algorithme consiste simplement en une descente de gradient, qui est une méthode d'optimisation universelle. On cherche à minimiser une fonction de coût (qui représente l'erreur entre la sortie désirée et la sortie obtenue), en suivant les lignes de plus grande pente.

La mise en œuvre de cet algorithme nécessite un enchaînement des opérations mathématiques données comme suit. Soit le couple  $(\mathbf{x}(n), \mathbf{d}(n))$  désignant la  $n$ -ième donnée d'apprentissage du réseau tel que :

$$\mathbf{x}(n) = (x_1(n), \dots, x_p(n)) \quad \text{entrées } 1, 2, \dots, n \quad (7)$$

$$\mathbf{d}(n) = (d_1(n), \dots, d_p(n)) \quad \text{sorties } 1, 2, \dots, n \quad (8)$$

L'algorithme de la rétropropagation consiste alors à mesurer l'erreur entre les sorties désirées  $\mathbf{d}_n$  et les sorties observées (prédites par le réseau de neurones)  $\mathbf{y}_n$ .

### 2.4.4 Mise en Œuvre de l'Algorithme

**Étape 1** : Initialisation de tous les poids à de petites valeurs aléatoires dans l'intervalle  $[-1, 1]$ .

**Étape 2** : Pour chaque donnée d'entraînement  $n$  :

- Calculer les sorties observées en propageant les entrées vers l'avant (*forward propagation*).
- Ajuster les poids en rétropropageant l'erreur observée (*backward propagation*) :

$$w_{ij}(n) = w_{ij}(n-1) + \Delta w_{ij}(n) \quad (9)$$

$$w_{ij}(n) = w_{ij}(n-1) + \eta \cdot \delta_j(n) \cdot y_i(n) \quad (10)$$

où  $\eta$  est le taux d'apprentissage tel que  $\eta \in [0, 1]$ .

**Étape 3** : Répéter les deux étapes précédentes jusqu'à un nombre maximum d'itérations ou jusqu'à ce que la valeur de l'erreur quadratique moyenne (EQM) soit inférieure à un certain seuil.

En effet, attendre que l'EQM soit inférieure à un seuil n'est pas sûr ; alors pour éviter le problème de la boucle ouverte (divergence), on fixe un nombre d'itérations maximum, généralement de l'ordre de plusieurs centaines, et dans ce cas, l'algorithme cherche à minimiser l'EQM en  $N_i$  itérations successives tel que  $N_i$  est inférieur au nombre d'itérations maximum.

La Figure 6 présente l'organigramme de l'algorithme de rétropropagation.

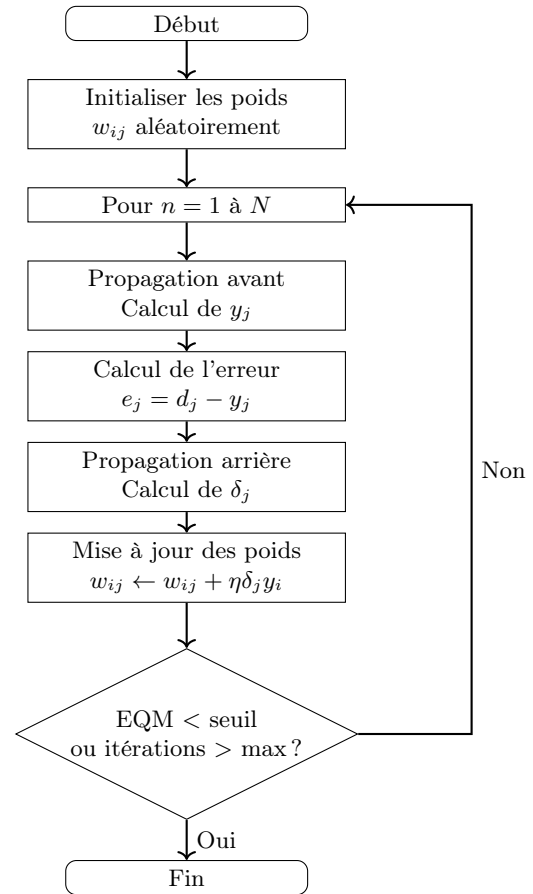


FIGURE 6 – Organigramme de l'algorithme de rétropropagation de l'erreur

## 3 Résultats et Discussion

Tout notre travail repose sur la modélisation d'une antenne réseau bâtie autour d'une structure élémentaire de type monopôle. Notre choix s'est porté sur une antenne évoluant dans la bande de fréquences de 2,4 GHz, très utilisée dans les systèmes de télécommunications.

