

Titre: Repérage de la direction d'arrivée d'un vaisseau par rapport à un réseau d'antennes disposées sur deux axes et à l'aide d'un réseau de neurones
Title:

Auteur: Michael Coudyser
Author:

Date: 2003

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Coudyser, M. (2003). Repérage de la direction d'arrivée d'un vaisseau par rapport à un réseau d'antennes disposées sur deux axes et à l'aide d'un réseau de neurones [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/7128/>

Document en libre accès dans PolyPublie

Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/7128/>
PolyPublie URL:

Directeurs de recherche: Jean-Jules Brault, & Jean-Jacques Laurin
Advisors:

Programme: Non spécifié
Program:

**In compliance with the
Canadian Privacy Legislation
some supporting forms
may have been removed from
this dissertation.**

**While these forms may be included
in the document page count,
their removal does not represent
any loss of content from the dissertation.**

UNIVERSITÉ DE MONTRÉAL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

REPÉRAGE DE LA DIRECTION D'ARRIVÉE D'UN FAISCEAU
PAR RAPPORT À UN RÉSEAU D'ANTENNES DISPOSÉES SUR DEUX AXES
ET À L'AIDE D'UN RÉSEAU DE NEURONES

MICHAEL COUDYSER
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES (M.Sc.A.)
(GÉNIE ÉLECTRIQUE)

AVRIL 2003



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-86404-9

Our file Notre référence

ISBN: 0-612-86404-9

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Canadä

UNIVERSITÉ DE MONTRÉAL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

REPÉRAGE DE LA DIRECTION D'ARRIVÉE D'UN FAISCEAU
PAR RAPPORT À UN RÉSEAU D'ANTENNES DISPOSÉES SUR DEUX AXES
ET À L'AIDE D'UN RÉSEAU DE NEURONES

présenté par : Michael Coudyser

en vue de l'obtention du diplôme de : Maîtrise ès Sciences Appliquées
a été dûment accepté par le jury d'examen constitué de :

M. Cevdet AKYEL, Ph.D., président

M. Jean-Jules BRAULT, Ph.D., membre et directeur de recherche

M. Jean-Jacques LAURIN, Ph.D., membre et co-directeur de recherche

M. Gilles PESANT, Ph.D., membre

*À Marie-Camille, ma grand-mère
À Jean-Louis et Marthe, mes parents*

REMERCIEMENTS

Je voudrais exprimer mes remerciements à mon directeur de recherche, le professeur Jean-Jules Brault, pour sa disponibilité, son support, pour la confiance qu'il m'a accordée mais également pour l'originalité de ses commentaires et sa bonne humeur durant mes deux ans passés à Montréal.

L'expression de ma reconnaissance va également à l'endroit de mon co-directeur de recherche, le professeur Jean-Jacques Laurin, pour sa disponibilité, sa compétence, ses conseils toujours pertinents, tout au long de ce projet. Mes remerciements vont ensuite à tous les membres du projet, Jean-Frédéric Gagné et Nelson Fronseca qui ont toujours été très disponibles pour m'aider si nécessaire.

Enfin, si ce mémoire a pu aboutir, c'est aussi grâce au soutien inconditionnel de ma famille, de mon frère et de mes amis, de Corse ou d'ailleurs, à qui je désire exprimer ma gratitude pour leurs encouragements et leur support infaillible.

RÉSUMÉ

Le présent mémoire présente une étude qui a pour objectif de créer un système de repérage de la direction d'arrivée d'un faisceau pour des angles d'arrivée contenus dans un cône de demi-angle au sommet de 45° . Ce système original et simple à la fois a été conçu pour pouvoir être embarqué sur un Micro satellite de l'Agence Spatiale Canadienne et offre une réponse avec une erreur de moins de 4° . Le système peut se décomposer en trois grandes parties : un réseau d'antennes qui capte le faisceau incident, un combinateur radio-fréquence qui combine les puissances reçues par les différentes antennes et un réseau de neurones qui retrouve la direction d'arrivée du faisceau à partir des signaux sortant du combinateur.

Le travail qui est présenté explique comment nous avons optimisé l'architecture des réseaux d'antennes et du combinateur afin de créer des signaux qui peuvent être interprétés avec le moins d'ambiguïté possible par le réseau de neurones. Néanmoins, le cœur du travail réside dans la conception et l'optimisation des réseaux de neurones.

Notre méthodologie a toujours été conditionnée par la recherche d'un système simple et performant. Pour cela nous avons employé des approches aussi bien qualitative qu'expérimentale que théorique tout au long de la conception. En effet c'est par des critères qualitatifs que le réseaux d'antenne a finalement été conçu, c'est une étude expérimentale comparée qui nous a permis d'élaborer différents filtres pour augmenter le rapport signal sur bruit des signaux et enfin ce sont des concepts théoriques que nous avons mis en œuvre pour optimiser les réseaux de neurones.

Les principales contributions de notre travail sont les suivantes.

- Nous avons proposé une méthode audacieuse mais performante pour la création de l'architecture du réseau d'antennes.
- Nous avons défini les critères qu'il faut prendre en compte pour choisir et réaliser un filtre qui augmente le rapport signal sur bruit.

- Nous avons optimisé et entraîner des réseaux de neurones pour donner l'angle d'arrivée du signal et s'assurer que la direction d'arrivée était réellement dans le cône d'étude
- Enfin nous avons validé le système à partir de données expérimentales provenant de notre réseau d'antennes.

Les résultats obtenus confirment qu'un système constitué seulement de quatre antennes peut détecter la direction d'arrivée à moins de 4° près, pour des valeurs de rapport signal sur bruit supérieures à 9dB.

ABSTRACT

This thesis presents a study that aims to create a system to locate the direction of arrival of a beam for angles of arrival contained in a cone whose vertex angle is 90° . We have conceived this original and simple system that gives a response with an error of less than 4° in order to embed it on a Micro satellite of the Canadian Space Agency. This system can be divided into three parts: an antenna array which receives the incidental beam, a radio frequency combiner which combines the powers received by the various antennas and a neural network which finds out the direction of arrival of the beam from the outputs of the combiner.

This report explains how we have optimized both the architecture of the antenna array and the characteristics of the combiner in order to create signals which can be interpreted by the neural network in the best conditions. Nevertheless, the core of this work resides in the design and the optimization of the neural networks.

Our methodology was always leaded by the curse of simplicity as well as performance. For that, we have used several approaches as qualitative, experimental or theoretical all along the study. Indeed, an antenna array was finally designed under qualitative criteria. Several experimentations revealed differences between filters increasing the signal to noise ratio and finally our neural networks optimization followed theoretical concepts. The principal contributions of our work are as follows.

- We proposed a daring method but still efficient to create the architecture of the antenna array.
- We defined the criteria to take into account to choose and create a filter which increases the signal to noise ratio.
- We have optimized and trained neural networks to give the angle of arrival and to assure that the signal was really in the cone of study
- We tested the system under real conditions with experimental dataset coming from our real antenna array.

The results obtained confirm that a system made up of only four antennas can determine the direction of arrival with an error under 4° for a signal to noise ratio above 9 dB.

TABLES DES MATIÈRES

REMERCIEMENTS	v
RÉSUMÉ	VI
ABSTRACT	VIII
LISTE DES TABLEAUX	XIII
LISTE DES FIGURES	XIV
LISTE DES SIGLES ET ABRÉVIATIONS	XIX
CHAPITRE I INTRODUCTION	1
1.1 Principe de l'estimation de la direction d'arrivée.....	1
1.2 Réseaux d'antennes.....	2
1.3 Réseaux de neurones.....	6
1.3.1 Définition des trois problèmes d'apprentissage.....	6
1.3.2 Neurones, fonction d'activation, couches	9
1.3.3 Les étapes de la conception d'un réseau.....	13

CHAPITRE II REVUE DE LITTERATURE..... 16

2.1 Différentes architectures de réseaux d'antennes	17
2.1.1 Notion de couplage inter-antennes	17
2.1.2 Architectures classiques.....	18
2.1.3 Architectures optimisées.....	20
2.2 Différents choix de combinateurs.....	26
2.3 Réseaux de neurones et apprentissage	29
2.3.1 Fonction coût	30
2.3.2 Descente de gradient et rétro-propagation de l'erreur	32
2.3.3 Les « Support Vector Machine » pour la classification.....	37
2.4 Le filtrage.....	39
2.4.1 Le rapport signal sur bruit.....	39
2.4.2 Filtre par moyenne	40
2.4.3 Filtre par l'estimation du maximum de vraisemblance	42

CHAPITRE III CONCEPTION..... 51

3.1 Réseaux d'antennes et combinateurs : une approche qualitative	53
3.1.1 Choix du cône d'arrivée.....	53
3.1.2 Premier prototype	57
3.1.3 Deuxième prototype.....	64

3.2 Optimisation des réseaux de neurones.....	68
3.2.1 Résolution à une ou deux dimensions.....	68
3.2.2 Caractéristiques générales.....	72
3.2.3 Régression pour le prototype 1	76
3.2.4 Régression pour le prototype 2	80
3.3 Choix du système : fiabilité et précision	85
3.3.1 Fiabilité de la réponse.....	85
3.3.2 Robustesse de la réponse face au bruit	89
3.3.3 Comparaisons des différents systèmes	92
CHAPITRE IV VALIDATIONS EXPERIMENTALES.....	99
4.1 Réalisation du réseau d'antennes, collecte des données	99
4.2 Apprentissage et choix du système global.....	103
4.2.1 Régression, méthode 1D	104
4.2.2 Régression, méthode 2D	106
4.2.3 Classification	108
4.3 Performance du système.....	109
CHAPITRE V CONCLUSION ET CONTRIBUTIONS	111
RÉFÉRENCES	114
ANNEXE	118

LISTE DES TABLEAUX

<i>Tableau 3.1 Valeurs optimales pour le prototype 1</i>	60
<i>Tableau 3.2 Valeurs optimales pour le prototype 2</i>	66
<i>Tableau 3.3 Valeurs de k</i>	69

LISTE DES FIGURES

<i>Figure 1.1 Détection de l'angle d'arrivée</i>	2
<i>Figure 1.2 Déphasage entre deux antennes.....</i>	3
<i>Figure 1.3 Combinaison de signaux.....</i>	4
<i>Figure 1.4 Direction d'arrivée en deux dimensions</i>	5
<i>Figure 1.5 Régression à partir d'un échantillon provenant d'un cosinus, avec un polynôme de degré 4.</i>	7
<i>Figure 1.6 Classification des maladies en fonction des symptômes.....</i>	8
<i>Figure 1.7 Exemple d'estimation de densité pour un jeu de fléchettes</i>	9
<i>Figure 1.8 Fonctionnement d'un neurone</i>	10
<i>Figure 1.9 Sigmoïde standard, $f(x) = \frac{1}{1 + \exp(-a * x)}$</i>	10
<i>Figure 1.10 Tangente hyperbolique, $f(x) = \frac{2}{1 + \exp(-2 * x)} - 1$</i>	11
<i>Figure 1.11 Fonction Gaussienne, $f(x) = \exp\left(\frac{-x^2}{a^2}\right)$</i>	11
<i>Figure 1.12 Perceptron multicouche</i>	12
<i>Figure 2.1 Couplage entre deux antennes</i>	18
<i>Figure 2.2 Géométrie en triangle</i>	19
<i>Figure 2.3 Réseau d'antennes espacées régulièrement sur un axe</i>	19
<i>Figure 2.4 Réseau d'antennes sur un axe</i>	21
<i>Figure 2.5 Partition de X indexée par k</i>	23
<i>Figure 2.6 Exemple d'un axe avec trois antennes</i>	24
<i>Figure 2.7 $A(d_1, d_2)$ pour d_1 et d_2 entre 0 et 1.....</i>	25
<i>Figure 2.8 Coupleur hybride</i>	27
<i>Figure 2.9 Combinateur sur le modèle d'une matrice de Blass triangulaire.....</i>	28
<i>Figure 2.10 (a) Erreur de Classification. (b) Erreur de régression.....</i>	31
<i>Figure 2.11 Taux d'apprentissage. [Duda, Hart, Stork (1998)].....</i>	33

<i>Figure 2.12 Conventions de notations pour un neurone</i>	35
<i>Figure 2.13 Perceptron à une couche cachée pour une régression.....</i>	35
<i>Figure 2.14 Classificateurs linéaires avec une erreur nulle</i>	38
<i>Figure 2.15 Frontière de décision avec la plus grande marge.....</i>	38
<i>Figure 2.16 (a) Répartition uniforme. (b) Filtre sur 30 points.....</i>	41
<i>Figure 2.17 Variance de X^{filtre} en fonction du nombre n de points moyennés</i>	41
<i>Figure 2.18 Maillage en carrés.</i>	43
<i>Figure 2.19 p_n pour différentes valeurs de N et de h, [Duda, Hart, Stork (1998)].</i>	45
<i>Figure 2.20 RBFN pour l'estimation de densité.....</i>	46
<i>Figure 2.21 (a) Maximum de vraisemblance. (b) Occurrences de X^{filtre}</i>	47
<i>Figure 2.22 Simulation 1</i>	49
<i>Figure 2.23 Simulation 2</i>	49
<i>Figure 2.24 Simulation 3</i>	49
<i>Figure 3.1 Organisation du chapitre 3.....</i>	52
<i>Figure 3.2 Différents cônes de direction d'arrivée</i>	54
<i>Figure 3.3 Système de coordonnées.....</i>	54
<i>Figure 3.4 $\left \frac{\partial X}{\partial \theta} \right$ sur l'intervalle $[0,180]$.....</i>	56
<i>Figure 3.5 Intervalle optimal.....</i>	56
<i>Figure 3.6 Signaux réalisant la division de l'intervalle en quatre sous-intervalles.....</i>	57
<i>Figure 3.7 Signaux optimum pour une régression sur un sous-intervalle.....</i>	58
<i>Figure 3.8 Variation des fonctions $f_i(L, \phi)$</i>	59
<i>Figure 3.9 Signaux obtenus avec les valeurs optimales</i>	60
<i>Figure 3.10 Réseau de quatre antennes avec les trois valeurs de d</i>	61
<i>Figure 3.11 Signal 2 avec $d=0.58\lambda$ ou 0.50λ.....</i>	61
<i>Figure 3.12 Réseau d'antennes et combinateur du prototype 1</i>	62
<i>Figure 3.13 Comparaison des architectures suivant la méthode utilisée.....</i>	63
<i>Figure 3.14 Signaux de sortie du prototype 1.....</i>	63
<i>Figure 3.15 Signaux d'objectif du prototype 2</i>	65

<i>Figure 3.16 Géométrie à priori du réseau et gain d'un coupleur hybride</i>	66
<i>Figure 3.17 Réseau d'antennes et combinateur pour le prototype 2</i>	67
<i>Figure 3.18 Signaux du prototype 2</i>	67
<i>Figure 3.19 Entrées normalisées pour la méthode 1D pour le prototype 1</i>	70
<i>Figure 3.20 Entrées normalisées pour la méthode 1D pour le prototype 2</i>	70
<i>Figure 3.21 Effet de la normalisation sur les signaux.....</i>	71
<i>Figure 3.22 Fonction coût L</i>	72
<i>Figure 3.23 Arrêt par validation croisée.....</i>	74
<i>Figure 3.24 Pouvoir expressif d'un MLP</i>	75
<i>Figure 3.25 Prototype 1: réseau et antenne élémentaire</i>	76
<i>Figure 3.26 Gains des antennes dans le demi-espace supérieur.....</i>	77
<i>Figure 3.27 Sorties X1,X3,X5,X7 en fonction de θ_x</i>	78
<i>Figure 3.28 Estimation de la taille du réseau pour résoudre le problème.....</i>	79
<i>Figure 3.29 MLP pour la régression à partir des signaux du prototype 1.....</i>	79
<i>Figure 3.30 Test des réseaux avec des signaux non bruités, (a) avec la méthode 1D, (b) avec la méthode 2D.....</i>	80
<i>Figure 3.31 Prototype 2 à partir de 4 antennes Patch ; Antenne Patch</i>	81
<i>Figure 3.32 Sorties du prototype 2 avec des gains dépendants de θ_x et θ_y</i>	82
<i>Figure 3.33 Estimation de la taille du réseau pour résoudre le problème.....</i>	83
<i>Figure 3.34 MLP pour la régression à partir des signaux du prototype 2.....</i>	83
<i>Figure 3.35 Test des réseaux avec des signaux non bruités, (a) avec la méthode 1D, (b) avec la méthode 2D.....</i>	84
<i>Figure 3.36 Réponses des réseaux pour des angles d'arrivée dans [0,180].....</i>	85
<i>Figure 3.37 Réseaux pour la classification des signaux avec la méthode 1D.....</i>	86
<i>Figure 3.38 (a) Répartition géométrique de [X1,X2,X3] quand θ varie dans [0 ,180] (b) Plan de classification</i>	87
<i>Figure 3.39 Cône d'étude</i>	88
<i>Figure 3.40 MLP de classification. méthode 2D.....</i>	88
<i>Figure 3.41 Classification des signaux par rapport au cône d'étude</i>	89

<i>Figure 3.42 Erreur de classification.....</i>	90
<i>Figure 3.43 Système embarqué sur un micro-satellite. (non à l'échelle).....</i>	91
<i>Figure 3.44 Choix de l'époque. (a) pour le réseau de la méthode 1D. (b) pour le réseau de la méthode 2D</i>	93
<i>Figure 3.45 Calcul de l'erreur. Ouvertures angulaires du cône.....</i>	93
<i>Figure 3.46 Performance du prototype 1 sous différentes conditions de bruit</i>	94
<i>Figure 3.47 Performance du prototype 2 sous différentes conditions de bruit</i>	94
<i>Figure 3.48 Erreur de classification (a) en fonction du niveau de bruit.....</i>	96
<i>Figure 3.49 Erreur de classification (a) en fonction du niveau de bruit (b) en fonction de la direction d'arrivée pour RSB=10dB.....</i>	96
<i>Figure 3.50 RSB=10dB. (a) Erreur de type 1. (b) Erreur de type 2.....</i>	97
<i>Figure 3.51 RSB=10dB. (a) Erreur de type 1. (b) Erreur de type 2.....</i>	97
<i>Figure 4.1 Antenne Patch carrée. (a) Vue éclatée. (b) Vue du dessous</i>	100
<i>Figure 4.2 (a) Procédé expérimental. (b) Réseau d'antennes et combinateur.....</i>	101
<i>Figure 4.3 Puissances des signaux en fonction de l'angle d'incidence</i>	102
<i>Figure 4.4 Données pour la méthode 1D dans le cône d'étude.....</i>	102
<i>Figure 4.5 Données pour la méthode 2D dans le cône d'étude.....</i>	103
<i>Figure 4.6 Erreur de régression en fonction du nombre de neurones cachés.....</i>	104
<i>Figure 4.7 Test des réseaux pour les deux axes de façon indépendante</i>	104
<i>Figure 4.8 Performance de la méthode 1D</i>	105
<i>Figure 4.9 Erreur de régression en fonction du nombre de neurones cachés.....</i>	106
<i>Figure 4.10 Cycles d'apprentissage : (a) Erreur. (b) et (c) Test du réseau.....</i>	106
<i>Figure 4.11 Performance de la méthode 2D</i>	107
<i>Figure 4.12 Erreur de classification (type 1 + type 2) en % en fonction du nombre de neurones sur la couche cachée.</i>	108
<i>Figure 4.13 Précision du système pour différentes ouvertures du cône d'arrivée</i>	109
<i>Figure 4.14 (a) Erreur de classification en fonction du bruit. (b) Classification pour RSB=10dB.....</i>	110
<i>Figure 4.15 Répartition angulaire des erreurs pour un RSB=10 dB</i>	110

Figure 5.1 Système d'estimation de la direction d'arrivée d'un signal 111

LISTE DES SIGLES ET ABRÉVIATIONS

DOA

Direction Of Arrival

DSP

Digital Signal Processing

GPS

Global Positioning System

MLP

Multi Layer Perceptron

MSat

Micro Satellite

RBFN

Radial Basis Function Network

RMS

Root Mean Square

RSB

Rapport Signal sur Bruit

SVM

Support Vector Machine

CHAPITRE I

INTRODUCTION

Connaître la direction d'arrivée d'un vaisseau est un sujet de recherche capital dans des domaines aussi variés que les télécommunications mobiles, la conquête spatiale ou le développement de matériel embarqué pour différents moyens de transport. En effet, que ce soit pour les radars, les sonars ou antennes de communication, connaître la direction d'arrivée (DOA) de façon précise et rapide peut s'avérer déterminant dans la conception des systèmes. Dans un environnement fortement bruité, l'estimation de la DOA permet de focaliser dans la bonne direction et donc augmenter les performances de réception du signal. L'Agence Spatiale Canadienne prospecte sur un système de DOA peu coûteux et fiable qui pourrait équiper des véhicules pour la conquête de Mars. L'estimation de l'angle d'arrivée d'un vaisseau pourrait servir à repérer la position d'un objet, connaître ses changements d'attitudes ou encore estimer sa vitesse. L'objectif de la recherche est donc de fournir un système complet de détection de la direction d'arrivée avec la meilleure précision possible. Cette précision devra être mesurée dans des conditions proches de celles que l'on rencontrera à bord d'un satellite.

1.1 Principe de l'estimation de la direction d'arrivée.

Quand un même signal est intercepté par deux récepteurs qui sont éloignés l'un de l'autre, la différence des signaux reçus porte une information sur l'angle d'arrivée du vaisseau. Par exemple, en acoustique, notre cerveau analyse les différences du son capté par l'oreille droite et l'oreille gauche et détermine ensuite l'angle d'où provient ce son, Figure 1.1.

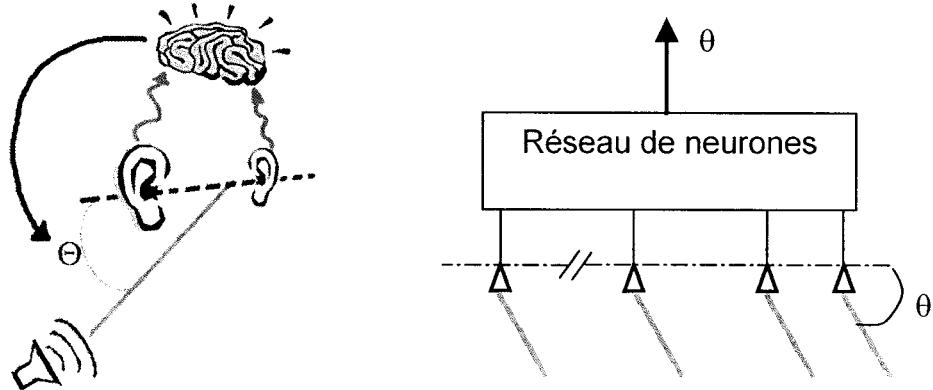


Figure 1.1 Détection de l'angle d'arrivée

Par analogie à la détection d'un son pour un être humain, dans le système que nous avons développé, c'est un réseau d'antennes qui joue le rôle des oreilles et un réseau de neurones qui remplace le cerveau, Figure 1.1.

1.2 Réseaux d'antennes

Quand deux antennes reçoivent une onde électromagnétique monochromatique plane, le déphasage Φ de l'onde entre les deux antennes (1.1) dépend de la longueur d'onde, de la distance entre les deux antennes et du cosinus de l'angle d'arrivée de l'onde par rapport à l'axe des antennes, Figure 1.2.

$$\Phi = \frac{2\pi * d}{\lambda} \cos(\theta) \quad (1.1)$$

où

λ est la longueur d'onde

d la distance entre deux antennes

θ l'angle entre la direction d'arrivée et l'axe passant par les antennes du réseau

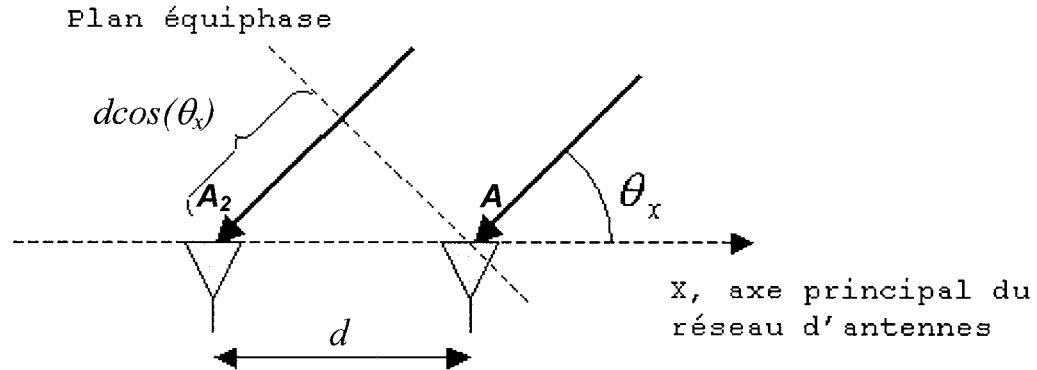


Figure 1.2 Déphasage entre deux antennes

Trouver la direction d'arrivée du faisceau consiste donc à retrouver la valeur de l'angle à partir du déphasage. Cependant connaître le cosinus d'un angle n'est pas suffisant pour caractériser la direction d'arrivée. Il faut avoir une autre information, comme le sinus de l'angle par exemple, pour éliminer l'ambiguïté à π près. En effet dans le cas de la détection du son par un être humain, il est possible de trouver la direction d'arrivée dans un demi-espace mais il est impossible de dire si le son vient de devant ou derrière. Une expérience mettant en lumière ce phénomène consiste à fermer les yeux et demander à une tierce personne de taper des mains devant ou derrière la tête : on est incapable (en dehors de tout écho) de déterminer d'où la personne a émis le son. Toute la problématique de l'estimation de la direction d'arrivée consiste donc à se donner suffisamment d'informations pour retrouver l'angle d'arrivée sans aucune ambiguïté. L'information de déphasage peut également être obtenue sous forme de puissance électrique. En effet, en sommant les signaux reçus pour les deux antennes de la Figure 1.2, la puissance résultante \$X\$ dépend du cosinus de l'angle d'arrivée \$\theta\$ (1.3) et (1.4). Soit \$A\$ l'onde électromagnétique reçue sur la première antenne, alors l'onde \$A_2\$ captée par la deuxième antenne a la forme décrite dans (1.2).

$$A_2 = A * e^{-j \frac{2\pi d}{\lambda} \cos(\theta)}$$

(1.2)

La puissance résultant de la somme de A et A_2 est alors

$$X(\theta) = \left\| A + A * e^{-j \frac{2\pi * d}{\lambda} \cos(\theta)} \right\|^2 \quad (1.3)$$

et en simplifiant à un facteur multiplicatif près,

$$\tilde{X}(\theta) = 1 + \cos\left(\frac{2\pi * d}{\lambda} \cos(\theta)\right) \quad (1.4)$$

Avec un réseau de plus de deux antennes, il est alors possible de générer $X_1 \dots X_n$ signaux de puissance en combinant les signaux provenant de différentes antennes ou en rajoutant des déphasages φ_i , (1.5) et Figure 1.3.

$$X_i(\theta) = 1 + \cos\left(\varphi_i + \frac{2\pi * d_i}{\lambda} \cos(\theta)\right) \quad (1.5)$$

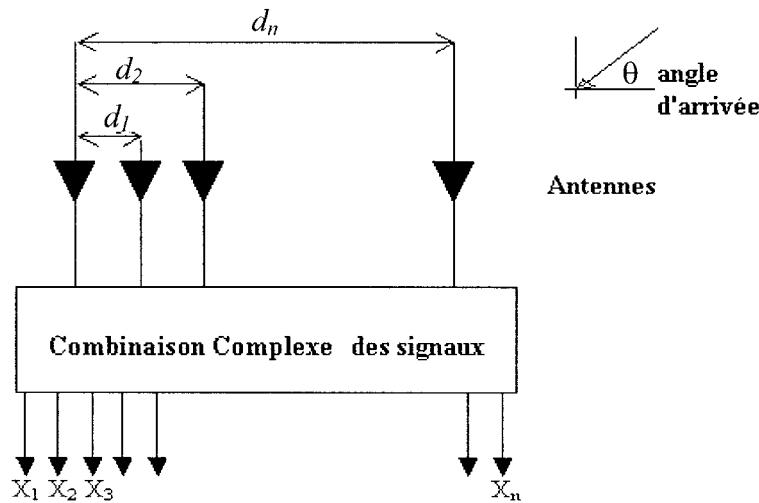


Figure 1.3 Combinaison de signaux

On peut définir à présent la fonction $Z(\theta)$ par (1.6)

$$\begin{aligned} Z(\theta) : [-\pi, \pi] &\rightarrow \Re^{\prime \prime} \\ Z(\theta) &= [X_1, X_2, X_3, \dots, X_n] \end{aligned} \quad (1.6)$$

Formellement, trouver la direction d'arrivée consiste à inverser la fonction $Z(\theta)$. Cela revient donc à créer une fonction F vérifiant (1.7).

$$\begin{aligned} F : \Re^{\prime \prime} &\rightarrow [-\pi, \pi] \\ \forall \theta \in [-\pi, \pi], \quad F \circ Z(\theta) &= \theta \end{aligned} \quad (1.7)$$

Pour l'instant nous avons seulement défini un problème planaire, c'est-à-dire que nous recherchons la direction d'arrivée par rapport à un axe. Cependant notre projet consiste à retrouver la direction d'arrivée de façon bidimensionnelle comme l'illustre la Figure 1.4 .