### 3.1 Monopôle Élémentaire

Le modèle obtenu est simulé sur le logiciel CST Microwave Studio. Le récapitulatif des dimensions de notre antenne est donné dans le Tableau 1. Ce modèle élémentaire constitue le socle de notre réseau car il fournit les performances approximativement correctes et satisfaisantes pour notre cahier des charges. Celles-ci sont obtenues après simulation et résolution des équations liées au magnétisme en champs lointains.

TABLE 1 – Dimensions du monopôle élémentaire

Paramètre	Valeur
Hauteur du brin vertical	88 mm ( $\lambda/4$ )
Rayon du fil	0,691 mm
Plan de masse	$50 \times 50 \text{ mm}^2$
Fréquence de résonance	2,4264 GHz
Épaisseur plan de masse	0.005 mm
Ecartement filament masse	1 mm
Impédance d'entrée	$50 \Omega$
Niveau lobes secondaires	-4,4 dB

Les Figures 8 et 9 indiquent que l'antenne ainsi conçue rayonne avec un niveau de lobes secondaires de -4,4 dB. De plus, elle rayonne à la fréquence de 2,4264 GHz, fréquence acceptable pour une communication en bande Wi-Fi.

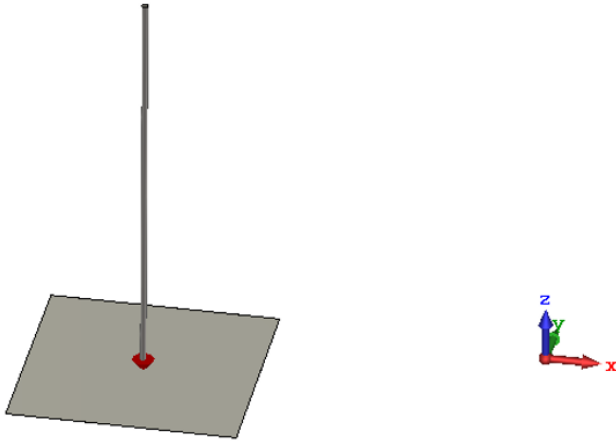


FIGURE 7 – Monopole (Vue 3D)

La Figure 10 présente le paramètre  $S_{11}$  du monopôle élémentaire, confirmant la résonance à 2,4 GHz.

### 3.2 Réseau Planaire 4×5 Monopôles

L'élément élémentaire conçu, il est possible de concevoir le réseau d'antennes planaire (Fig. 11)  $4 \times 5$  monopôles avec un espacement inter-éléments de  $d_x = d_y = 0,4\lambda$ .