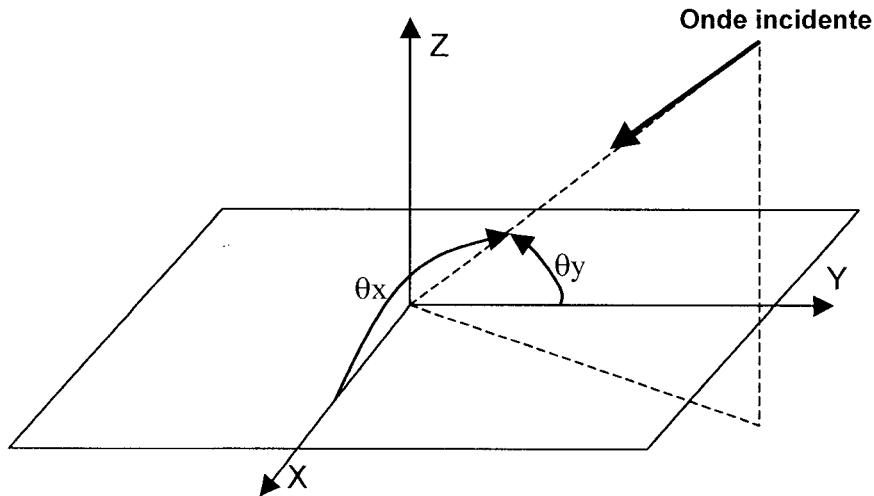


Figure 1.4 Direction d'arrivée en deux dimensions

L'équation (1.7) devient alors,

$$\forall (\theta_x, \theta_y) \in [-\pi, \pi] \times [-\pi, \pi], \quad F \circ Z(\theta_x, \theta_y) = (\theta_x, \theta_y) \quad (1.8)$$

Lors de la construction de la fonction $Z(\theta_x, \theta_y)$, il faut donc veiller à ce qu'elle puisse être inversible même en présence de bruit. Le choix du réseau d'antennes s'avère donc stratégique pour avoir accès précisément à la direction d'arrivée. Le réseau peut avoir différentes géométries : axial, circulaire, triangulaire. L'architecture du réseau d'antennes est en fait doublement importante car elle va à la fois déterminer l'inversibilité de $Z(\theta_x, \theta_y)$ mais également déterminer le couplage qui peut exister entre les antennes et donc influencer le rapport signal bruit du système.

La problématique et le vocabulaire de l'estimation de la direction d'arrivée étant définis, nous allons à présent définir les outils qui vont nous permettre de résoudre ce problème. Retrouver la fonction F à partir de $Z(\theta_x, \theta_y)$ est ce qui est appelé un problème de régression, c'est-à-dire qu'il faut créer une application qui associe $Z(\theta_x, \theta_y)$ à (θ_x, θ_y) . Or certaines catégories de réseaux de neurones sont des régresseurs universels, c'est-à-dire qu'à partir d'un ensemble de points X et d'un ensemble de points Y , ils peuvent retrouver la relation $Y=f(X)$ qui relie X à Y . Dans notre recherche nous avons donc décidé d'utiliser des réseaux de neurones pour retrouver la direction d'arrivée. Nous allons introduire dans la section 1.3 les notions de bases pour comprendre notre travail sur les réseaux de neurones.

1.3 Réseaux de neurones

Les réseaux de neurones, [Haykin, S. (1999)], constituent une branche très étudiée au sein du vaste domaine des algorithmes d'apprentissage. Un algorithme d'apprentissage est un outil qui peut apprendre une tâche seulement à partir d'exemples.

1.3.1 Définition des trois problèmes d'apprentissage

On distingue trois problèmes d'apprentissage (ou tâches) différents: la *régression*, la *classification* et l'*estimation de densité*.

La régression consiste à découvrir les relations qui lient deux variables que l'on appelle entrées et sorties, Figure 1.5. Il s'agit donc de découvrir l'application F qui va de l'espace des entrées vers l'espace des sorties et qui réalise (1.9)

$$\text{Sorties} = F(\text{Entrées}) \quad (1.9)$$

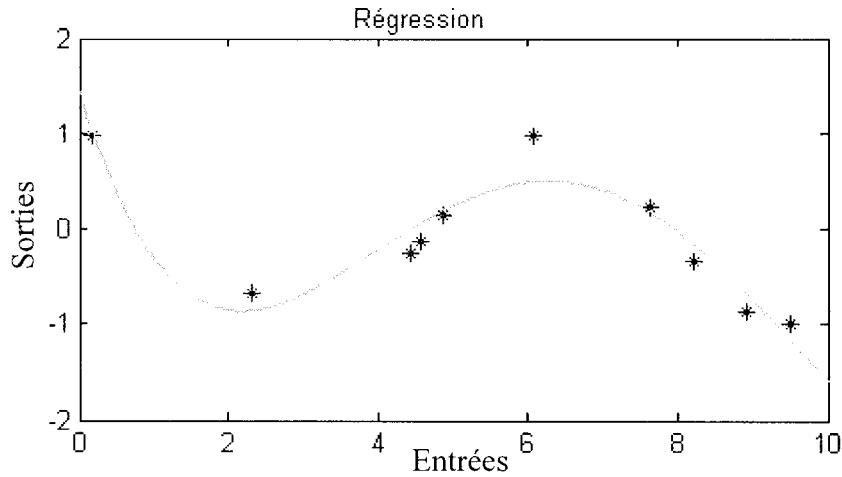


Figure 1.5 Régression à partir d'un échantillon provenant d'un cosinus, avec un polynôme de degré 4.

Par exemple dans le problème de direction d'arrivée, trouver l'angle d'arrivée du signal (Sortie) à partir des signaux reçus par les antennes (Entrées) est un problème de régression.

La classification est un procédé qui a pour but de classer des objets dans différentes catégories (ou classes) à partir de traits de caractères observables. Par exemple, quand un médecin examine un patient, il observe différents symptômes visibles comme la température, une douleur à un endroit précis ou toute autre réaction physionomique et il doit en déduire de quelle maladie (classe) souffre le patient, voir Figure 1.6. Sans le savoir, le médecin vient de faire un travail de classification : il a utilisé ses connaissances acquises pour faire correspondre des symptômes à un diagnostique.



©Barthomeuf Etienne

Figure 1.6 Classification des maladies en fonction des symptômes

L'estimation de densité a pour but de déterminer la probabilité pour qu'un événement se produise. Une fois de plus, cette estimation est basée uniquement sur l'observation et non sur l'étude physique du phénomène. Par exemple, on demande à un joueur de fléchette de lancer dix fléchettes en direction de la cible. La cible est divisée en secteurs concentriques, Figure 1.7. Pour chaque fléchette, on note dans quel secteur elle se plante. Comme il n'y a que dix fléchettes, il n'y aura au maximum que dix mesures d'éloignement différentes mais il est néanmoins possible d'estimer une probabilité pour qu'une fléchette retombe dans un secteur donné.

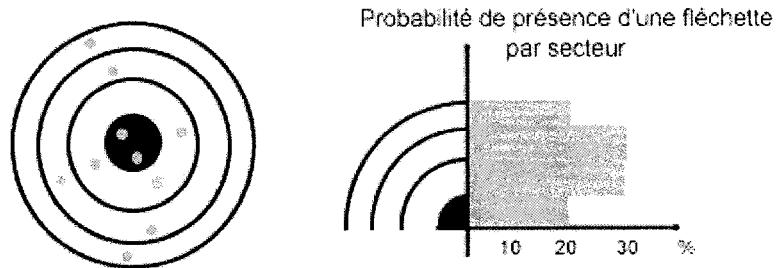


Figure 1.7 Exemple d'estimation de densité pour un jeu de fléchettes

1.3.2 Neurones, fonction d'activation, couches

Un réseau de neurones est constitué de cellules ou *neurones*. Ces cellules sont reliées entre elles par des connexions, ce qui leur permet de disposer d'un canal pour envoyer et recevoir des signaux en provenance d'autres cellules du réseau. Chacune de ces connexions reçoit une valeur, un poids, appelé poids synaptique par analogie à la biologie. Les cellules disposent d'une entrée, qui permet de recevoir de l'information d'autres cellules. Elles disposent aussi d'une *fonction d'activation*, qui est dans les cas les plus simples, une simple identité du résultat obtenu par l'entrée. La Figure 1.8 présente le fonctionnement d'un neurone formel tel que l'ont décrit McCulloch et Pitts, [McCulloch, Pitts (1943)]. Il s'agit donc d'un composant calculatoire faisant la somme pondérée des signaux reçus en entrée à laquelle est appliquée une fonction d'activation afin d'obtenir la réponse du neurone (1.10).

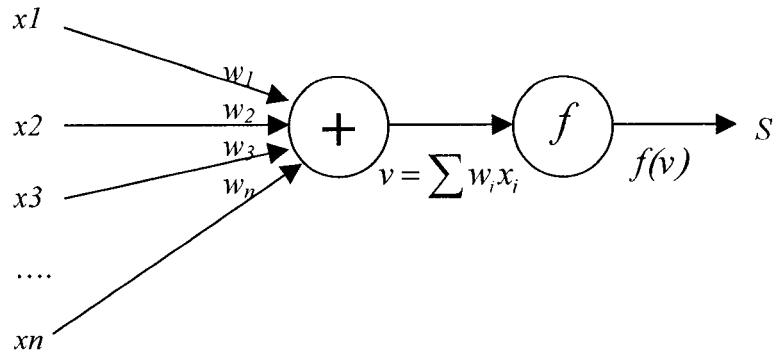


Figure 1.8 Fonctionnement d'un neurone

$$S = f\left(\sum_{i=1}^n w_i * x_i\right) \quad (1.10)$$

Le choix d'une fonction d'activation se révèle être un élément constitutif important des réseaux de neurones. Ainsi, la fonction identité n'est pas toujours suffisante et souvent des fonctions non linéaires et plus évoluées seront nécessaires. A titre illustratif, les figures ci-dessous représentent quelques fonctions couramment utilisées comme fonctions d'activation.

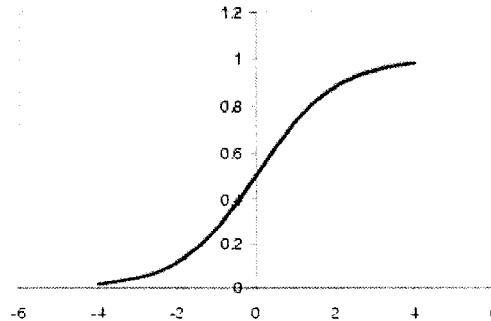


Figure 1.9 Sigmoïde standard, $f(x) = \frac{1}{1 + \exp(-\alpha * x)}$

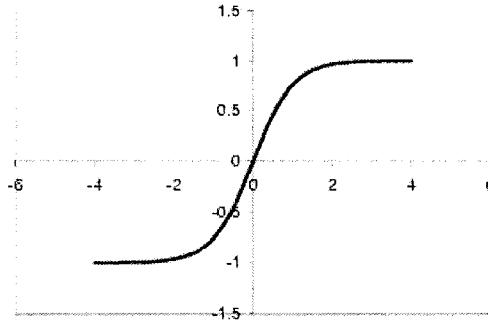


Figure 1.10 Tangente hyperbolique, $f(x) = \frac{2}{1 + \exp(-2 * x)} - 1$

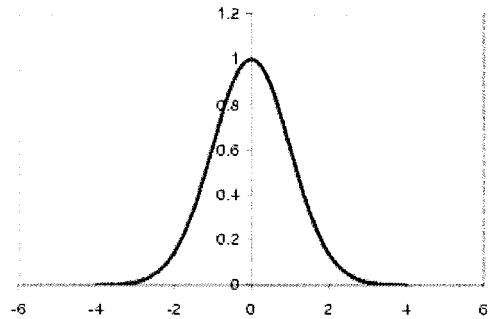


Figure 1.11 Fonction Gaussienne, $f(x) = \exp\left(\frac{-x^2}{a^2}\right)$

Un exemple simple de réseau de neurones est donné par le *perceptron multicouches*. Après les travaux effectués par Rosenblatt [Rosenblatt(1962)] sur le perceptron simple, les travaux les plus marquants sur le perceptron multicouche (MLP) ont été développés dans les années 80 par Rumelhart, Hinton, et Williams [Rumelhart, Hinton, Williams]. Un MLP peut se représenter formellement par un graphe orienté, Figure 1.12, où chaque nœud est un neurone qui se comporte comme décrit précédemment.

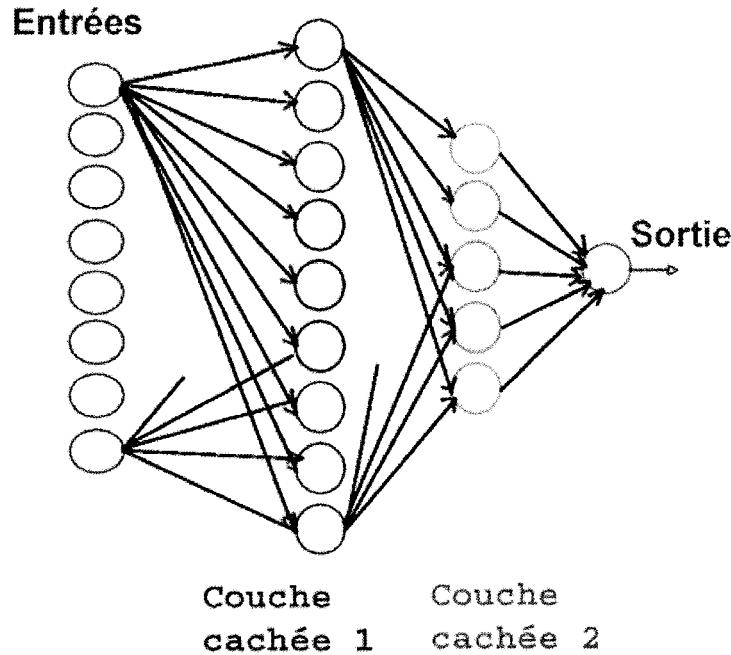


Figure 1.12 Perceptron multicouche

Afin d'illustrer un peu ces propos, voici le rôle de chacune des couches :

Les neurones d'entrée :

La première couche est appelée couche d'entrée. Elle recevra les données sources que l'on veut utiliser pour l'analyse. Dans le cas de l'estimation de la direction d'arrivée, cette couche recevra les signaux des antennes. Sa taille est donc directement déterminée par le nombre de variables d'entrées.

Les neurones cachés :

Dans l'exemple de la Figure 1.12, les deux couches du milieu s'appellent couches cachées en ce sens qu'elles n'ont pas de contact direct avec l'*extérieur*. Les fonctions d'activation sont en général non linéaires sur cette couche mais il n'y a pas de règle à respecter. Le

choix de sa taille n'est pas implicite et doit être ajusté. Le choix de la taille de ces couches est un des sujets d'étude prépondérants des réseaux de neurones.

Les neurones de sortie :

La dernière couche est appelée couche de sortie. Elle donne le résultat obtenu après le traitement des données d'entrée par le réseau. Dans le cas de l'estimation de la direction d'arrivée, cette couche donne l'angle d'arrivée. Sa taille est donc directement déterminée par le nombre de variables qu'on veut en sortie.

1.3.3 Les étapes de la conception d'un réseau

Une fois que les fonctions d'activations sont choisies, toute l'étude porte sur la découverte des poids synaptiques pour que la sortie donne la réponse désirée en fonction des entrées. Cependant, il peut être surprenant d'apprendre que pour construire un réseau de neurones, la première chose à faire n'est pas de choisir le type de réseau mais de bien choisir son/ses échantillons de données *d'apprentissage*. On appelle échantillon tout ensemble de couples entrée/sortie relatif au problème que l'on étudie. Par exemple, dans le cas de l'estimation de la direction d'arrivée, les couples sont constitués par les signaux reçus par les antennes et l'angle d'arrivée correspondant. Ce n'est que plus tard que le choix du type de réseau interviendra. Afin de clarifier un peu les idées, voici chronologiquement les trois grandes étapes qui doivent guider la création d'un réseau de neurones.

Le choix des échantillons

Le processus d'élaboration d'un réseau de neurones commence toujours par le choix des échantillons. Cette étape est cruciale car elle aider le concepteur à déterminer le type de réseau le plus approprié pour résoudre son problème. La nature de l'échantillon conditionne le type de réseau: le nombre de cellules d'entrée, le nombre de cellules de sortie et la façon dont il faudra mener l'apprentissage, les tests et la validation.

L'élaboration de la structure du réseau

La structure du réseau dépend étroitement du type des échantillons. Il faut d'abord choisir le type de réseau : dans le cas d'un perceptron multicouches, il faudra choisir le nombre de couches cachées ainsi que le nombre de neurones dans chaque couche cachée. Il est bien évident que plus la taille du réseau est grande et plus le réseau pourra réaliser des fonctions sophistiquées entre les entrées et les sorties, cependant il est peu recommandé d'avoir un réseau plus grand que nécessaire car cela pourrait être pénalisant sur le plan du temps de calcul ou encore le réseau pourrait apprendre le bruit lié aux échantillons.

L'apprentissage

L'apprentissage est la phase la plus délicate de l'élaboration d'un réseau de neurones. Elle consiste à découvrir les poids synaptiques optimaux des différentes liaisons afin que le réseau réalise la tâche désirée. Pour cela, il faut disposer d'un échantillon d'entraînement et on procède comme suit : on présente les entrées au réseau et on calcule les sorties correspondantes ; tant que les sorties du réseau ne sont pas suffisamment proches des couples entrés/sorties de l'échantillon en acceptant une marge d'erreur prédéfinie, on modifie les poids synaptiques.

De nombreux outils mathématiques ont été mis au point pour optimiser la conception des réseaux de neurones tant au niveau de l'architecture qu'au niveau du choix des échantillons ou de la méthode d'apprentissage. Les objectifs pour la maîtrise sont d'élaborer un système de repérage de DOA qui aurait une résolution proche de 1° pour un cône avec une ouverture (demi-angle au sommet) de 45° . Pour cela nous optimiserons de façon globale, l'architecture du réseau d'antennes, le choix du combinateur, la méthode de filtrage et bien entendu le réseau de neurones. Au Chapitre 2, nous étudierons donc plus en profondeur les différentes théories sur la conception des réseaux de neurones. Nous analyserons également les différentes méthodes disponibles dans la littérature sur le problème de l'estimation de la direction d'arrivée et nous porterons un regard tout particulier sur les réalisations qui utilisent des réseaux de neurones. Au Chapitre 3 nous présenterons la méthode choisie dans cette recherche pour réaliser l'estimation de la direction d'arrivée. Au Chapitre 4 nous ferons la synthèse des résultats obtenus en évaluant notre système sur des données expérimentales et enfin au Chapitre 5 nous commenterons les performances du système proposé.

CHAPITRE II

REVUE DE LITTERATURE

La technique conventionnelle de repérage de la direction d'arrivée ou DOA d'un faisceau est basée sur des systèmes radio-fréquence qui combinent des sommes et des différences des signaux captés par les antennes. Ces signaux sont ensuite traités afin d'obtenir la direction d'arrivée. Pour les satellites de communication mobile, on retrouve de tels systèmes [Huang (1991)], avec des réseaux d'antennes allant de deux à quatre éléments. Ces techniques ont une bonne résolution mais demandent un calibrage minutieux de l'ensemble du système et exigent des antennes d'excellente qualité. Une étude a également été menée [Hoyle (2002)] pour déterminer la faisabilité d'un système de détermination *d'attitude* à partir d'un réseau d'antennes relié à des récepteurs GPS. L'attitude est définie comme l'inclinaison et l'orientation du réseau d'antennes dans l'espace par rapport à une orientation de référence, ce qui en finalité est proche de l'étude sur la direction d'arrivée. L'étude a démontré que ce système basé sur le GPS offre une bonne résolution, c'est-à-dire une erreur angulaire sur l'attitude de l'ordre de 1°. Cependant le réseau de trois antennes utilisé reste encombrant puisque les distances inter-antennes sont supérieures à 5 longueurs d'onde et nécessite les signaux de plusieurs satellites (système GPS).

Depuis une décennie, on peut également lire des études qui proposent des systèmes de DOA à l'aide de réseaux de neurones. Les réseaux de neurones ou tout autre algorithme d'apprentissage, ont l'avantage de s'adapter aux données réelles qui lui sont fournies au cours de l'entraînement et cette particularité est très appréciée quand elle évite un calibrage laborieux des antennes. Les topologies des réseaux de neurones varient des réseaux de fonctions à bases radiales, RBFN, utilisés par [Lo (1994)], aux *Support Vector Machine*, SVM, choisis par [Charpentier (1999)] ou jusqu'aux algorithmes plus élaborés d'optimisations de réseaux, [El Zooghby, (1997)], [Shieh (2000)]. Dans la

section 2.1, nous analysons les différentes architectures de réseaux d'antennes et nous exposons les outils qui permettent de faire le choix de l'architecture. Dans la section 2.2, nous examinons les différentes façons de combiner les signaux entre eux. Dans la section 2.3 nous étudions les propriétés de différents réseaux de neurones. Enfin la section 2.4 présente des techniques pour diminuer l'effet du bruit superposé à un signal.

2.1 Différentes architectures de réseaux d'antennes

Comme nous l'avons vu au Chapitre 1, l'architecture du réseau d'antennes conditionne la forme des signaux qui serviront plus tard à retrouver la direction d'arrivée. Pour réaliser un système de DOA, il est donc préférable de choisir son architecture avec soin.

2.1.1 Notion de couplage inter-antennes

Une variation de courant dans un conducteur crée un champ magnétique qui rayonne autour de ce conducteur. Un circuit voisin peut alors voir apparaître une tension induite perturbatrice si la variation de courant est importante, Figure 2.1. Le coefficient de couplage entre deux antennes mesure le rapport entre la puissance reçue par l'une et la puissance émise par l'autre. Ce rapport dépend donc des propriétés électromagnétiques de chaque antenne mais également de la position de l'une par rapport à l'autre. Les courants induits dans une antenne par la présence des autres antennes sont une source de perturbation quand on cherche à capter seulement un signal extérieur aux antennes.

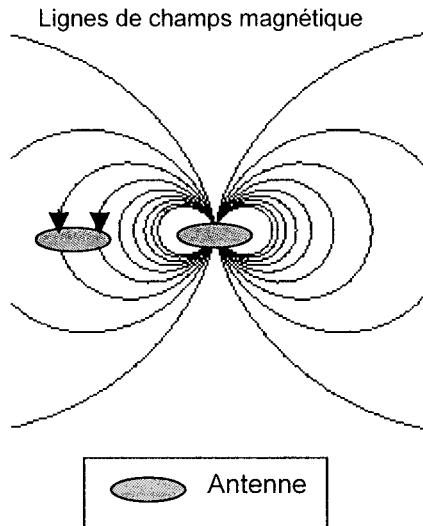


Figure 2.1 Couplage entre deux antennes

2.1.2 Architectures classiques

Réseau en triangle :

Dans [Charpentier (1999)], les trois antennes sont disposées en triangle isocèle. Cette géométrie permet théoriquement de couvrir la totalité de secteur 0-360° et de détecter un signal dans une direction d'azimut donnée. Les distances qui séparent les trois capteurs ont été calculées de façon à obtenir une variation rapide d'amplitude des signaux combinés lorsque l'angle d'arrivée du signal incident est situé près de 0° ou de 180° (i.e. axe qui est la hauteur du triangle isocèle formé par les antennes) en azimut. Dans ce cas, c'est autour de cet axe qu'on recherche le maximum de précision sur la direction du signal, Figure 2.2. Cependant la précision de la DOA à partir de ce système est selon [Charpentier (1999)] de l'ordre de 10°.

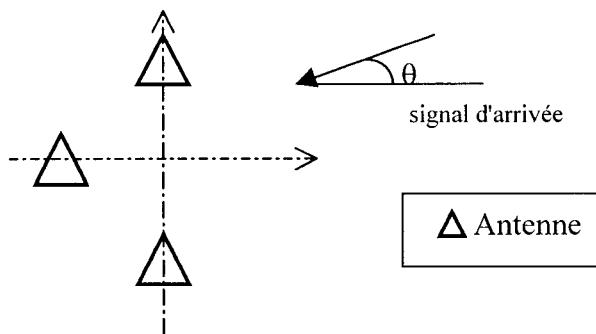


Figure 2.2 Géométrie en triangle

Réseau axial à intervalles réguliers:

L'architecture la plus couramment rencontrée dans la littérature est le réseau d'antennes disposées sur un axe et espacées régulièrement, Figure 2.3. Cette disposition adoptée par [El Zooghby, (1997)] et [Lo (1994)] est décrite dans de nombreux livres traitant de réseaux d'antennes [Johnson 1993].

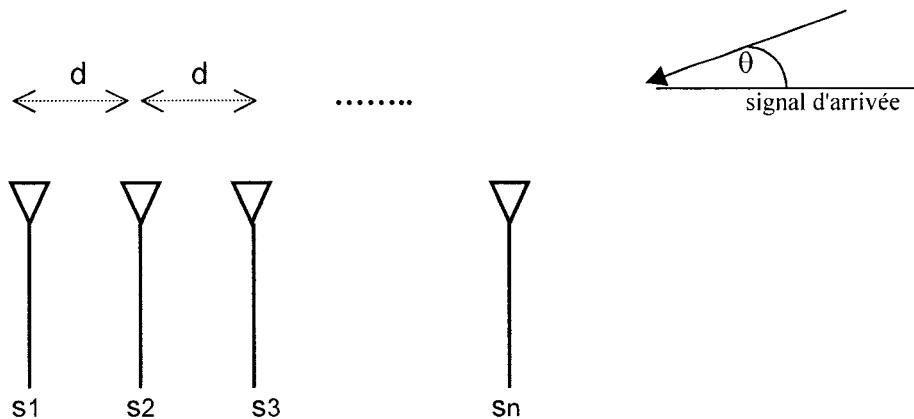


Figure 2.3 Réseau d'antennes espacées régulièrement sur un axe

Les signaux captés par ce réseau de n antennes réalisent une bijection entre l'espace des angles d'arrivées, $\{0^\circ - 180^\circ\}$ et l'espace de sortie des capteurs $\{ S = [s_1, s_2, s_3, \dots, s_n] \}$.

$$s_m = g_m e^{j2\pi \left(\frac{md}{\lambda} \cos(\theta) \right)}$$

(2.1)

où

g_m est l'amplitude captée par la $m^{\text{ème}}$ antenne,

λ la longueur d'onde du signal d'arrivée

θ l'angle d'arrivée.

2.1.3 Architectures optimisées

On peut définir le problème de l'ambiguïté lors de l'estimation de la direction d'arrivée comme la précision théorique d'un réseau d'antennes pour un angle donné [Gavish (1995)]. Comme nous le verrons par la suite, du point de vue de l'ambiguïté, l'architecture linéaire n'est pas optimum en terme de précision par rapport au nombre d'antennes.

Réseau axial général :

Dans [Shieh (2000)], la position des antennes sur un axe a été optimisée afin de diminuer l'ambiguïté conformément au travaux de Goodwin, [Goodwin (1976)]. L'étude de l'ambiguïté de la DOA permet donc de diminuer le nombre d'antennes tout en maintenant une résolution finale théorique équivalente. La diminution du nombre d'antennes est tout à fait prépondérante quand on pense à la réalisation finale du projet : cela permet de diminuer le couplage entre les antennes et donc de disposer en sortie de signaux plus faciles à analyser.

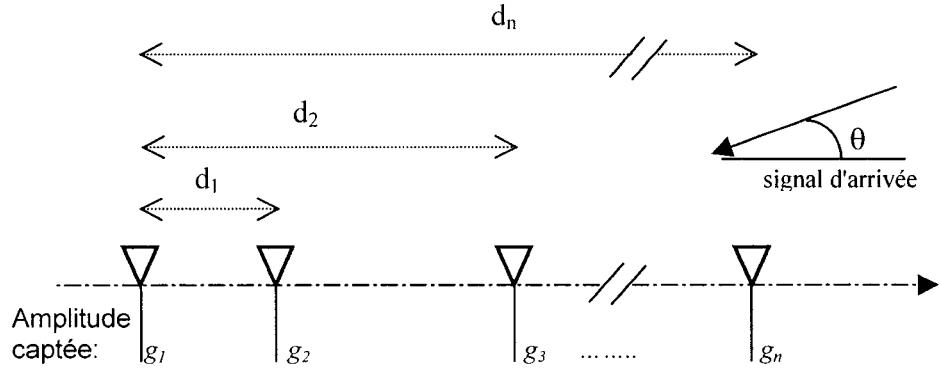


Figure 2.4 Réseau d'antennes sur un axe

Pour optimiser la disposition des antennes sur l'axe, Gavish [Gavish (1995)] introduit une fonction $Q(\theta_1, \theta_2)$ qui mesure l'écart des réponses du réseau pour deux angles θ_1 et θ_2 différents. Le but de l'optimisation est donc que Q soit maximale sur l'intervalle des angles d'arrivée considéré. Si l'on se place dans le cas d'un réseau de n antennes, on définit la réponse du réseau par $S(\theta)$ par (2.2)

$$S(\theta) = [g_1, g_2 e^{j2\pi d_1 \cos \theta}, g_3 e^{j2\pi d_2 \cos \theta}, \dots, g_n e^{j2\pi d_{n-1} \cos \theta}] \quad (2.2)$$

où

d_n est la distance entre la première et la $n^{\text{ième}}$ antenne exprimée en multiple de la longueur d'onde.