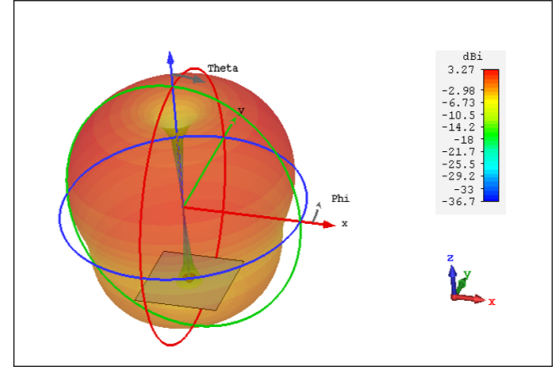


FIGURE 8 – Diagramme de rayonnement 3D du monopôle élémentaire

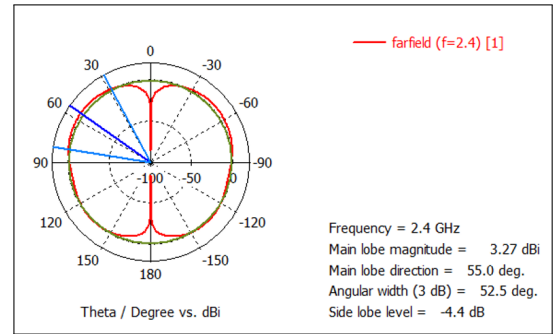


FIGURE 9 – Diagramme de rayonnement du monopôle élémentaire en coordonnées polaires

### 3.3 Loi d'Alimentation par RNA

Le Tableau de la Fig. 12 présente les résultats du calcul des poids pour un monofaisceau pointant dans la direction  $\theta = 330^\circ$  et  $\phi = 30^\circ$ .

### 3.4 Diagrammes de Rayonnement

La Figure 13 donne la représentation 2D du diagramme de rayonnement monofaisceau obtenu par la méthode des réseaux de neurones à partir des valeurs des coefficients amplitudes et phases présentées dans le Tableau de la Fig 12.

La Figure 15 présente les niveaux de puissance du faisceau obtenu pour le réseau planaire en dB, dans le plan de coupe  $\phi = 90$ . Ces courbes donnent un maximum de puissance du lobe principal dans les directions théoriques  $\theta = 330$  et  $\phi = 30$ . Par contre, les résultats en simulation (Fig. 15) montrent un maximum de puissance dans la direction  $\phi = 35,034$  avec un niveau de lobes secondaires de -10,6 dB.

La Figure 16 donne la représentation 3D du diagramme de rayonnement ainsi obtenu.

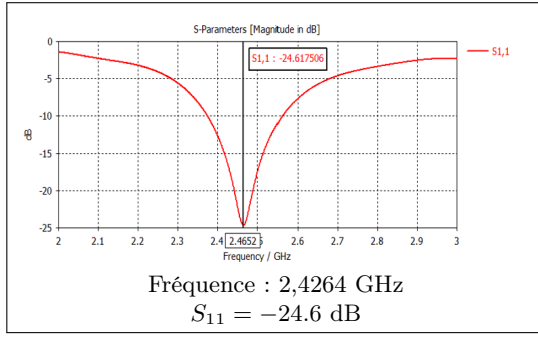


FIGURE 10 – Coefficient de réflexion du monopôle élémentaire

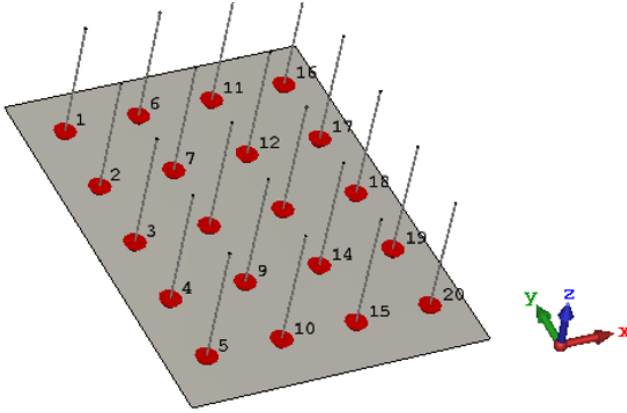


FIGURE 11 – Réseau planaires de monopôles 4\*5 vue en 3D

### 3.5 Analyse Comparative

Les résultats parlent d'eux-mêmes et on constate les avantages que revêtent les systèmes adaptatifs (Figure 17) basés sur les RNA, notamment l'amélioration du paramètre de directivité qui est de 0,2 dB par rapport à la méthode analytique de Dolph-Tchebychev.

Toutefois, ce type de système présente un comportement indéterministe et non contrôlable pour une zone hors couverture ( $270 < \theta < 380$  et  $0 < \phi < 55$ ), auquel s'additionne la difficulté d'élaboration d'un RNA par absence de formalisme optimal établi.