Chaque vecteur $S(\theta)$ est caractérisé par un vecteur de phase :

$$\Phi(\theta) = 2\pi [0, d_1 \cos \theta, d_2 \cos \theta, \dots, d_{n-1} \cos \theta] \quad (2.3)$$

et une matrice regroupant les amplitudes captée appelée matrice de gain G :

$$G = \text{diag}\{g_1, g_2, \dots, g_n\} \quad (2.4)$$

Avec ces notations on peut définir la fonction Q :

$$Q(\theta_1, \theta_2) = \frac{1}{4\pi^2} \|G.(\Phi(\theta_1) - \Phi(\theta_2))_{2\pi}^T\|^2 \quad (2.5)$$

où

$(Z)_{2\pi}$ représente Z modulo 2π pris dans $[-\pi, \pi]$

A présent pour réaliser l'optimisation, on peut simplifier les notations en posant :

$$d = [d_1, d_2, \dots, d_n]^T \quad (2.6)$$

$$x = \cos \theta_1 - \cos \theta_2 \quad (2.7)$$

En combinant (2.5), (2.6) et (2.7) on obtient :

$$Q(x) = \|G.(xd - k)\|^2 \quad (2.8)$$

où

k est un vecteur de nombres entiers choisis de sorte que chaque élément du vecteur $xd - k$ soit dans l'intervalle $[-1/2, 1/2]$.

Notons que x prend ses valeurs dans un intervalle X qui dépend de l'intervalle de θ . On peut définir K comme l'ensemble des valeurs prises par k . On peut diviser X en sous-intervalles en fonction des valeurs de k . En d'autres termes, on peut réaliser une partition de X indexée par k , i.e. k est fixe sur chaque sous-intervalle et chaque sous-intervalle correspond à un k différent, comme illustré à la Figure 2.5.

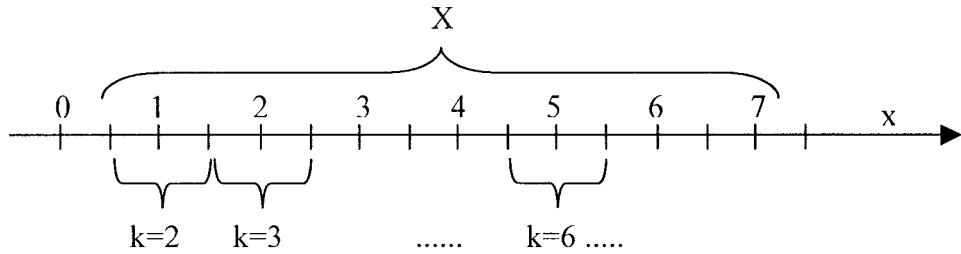


Figure 2.5 Partition de X indexée par k .

Pour un k donné, on remarque que pour tout x appartenant au sous-intervalle correspondant :

$$Q(x) \geq \|Dk\|^2 \quad (2.9)$$

où

D est la matrice :

$$D = [Gd(d^T G^2 d)^{-1} d^T G - I]G \quad (2.10)$$

Soit A définie par

$$A = \min_{k \in K} \|Dk\|^2 \quad (2.11)$$

Par définition, A peut être utilisé comme une mesure de l'ambiguïté d'une architecture. En conclusion pour découvrir l'architecture optimum il faut effectuer la liste exhaustive des architectures possibles en fonctions de nos besoins puis calculer A associé à chaque architecture et enfin choisir l'architecture avec le plus grand A .

Cependant, on peut raffiner l'étude et prendre comme mesure de l'ambiguïté l'espérance de $Q(x)$. Pour illustrer cette méthode, nous allons l'appliquer pour un système de trois antennes supposées idéales, i.e. la matrice de gain G est définie égale à l'identité. Pour le calcul de $\Phi(\theta)$ on va tenir compte des trois distances inter-antennes. Figure 2.6 . On

remarque toutefois que les trois distances ne sont pas linéairement indépendantes et l'optimisation ne portera donc que sur deux paramètres.

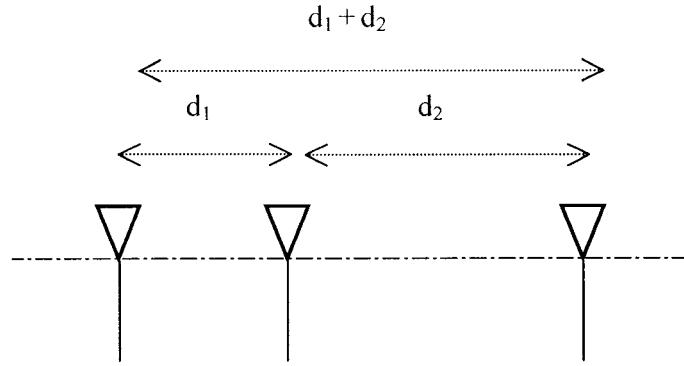


Figure 2.6 Exemple d'un axe avec trois antennes

Avec ces hypothèses (2.3) et (2.8) deviennent :

$$\Phi(\theta) = 2\pi [0, d_1 \cos \theta, d_2 \cos \theta, (d_1 + d_2) \cos \theta] \quad (2.12)$$

$$Q(x) = (d_1 x - k_1)^2 + (d_2 x - k_2)^2 + ((d_1 + d_2)x - k_3)^2 \quad (2.13)$$

On définit A comme une espérance :

$$A(d_1, d_2) = \int_{x \in X} Q(x) p(x) dx \quad (2.14)$$

où p est la densité de x .

On remarque que la fonction A est symétrique i.e. $A(x,y) = A(y,x)$ ce qui signifie que la fonction est symétrique par rapport à la droite définie par $d_1 = d_2$.

L'implantation des fonctions Q et A a été effectuée sous Matlab et les codes sont disponibles en Annexe. Le calcul de la fonction A , en faisant varier d_1 et d_2 entre 0 et 1, Figure 2.7, a permis de déterminer que le maximum est atteint pour $d_1 = d_2 = 0.667$.

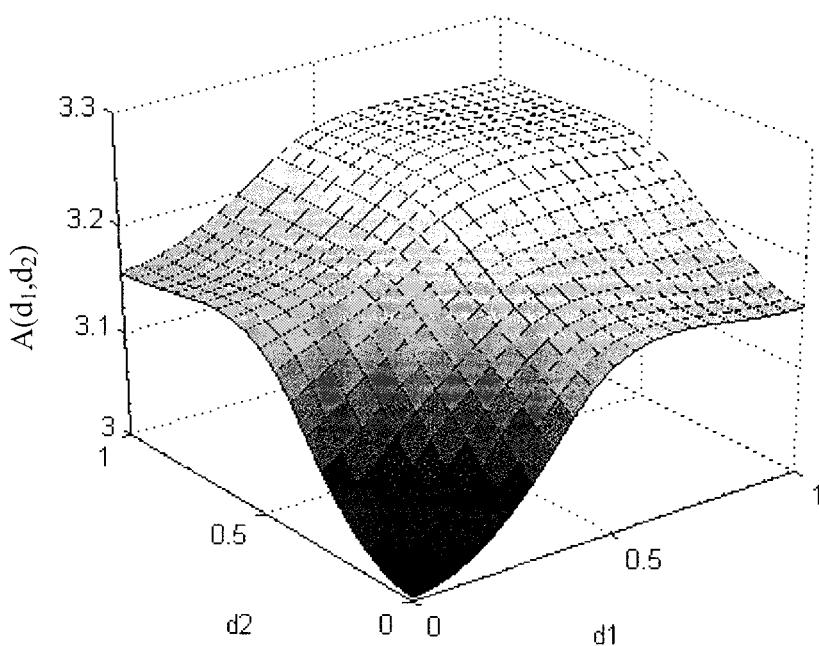


Figure 2.7 $A(d_1, d_2)$ pour d_1 et d_2 entre 0 et 1.

Nous venons de parcourir différentes architectures de réseaux d'antennes et nous avons également étudié une méthode pour optimiser la position des antennes en se basant sur la notion d'ambiguïté pour deux angles d'arrivée. Nous allons à présent, considérer différentes manières de combiner les signaux captés par les antennes afin de retrouver l'information sur l'angle d'arrivée.

2.2 Différents choix de combinateurs

Une fois que l'on dispose d'un réseau d'antennes, il faut construire les signaux qui porteront l'information de l'angle d'arrivée. Nous avons établi à la section 1.2 que l'information était portée par les différences de phase des signaux qui émanent des antennes et que cette information peut être accessible sous la forme d'une puissance.

Les comparateurs de phases :

En comparant les phases des signaux provenant des différentes antennes, on a un accès direct à l'information pertinente. El Zooghby [El Zooghby, (1997)], Shieh [Shieh (2000)] et Lo [Lo (1994)] utilisent ce type de composants dans leurs systèmes. Cela permet de restituer l'information de l'angle d'arrivée sous la forme de déphasages mesurés entre les différentes antennes du réseau. Cependant, les comparateurs de phase sont très exigeants en terme de paramétrage et de synchronisation ce qui rend leur utilisation délicate à bord d'un satellite.

Les récepteurs GPS :

Une étude, [Hoyle (2002)], a démontré qu'il était possible de faire de l'estimation de l'attitude à partir de signaux provenant de récepteurs GPS. Une des grandes difficultés consiste à synchroniser les différents récepteurs afin que la résolution puisse être conduite correctement.

Les coupleurs hybrides :

Un coupleur hybride prend en entrée les signaux provenant de deux antennes et donne en sortie la puissance résultant des somme et différence de ces deux signaux. De plus au moment de la somme et de la différence un déphasage de 90° est ajouté à l'un des deux signaux (Figure 2.8). Charpentier [Charpentier (1999)] a basé son combinateur de

signaux sur l'usage de plusieurs coupleurs hybrides. Qualitativement, il faut créer des signaux somme ou différence qui ont leur maximum pour différentes valeurs de l'angle d'arrivée. La finalité étant d'avoir suffisamment de signaux pour couvrir tout l'intervalle des angles d'arrivée avec un niveau de puissance suffisamment élevé pour être robuste au bruit.

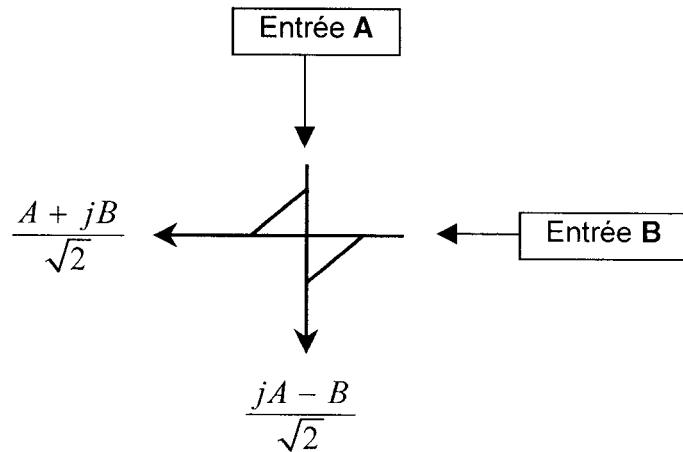


Figure 2.8 Coupleur hybride

Les matrices de Blass :

Une matrice de Blass est un agencement spécifique de coupleurs hybrides afin de créer des signaux possédant les mêmes niveaux de puissance. Dans l'exemple de la Figure 2.9, les trois sorties U_1 , U_2 et U_3 ont le même niveau de puissance en supposant, bien entendu, que les antennes possèdent les mêmes gains. L'uniformité des puissances de sortie est particulièrement importante en présence de bruit car cela nous assure que les trois sorties présentent le même rapport signal sur bruit.

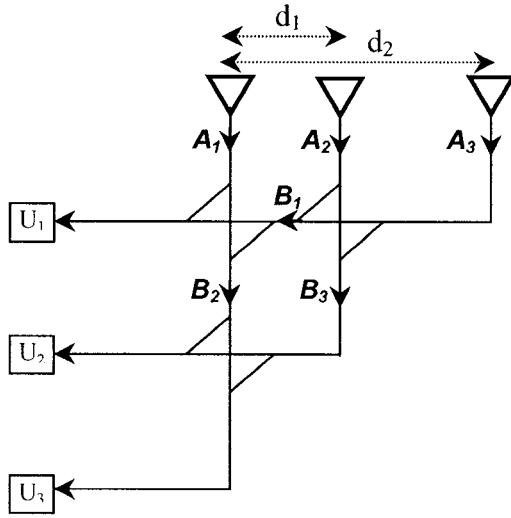


Figure 2.9 Combinateur sur le modèle d'une matrice de Blass triangulaire

Analytiquement, les puissances en sortie du combinateur sont exprimées par (2.15).

$$\begin{aligned}
 B_1 &= \frac{A_2 + jA_3}{\sqrt{2}} & U_1 &= \left\| \frac{A_1 + jB_1}{\sqrt{2}} \right\|^2 \\
 B_2 &= \frac{A_1 + jB_1}{\sqrt{2}} & U_2 &= \left\| \frac{B_2 + jB_3}{\sqrt{2}} \right\|^2 \\
 B_3 &= \frac{jA_2 - A_3}{\sqrt{2}} & U_3 &= \left\| \frac{jB_2 - B_3}{\sqrt{2}} \right\|^2
 \end{aligned} \tag{ 2.15}$$

Si on fait l'hypothèse que les puissances captées par les trois antennes ont le même niveau de puissance, alors les niveaux de puissance de U_1 U_2 et U_3 sont les mêmes et on peut l'exprimer à un coefficient multiplicatif près par (2.16).

$$\max_{\theta}(\|U_{1,2,3}\|^2) = \max_{\theta} \left(1 - \frac{1}{\sqrt{2}} \left(\sin\left(2\pi \frac{d_2 - d_1}{\lambda} \cos(\theta)\right) - \cos\left(2\pi \frac{d_2}{\lambda} \cos(\theta)\right) - \sin\left(2\pi \frac{d_1}{\lambda} \cos(\theta)\right) \right) \right)$$

(2.16)

où

λ est la longueur d'onde

d_1 et d_2 les distances de deux antennes par rapport à la troisième

θ l'angle entre la direction d'arrivée et l'axe passant par les antennes du réseau

Dans la section 3.1.3 nous étudierons un exemple de combinateur sur le modèle des matrices de Blass et nous pourrons vérifier cette propriété.

Nous venons de présenter les architectures de réseaux d'antennes les plus souvent rencontrées dans la littérature, ainsi que différentes manières de combiner les signaux entre eux. Nous allons à présent étudier la manière d'interpréter ces signaux avec un réseau de neurones afin de découvrir l'angle d'arrivée. Pour cela nous allons exposer la théorie nécessaire à la compréhension des méthodes que nous avons choisies.

2.3 Réseaux de neurones et apprentissage

Au Chapitre 1, nous avons défini les grandes étapes dans l'étude des réseaux de neurones. Nous allons à présent exposer les bases théoriques nécessaires à la compréhension du

projet

2.3.1 Fonction coût

Nous allons redéfinir d'un point de vue mathématique, les trois problèmes d'apprentissage que sont la classification, la régression et l'estimation de densité de façon formelle. Pour simplifier, en ce qui concerne la classification, nous ne considérons que le problème à deux classes.

Tout d'abord, une donnée d'apprentissage Z se définit :

- Pour la classification : $Z = (X, Y) \in \mathbb{R}^d \times \{-1, 1\}$
- Pour la régression : $Z = (X, Y) \in \mathbb{R}^d \times \mathbb{R}$
- Pour l'estimation de densité : $Z \in \mathbb{R}^d$

Pour chacun des problèmes, l'ensemble F des solutions possibles est :

- Pour la classification : $F = \{\xi, \xi : \mathbb{R}^d \rightarrow \{-1, 1\}\}$
- Pour la régression : $F = \{\xi, \xi : \mathbb{R}^d \rightarrow \mathbb{R}\}$
- Pour l'estimation de densité : $F = \{\xi, \text{fonction de densité}\}$

Pour un échantillon d'apprentissage et une solution ξ , on définit une fonction coût $J(\xi)$ comme la somme des erreurs $L(\xi, Z)$ sur chaque donnée Z .

$$J(\xi) = \sum_{\text{échantillon}} L(\xi, Z), \text{ en définissant par exemple } L \text{ comme suit.}$$

- Pour la classification: $L(\xi, Z) = L((X, Y), \xi) = \begin{cases} 0 & \text{si } \xi(X) = Y \\ 1 & \text{si } \xi(X) \neq Y \end{cases}$
- Pour la régression : $L(\xi, Z) = L((X, Y), \xi) = (Y - \xi(X))^2$
- Pour l'estimation de densité : $L(\xi, Z) = -\log \xi(Z)$

La Figure 2.10(a) illustre un problème de classification. L'échantillon d'entraînement est constitué de points pris dans $\mathbb{R}^2 \times \{-1,1\}$. La fonction de classification est dans le cas présent une application linéaire sur \mathbb{R}^2 . Enfin l'erreur de classification vaut 1 si la donnée est mal classifiée.

La Figure 2.10(b) illustre un problème de régression. Les données d'entraînement sont constituées de points (x,y) tels que $y=\cos(x)$. La fonction de régression est dans le cas présent un polynôme de degré 3 sur \mathbb{R} . L'erreur est l'écart quadratique comme défini précédemment.

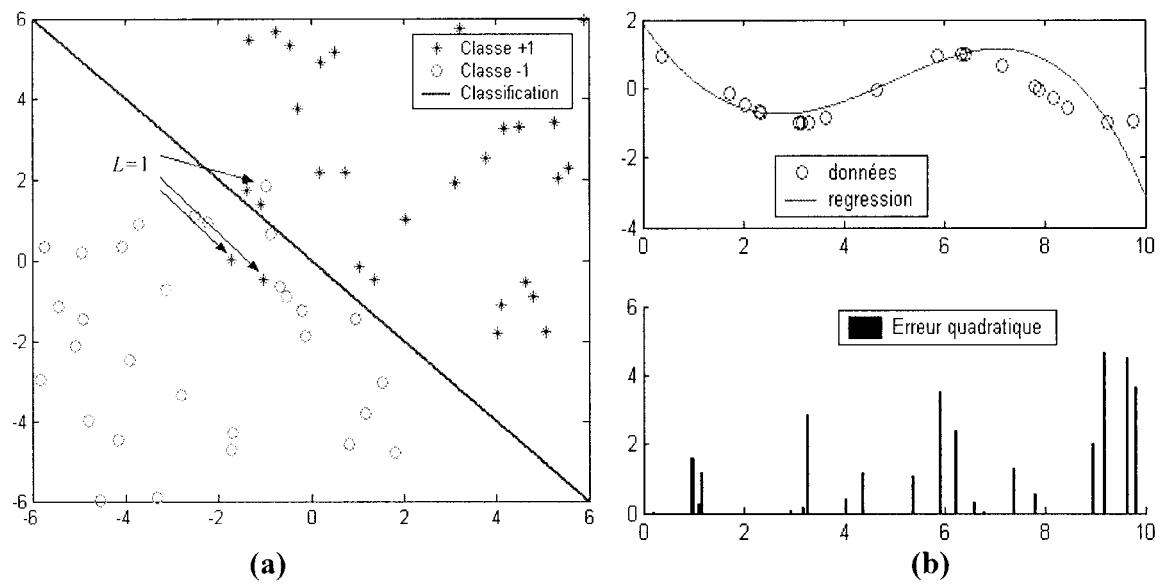


Figure 2.10 (a) Erreur de Classification. (b) Erreur de régression

L'apprentissage consiste donc à trouver la fonction ξ^* qui minimise l'espérance mathématique de $J(\xi, Z)$.

$$\begin{cases} J(\xi) = \sum_{echantillon} L(\xi, Z) \\ \xi^* = \arg \min_{\xi} (J(\xi)) \end{cases} \quad (2.17)$$

Ainsi, une fois que la fonction J est définie pour le problème étudié, le choix des poids synaptiques (section 1.3.2) devient un problème d'optimisation. Il existe différentes heuristiques de recherche de minima disponibles dans la littérature qui varient selon le problème -classification, régression ou estimation de densité- et selon l'architecture du réseau employé. Nous allons dans la suite présenter quelques méthodes qui sont tout particulièrement utilisées dans nos travaux.

2.3.2 Descente de gradient et rétro-propagation de l'erreur

Pour le perceptron ou le perceptron multicouches la fonction ξ telle que définie précédemment peut être identifiée à l'ensemble des poids synaptiques du réseau.

Soit W le vecteur constitué de l'ensemble des poids w_i de chaque arc du réseau alors,

$$L(\xi, Z) = L(W, Z) = L([w_1, \dots, w_n], Z) \quad (2.18)$$

La méthode d'apprentissage la plus courante pour le perceptron multicouche est la *rétro-propagation de l'erreur*. Cette méthode est basée sur l'optimisation par *descente du gradient*.

Descente de gradient, taux d'apprentissage :

La descente du gradient est une méthode d'optimisation locale qui consiste à trouver le minimum d'une fonction en se déplaçant dans le sens opposé du gradient de la fonction.

Soit Δw la correction des poids par descente de gradient,

$$\Delta w = -\eta \frac{\partial L}{\partial w} \quad (2.19)$$

où

η est un paramètre appelé le taux d'apprentissage.

Il existe des méthodes pour accélérer la recherche de minimum en optimisant η comme illustré à la Figure 2.11 .

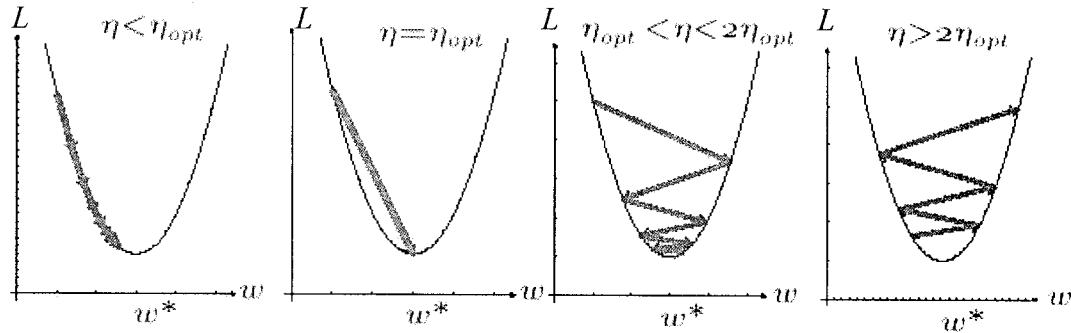


Figure 2.11 Taux d'apprentissage, [Duda, Hart, Stork (1998)]

Les méthodes de second-ordre proposent d'optimiser la valeur du paramètre en calculant la dérivée seconde (la matrice Hessienne) de L .

En effet, si l'on calcule l'approximation de Taylor-Young (Annexe) à l'ordre 2 :

$$\begin{aligned} \Delta L(w) &= L(w + \Delta w) - L(w) \\ &\approx \left(\frac{\partial L(w)}{\partial w} \right)' \Delta w + \frac{1}{2} \Delta w^T H \Delta w \end{aligned} \quad (2.20)$$

où

H est la matrice Hessienne.

En dérivant (2.20) par rapport à Δw , on trouve que ΔL s'annule pour :

$$\frac{\partial L(w)}{\partial w} + H\Delta w = 0 \quad (2.21)$$

et donc la variation optimum des poids, en supposant que H soit inversible est :

$$\Delta w^* = -H^{-1} \left(\frac{\partial L(w)}{\partial w} \right) \quad (2.22)$$

L'évaluation de η_{opt} nécessite donc l'inversion de la Hessienne, [Duda, Hart, Stork (Chap 6.9)] , ce qui est risqué car cette matrice n'est pas nécessairement inversible. L'optimisation de Levenberg-Marquart [Levenberg (1944)][Marquardt (1963)] propose une estimation de l'inverse de la matrice Hessienne, H^{LM} , et permet donc de calculer une approximation du paramètre η_{opt} .

$$H^{LM} = [H + \mu I]^{-1} \quad (2.23)$$

où

I est la matrice identité

μ est un coefficient qu'il faut minimiser dépendamment des valeurs propres de H

Rétro-propagation de l'erreur :

Pour évaluer les corrections à effectuer sur tous les poids du perceptron multicouches, on peut appliquer l'algorithme de la *rétro-propagation de l'erreur*. Nous utiliserons les notations de la Figure 2.12 et Figure 2.13 où x est l'entrée, y la sortie et t est l'objectif.

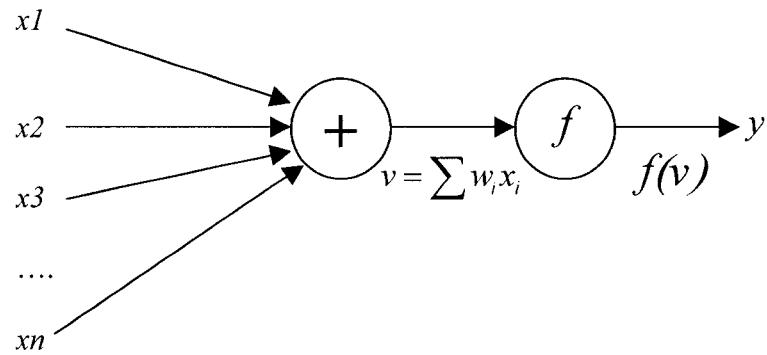


Figure 2.12 Conventions de notations pour un neurone

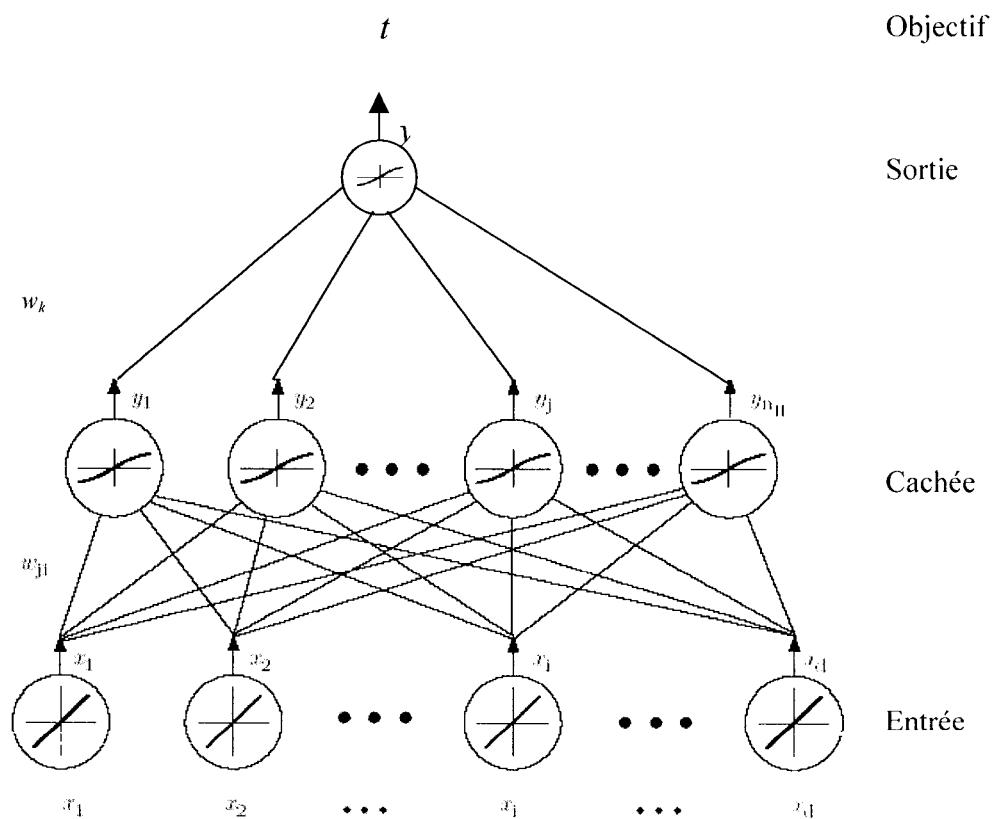


Figure 2.13 Perceptron à une couche cachée pour une régression

On peut alors définir,

- Erreur d'entraînement :

$$L(w) = \frac{1}{2}(t - y)^2 \quad (2.24)$$

- Descente de gradient :

$$\Delta w_{pq} = -\eta \frac{\partial L}{\partial w_{pq}} \quad (2.25)$$

- Mise à jour des poids à l'étape m :

$$w(m+1) = w(m) + \Delta w(m) \quad (2.26)$$

- Définition de la sensibilité δ d'un neurone, [Duda, Hart, Stork (1998)],

Avec les notations de la Figure 2.12,

- Pour la couche de sortie :

$$\delta_s = -\frac{\partial L}{\partial v} = (t - y)f'(v_s) \quad (2.27)$$

- Pour les couches cachées,

$$\delta_i = \left(\sum_{j=1}^{n_H} w_{ji} \delta_j \right) f'(v_j) \quad (2.28)$$

La formule générale de la rétro-propagation de l'erreur, [Haykin, chap4], à l'étape m:

$$\begin{pmatrix} \text{Correction} \\ \text{des poids} \\ \Delta w_{ji}(m) \end{pmatrix} = \begin{pmatrix} \text{taux} \\ \text{d'apprentissage} \\ \eta(m) \end{pmatrix} \cdot \begin{pmatrix} \text{sensibilité} \\ \text{neuronale} \\ \delta_j(m) \end{pmatrix} \cdot \begin{pmatrix} \text{i}^{\text{eme}} \text{ signal d'entrée} \\ \text{au neurone j} \\ x_i(m) \end{pmatrix} \quad (2.29)$$

L'algorithme de la rétro-propagation de l'erreur porte son nom du fait que l'erreur se calcule de la couche d'entrée vers la couche de sortie, alors que la sensibilité des neurones se calcule de la couche de sortie vers la couche d'entrée.