### 3.6 Discussion

Ainsi, le RNA présente un axe d'amélioration. En effet, l'apprentissage du RNA n'est possible que grâce à une base de données enrichie par des mesures obtenues analytiquement par la méthode de Tchebychev. Cette base de données pose une limite dans le formalisme des polynômes du fait de la restriction des données en pairs ou impairs en termes du nombre d'éléments du réseau d'antenne.

Il est donc possible d'utiliser des données obtenues par des mesures en conditions réelles ou même de modifier

Poids	Angle $\theta$	Angle $\varphi$
	330	30
	Amplitude	Phase
1.1	0,0187973920146745	178,707658144959
1.2	0,0478557097772688	142,707658144959
1.3	0,0390955772661847	106,707658144959
1.4	0,0478557097772687	70,7076581449593
2.1	0,0187973920146745	116,353829072480
2.2	0,0494484555534090	80,3538290724797
2.3	0,0771363354757668	44,3538290724796
2.4	0,101296656064291	8,35382907247962
3.1	0,0771363354757668	54,0000000000001
3.2	0,0494484555534090	18,0000000000001
3.3	0,0494484555534090	-18
3.4	0,0771363354757668	-54,0000000000000
4.1	0,101296656064291	-8,35382907247958
4.2	0,0771363354757668	-44,3538290724796
4.3	0,0494484555534090	-80,3538290724796
4.4	0,0187973920146745	-116,353829072480
5.1	0,0478557097772687	-70,7076581449593
5.2	0,0390955772661847	-106,707658144959
5.3	0,0478557097772687	-142,707658144959
5.4	0,0187973920146745	-178,707658144959

FIGURE 12 – Amplitude et phase obtenues par la méthode des réseaux de neurones

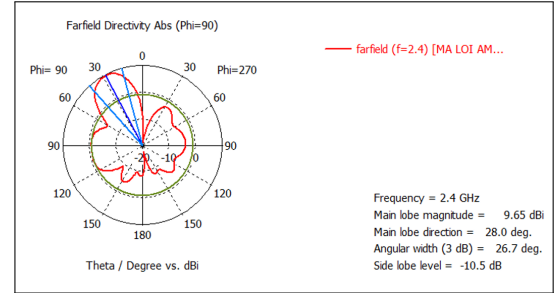


FIGURE 13 – Diagramme de rayonnement 2D obtenu par le réseau de neurones

la technique d'apprentissage en utilisant des méthodes adaptatives telles que le LMS appliqué dans [4] et qui présente une meilleure robustesse pour un réseau d'antennes patch. On note également une augmentation du balayage lors de l'incrément du nombre d'éléments du réseau, mais aussi une augmentation du nombre de lobes secondaires.

## 4 Conclusion

Ce travail nous a permis d'améliorer les performances des réseaux planaires monopôles par une prédiction de loi d'alimentation par réseau de neurones. Toutefois, ce travail est une étape en plus conduisant à la réalisation de solutions matérielles sur puces (SoC ou SiP).

Le réseau de neurones élaboré est un PMC bâti autour de trois (3) couches cachées pour deux (2) entrées (azimut et élévation) et quarante (40) sorties correspondant au couple amplitude-phase de chaque élément du réseau. Les performances en temps des capacités de prédiction pour une fréquence de résonance de 2,4 GHz du système sont évaluées par la fonction `tic-toc` de MAT-



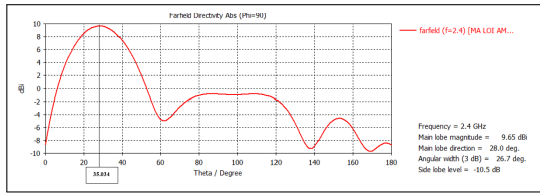


FIGURE 14 – Niveaux de puissance des lobes pour les directions  $\theta = 330$  et  $\phi = 30$