Ainsi à chaque étape m, on corrige les poids synaptiques en fonction de l'erreur commise sur tout l'échantillon d'entraînement.

Algorithme d'apprentissage :

Soit W le vecteur constitué de l'ensemble des poids w_i

```

1      initialiser  $W$ , époque=0
2      faire
3          époque=époque+1
4           $\Delta W \leftarrow 0$ 
5          pour  $k=1$  à  $n$  faire (où n est la taille de l'échantillon d'entraînement)
6               $x \leftarrow x_k$ 
7              Calcul de l'approximation de  $H^{-1} : H^{LM}$ 
8               $\Delta W_k = -H^{LM} \frac{\partial L(x, W)}{\partial W}$ 
9               $\Delta W \leftarrow \Delta W + \Delta W_k$ 
10          $W = W + \Delta W$  , jusqu'à ce que  $\Delta W < \varepsilon$  , ou autre critère d'arrêt
11         retourner  $W$ 
```

2.3.3 Les « Support Vector Machine » pour la classification

Quand on veut faire de la classification sur deux classes, il y a parfois plusieurs solutions équivalentes par rapport à l'erreur d'entraînement, Figure 2.14.

Il existe cependant un critère pour identifier un classificateur qui serait meilleur que les autres. En effet, on appelle *marge* la plus petite distance séparant la frontière de décision à un élément de l'une des classes. Les vecteurs de supports sont les éléments d'une classe ayant la plus petite marge, Figure 2.15. L'apprentissage de la meilleure frontière de

décision se fait en cherchant la frontière possédant la plus grande marge par rapport à l'échantillon d'entraînement.

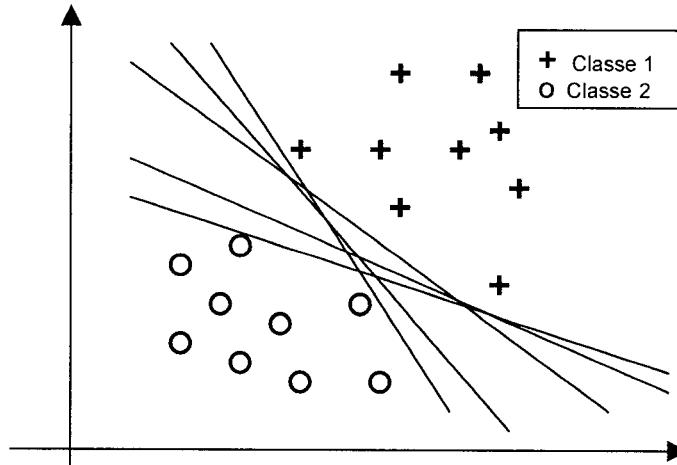


Figure 2.14 Classificateurs linéaires avec une erreur nulle

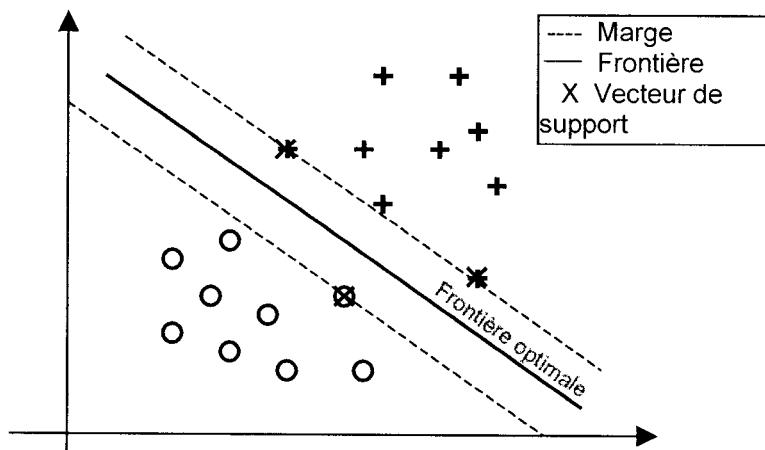


Figure 2.15 Frontière de décision avec la plus grande marge

2.4 Le filtrage

En traitement du signal, les données auxquelles on a accès sont souvent bruitées, c'est à dire qu'en plus du phénomène que l'on étudie, il se peut que des petites perturbations viennent modifier les résultats. Ces perturbations (ou bruit) peuvent provenir de l'environnement ou des appareils de mesure. Ainsi, il est souvent utile de filtrer les données, c'est-à-dire de réduire ce bruit, avant d'interpréter les résultats. Les méthodes de filtrage que l'on présente ici sont issues de propriétés statistiques et probabilistes c'est-à-dire que le bruit sera réduit en comparant (en manipulant) des signaux bruités qui proviennent du même phénomène physique.

2.4.1 Le rapport signal sur bruit

Dans le problème de l'estimation de la direction d'arrivée le *rapport signal sur bruit* (RSB) est un paramètre prédominant dans l'interprétation des résultats. En effet, les données qui parviendront au réseau de neurones auront été affectées par des perturbations pouvant provenir de l'extérieur : rayonnement solaire, réflexion et diffraction du signal et par des perturbations internes : bruit thermique, instabilité de certains composants. L'outil pour mesurer l'importance relative du bruit est le rapport signal sur bruit. Mathématiquement cela se détermine de la façon suivante :

Pour un bruit de moyenne nulle, d'écart type σ_{bruit} et un signal d'amplitude S :

$$RSB = \frac{S}{\sigma_{bruit}} \quad (2.30)$$

Par exemple, dans le cas d'un bruit suivant une loi X , uniforme entre -a et a :

La variance vaut alors

$$\text{Var}(X) = 2 \frac{a^3}{3} \quad (2.31)$$

donc si S=1,

$$RSB = \sqrt{\frac{3}{2a^3}} \quad (2.32)$$

En traitement du signal, le RSB est souvent exprimé en décibel. De plus, les signaux que nous traiterons seront des signaux de puissance et donc par définition,

$$(RSB)_{dB} = 10 \log(RSB_{réel})$$

2.4.2 Filtre par moyenne

D'après le théorème de la limite centrale :

La moyenne d'une séquence $\{X_i\}$ indépendante et identiquement distribuée converge en probabilité vers une loi Normale [Annexe] de moyenne $E[X]$ et de variance $Var(X)/n$.

Si X_i est une occurrence d'un bruit de moyenne nulle alors si $F = \frac{1}{n} \sum X_i$, alors σ , la

variance de F converge vers

$$\sigma = \sqrt{\frac{Var(X)}{n}} \quad (2.33)$$

et donc si F est la sortie d'un filtre qui fait la moyenne de n entrées :

$$\sigma_{estimé}^{Filtre} = \frac{\sqrt{Var(X)}}{\sqrt{n}} \quad (2.34)$$

En conséquence,

$$RSB_{estimé}^{Filtre} = \sqrt{n} * RSB_{sans filtre}^{sans filtre} \quad (2.35)$$

Le filtrage par moyenne améliore donc le rapport signal sur bruit dans l'hypothèse où le bruit est de moyenne nulle et provient toujours de la même origine.

Par exemple Figure 2.16(a) présente les occurrences de 10000 points issus d'une distribution uniforme X entre -0.5 et +0.5. La Figure 2.16(b) présente la moyenne X^{filtre} sur 30 points de cette même distribution X .

$$\text{D'après (2.31), } \text{Var}(X) = 2 * \frac{0.5^3}{3} = \frac{1}{12}.$$

D'après (2.33), $\text{Var}(X^{\text{filtre}})$ converge vers $\frac{\text{Var}(X)}{30}$ soit 0.0028.

Expérimentalement, Figure 2.17, on vérifie la convergence rapide du théorème de la limite centrale. Cette mesure a été réalisée à partir des outils de statistique de Matlab®.

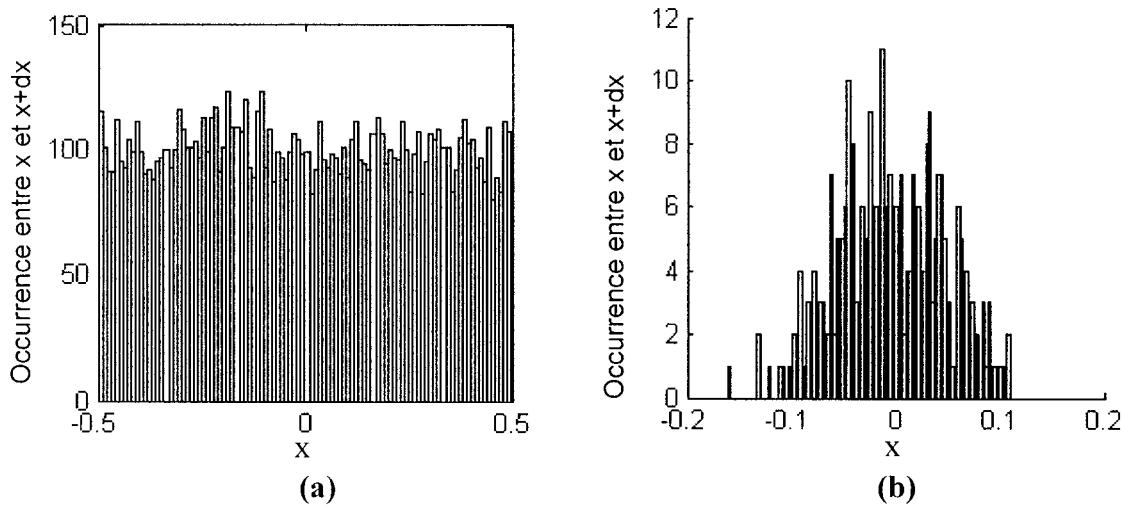


Figure 2.16 (a) Répartition uniforme. (b) Filtre sur 30 points

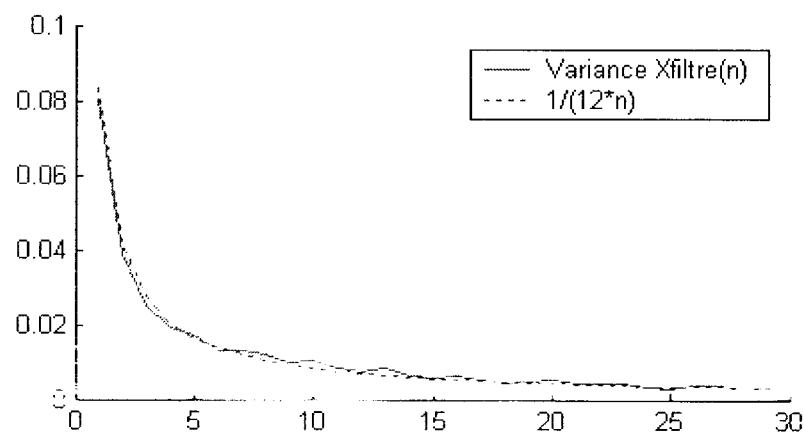


Figure 2.17 Variance de X^{filtre} en fonction du nombre n de points moyennés

2.4.3 Filtre par l'estimation du maximum de vraisemblance

Pour un événement X qui suit une loi de probabilité P , l'estimation du maximum de vraisemblance, a pour but de déterminer quelle est la valeur x_o de X telle que $P(X=x_o)$ soit maximale. Toute la difficulté de l'estimation est que l'expérimentateur, en général, ne connaît pas la loi P et il faut donc l'inférer en fonction d'un échantillon X_i .

La première étape consiste donc à estimer la densité de probabilité de X . Une fois la loi de probabilité de X évaluée, il faut prendre la valeur x_o qui correspond au maximum de cette probabilité.

Méthode statistique pour l'estimation de densité :

Une technique consiste à faire un maillage de l'espace et d'évaluer statistiquement la densité de probabilité dans chaque sous espace R_n du maillage, Figure 2.18. Mathématiquement cela se justifie par le théorème T1, [Duda, Hart, Stork (1998)].

Théorème T 1 :

Soit un échantillon issu d'une densité de probabilité p .

Soit N la taille de l'échantillon. Soit V_N le volume des sous espaces R_n .

Soit k_n le nombre de points de l'échantillon dans le sous espace R_n .