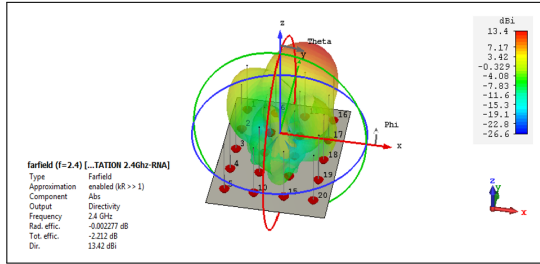


FIGURE 15 – Niveaux de puissance des lobes pour les directions  $\theta = 330$  et  $\phi = 30$

LAB, ce qui a permis d'atteindre un temps de calcul de **1,273532 seconde**. Ceci reste acceptable pour une utilisation en bande WiFi.

Toutefois, le temps d'apprentissage du réseau reste très élevé pour l'algorithme de rétropropagation, évalué à **27 heures 59 minutes 58 secondes**.

#### 4.1 Perspectives

Dans de prochains travaux, nous essayerons d'améliorer :

- Contrôler la formation des faisceaux en contrôlant les effets d'auto-couplage pour réduire le nombre d'alimentation
- L'angle d'ouverture (angle solide) qui reste encore relativement grand jusqu'ici
- L'angle maximal de balayage
- La réjection des interférences par le placement des zéros

Ceci en modifiant les données d'apprentissage : données d'apprentissage prélevées en conditions réelles ou en utilisant une méthode heuristique alternative.

On notera aussi l'importance du nombre d'éléments rayonnants, mais aussi le caractère heuristique de la méthode des réseaux de neurones (optimums locaux), autant d'aspects constituant des axes de recherche.

## Références

- [1] D. Grenier, *Antennes et Propagation Radio*, 2019.
- [2] O. Picon et al., *Les Antennes : Théorie, Conception et Application*, Dunod, 2009.
- [3] L. C. Godara, *Smart Antenna*, CRC Press, 2004.

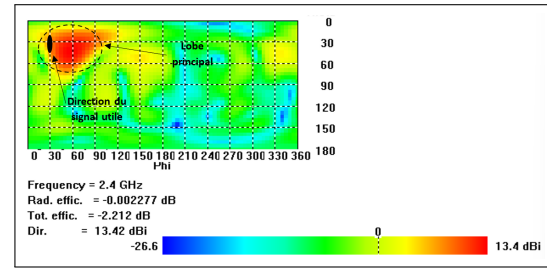


FIGURE 16 – Niveaux de puissance des lobes pour les directions  $\theta = 330$  et  $\phi = 30$

Caractéristiques du faisceau (Diagramme de rayonnement)	Résultats par Méthode Analytique (Polynôme de Dolph Tchebychev)	Résultats par Méthode de Réseaux de neurone
Angle d'arrive	$\varphi = 30^\circ$	$\varphi = 30^\circ$
	$\vartheta = 330^\circ$	$\vartheta = 330^\circ$
Angle de pointage du faisceau	$\varphi = 34,044^\circ$	$\varphi = 35,034^\circ$
	$\vartheta = 330^\circ$	$\vartheta = 330^\circ$
Directivité	$D_{dB} = 12.89$ dB	$D_{dB} = 12.96$ dB
Niveau des lobes secondaires	$SLL_{dB} = -11.70$ dB	$SLL_{dB} = -10.6$ dB

FIGURE 17 – Tableau récapitulatif des résultats obtenus par la méthode de Dolph T. et la méthode des réseaux de neurones

- [4] E. Tonye and S. Kepchabe, "Simulation des performances des antennes intelligentes par approches analytiques," *Vandana Publications IJEMR*, pp. 320–331, août 2015.
- [5] E. Tonye and S. Kepchabe, "Optimisation des performances des antennes intelligentes par synthèse au moyen des algorithmes génétiques," *Sciences, Technologies et Développement*, Édition spéciale, pp. 172–179, juillet 2016.
- [6] H. B. Demuth, M. H. Beale, and M. T. Hagan, *Neural Network Toolbox 7 User's Guide*, MathWorks, 2010.
- [7] CST, *CST Studio Suite 2006B Application Note*, 2006.