Alors si $x \in R_n$,

$$p_N(x) = \frac{k_n}{V_N} \quad (2.36)$$

et

$$p_N(x) \xrightarrow{N \rightarrow \infty} p(x) \text{ si } \begin{cases} \lim_{N \rightarrow \infty} V_N = 0 \\ \lim_{N \rightarrow \infty} k_n = \infty \\ \lim_{N \rightarrow \infty} \frac{k_n}{N} = 0 \end{cases} \quad (2.37)$$

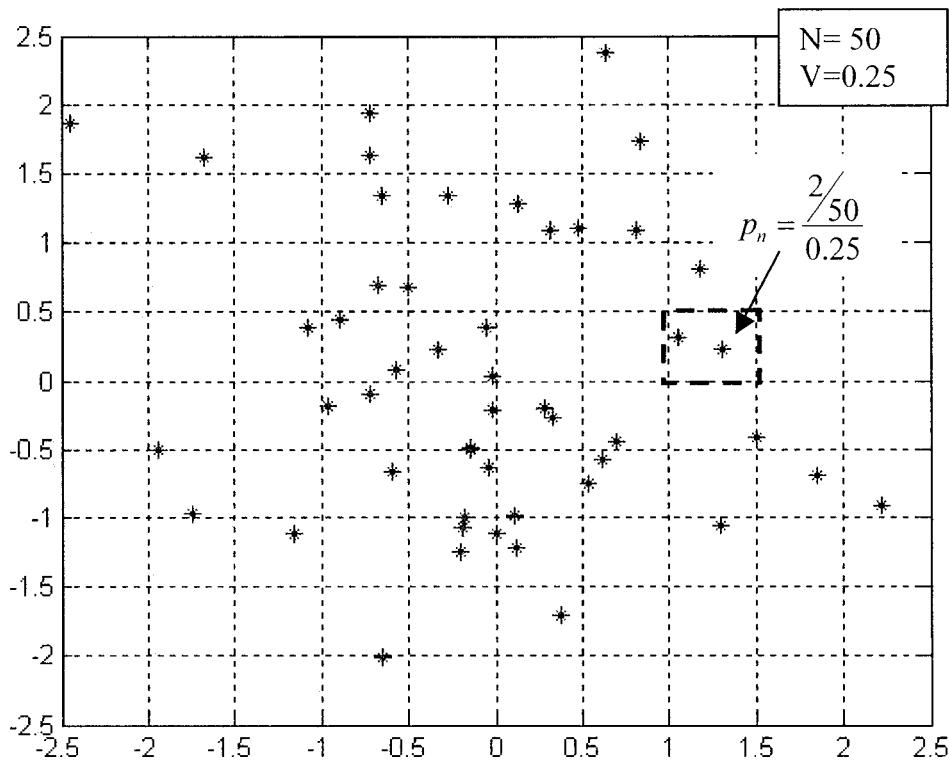


Figure 2.18 Maillage en carrés.

En pratique N est limité, donc :

Si V est trop petit (par rapport à N) alors k_n n'est plus représentatif de la distribution.

Si V est trop grand alors l'estimé devient une sorte d'intégration de la vraie distribution.

Fenêtres de Parzen :

Pour définir les fenêtres de Parzen [Parzen (1962)], nous allons tout d'abord supposer que les sous espaces Rn sont des hyper cubes de largeur h_N et de volume $V_N = h_N^d$.

On définit ensuite les fonctions de fenêtres φ :

$$\left\{ \begin{array}{l} \varphi(x) \geq 0 \\ \int_{espace} \varphi(x) dx = 1 \end{array} \right. \quad (2.38)$$

Ainsi pour une fenêtre de Parzen, le nombre d'échantillons k_n qui se trouvent dans l'hyper cube R_n centré en x est :

$$k_n = \sum_{i=1}^N \varphi\left(\frac{x-x_i}{h_n}\right) \quad (2.39)$$

D'après (2.36) et (2.39) :

$$p_N(x) = \frac{1}{N} \sum_{i=1}^N \frac{1}{V_N} \varphi\left(\frac{x-x_i}{h_N}\right) \quad (2.40)$$

Théorème T 2, [Duda, Hart, Stork (1998)] :

p_N converge vers p si :

$$\begin{cases} \lim_{N \rightarrow \infty} N \cdot V_N = \infty \\ \lim_{N \rightarrow \infty} V_N = 0 \end{cases}$$

Pour illustrer l'étude, considérons un échantillon X issu d'une loi de probabilité normale p . Nous pouvons évaluer p_N pour différentes valeurs de N ($N=1, 10, 100, \infty$) et pour différentes largeurs d'hyper cube $h = \frac{h_l}{\sqrt{N}}$ (avec $h_l=2, 1, 0.5$). Figure 2.19 illustre bien le fait que V doit être optimisé si l'on veut que p_N soit proche de p . En effet quand $N=10$ et $h_l=0.5$ la contribution de chaque échantillon est visible ce qui n'est pas le cas pour $h_l=2$. Quelle que soit la valeur de h_l , quand N devient très grand, p_N converge vers p .

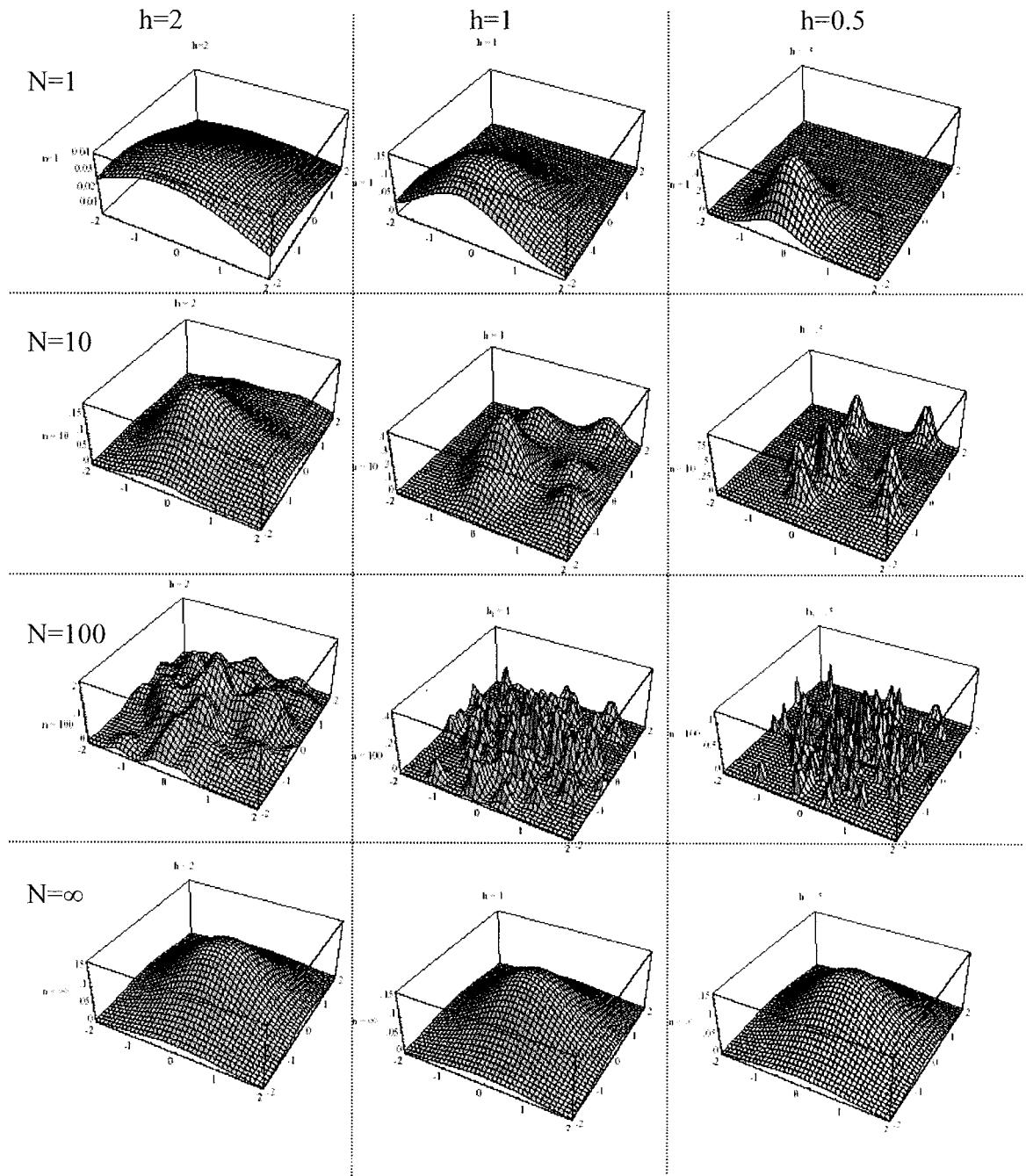


Figure 2.19 p_n pour différentes valeurs de N et de h , [Duda, Hart, Stork (1998)].

Réseaux de fonctions à bases radiales RBFN

Les réseaux de fonctions à bases radiales utilisent les propriétés des fenêtres de Parzen en prenant comme fonction de fenêtre φ des fonctions radiales le plus souvent gaussiennes [Annexe].

Les RBFN, prennent les données d'entrée sous forme de centroïdes de bases radiales et la sortie est la somme de tous les neurones, Figure 2.20.

D'après le Théorème T 2, ces réseaux donnent une approximation de la densité de probabilité de l'échantillon d'entrée.

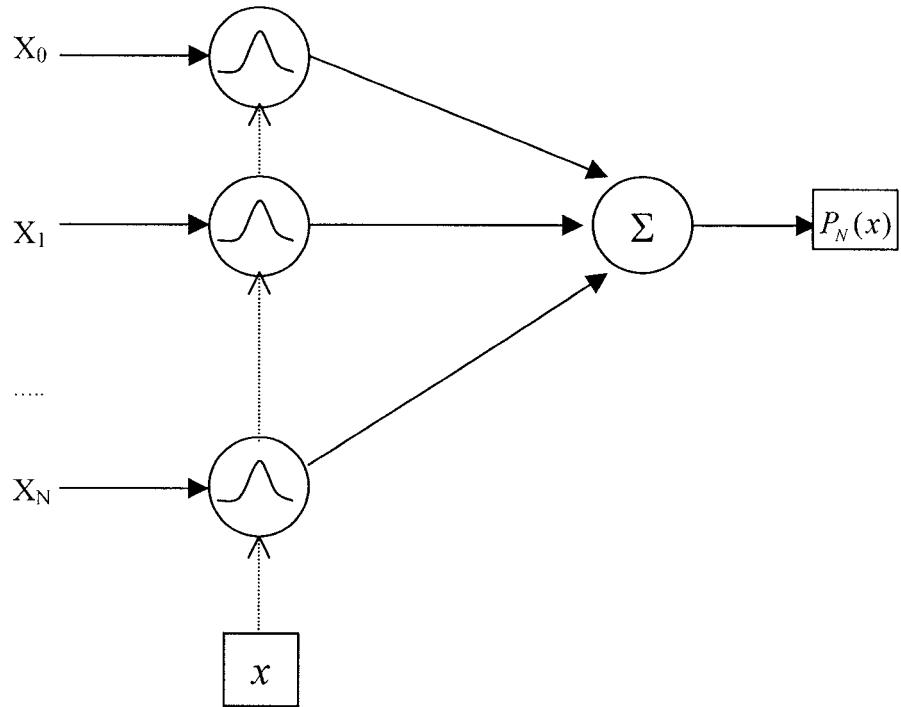


Figure 2.20 RBFN pour l'estimation de densité

Il existe différentes stratégies d'apprentissage suivant que l'on fixe par avance le nombre de centroïdes (neurones) ou que l'on décide de créer autant de centroïdes que données d'entrée. Dans notre étude, les RBFN serviront à filtrer un signal et comme nos données d'entrées n'excèderont jamais la cinquantaine, nous choisirons de créer autant de centroïdes que de données.

Pour illustrer cela, nous allons reprendre l'expérience faite à la page 41. Soit X une variable aléatoire avec une densité de probabilité uniforme entre -0.5 et $+0.5$. Le filtre que l'on réalise consiste à prendre 30 occurrences de X , et d'évaluer le maximum de vraisemblance sur ces 30 points, Figure 2.21(a). Les valeurs ainsi obtenues sont appelées X^{filtre} . Précisons enfin que la variance des gaussiennes des RBFN est fixée arbitrairement à 1. Enfin, nous allons considérer 10 000 occurrences de X que nous allons filtrer par la méthode du maximum de vraisemblance sur 30 points pour obtenir les occurrences de X^{filtre} , Figure 2.21(b).

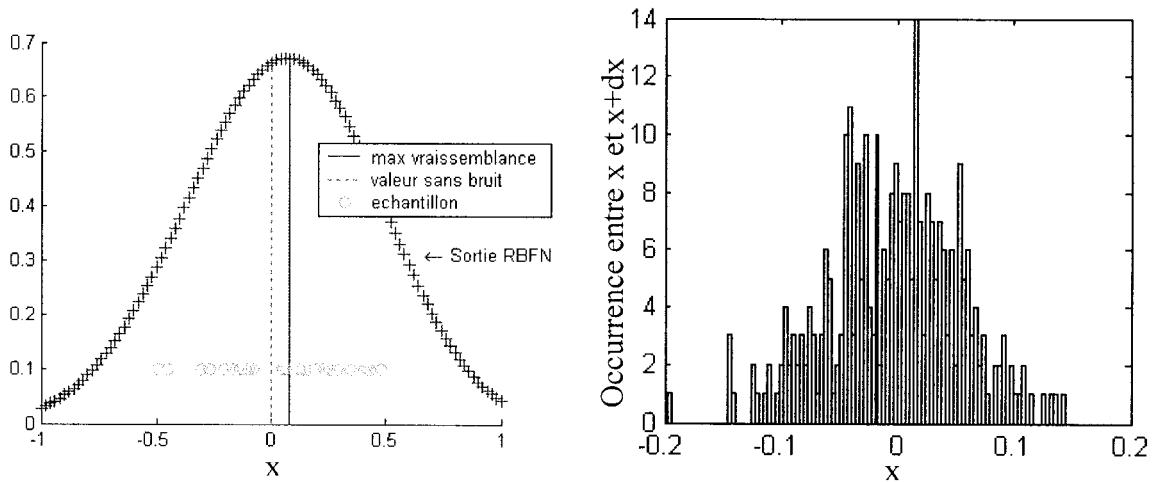


Figure 2.21 (a) Maximum de vraisemblance. (b) Occurrences de X^{filtre}

Nous pouvons étudier l'influence de différents paramètres sur la variance de X^{filtre} . Nous avons tout particulièrement étudié l'influence de la nature de la distribution de X , de la variance h des gaussiennes utilisées dans les RBFN et du nombre de points utilisés dans le filtre.

Simulation 1, Figure 2.22

On utilise 10 000 occurrences de X qui est issu d'une distribution uniforme dans $[-.5,.5]$.

On mesure la variance de X^{Filtre} en fonction du nombre n de points utilisés dans le filtre.

On effectue cette opération pour différentes valeurs h de la variance des gaussiennes du RBFN.

On rappelle que $Var(X)=1/12$.

Simulation 2, Figure 2.23

On utilise 10 000 occurrences de X qui est issu d'une distribution normale $N(0, \sigma^2)$.

On mesure la variance de X^{Filtre} en fonction de σ^2 pour un filtre sur 30 points.

La variance des gaussiennes du RBFN est fixée à 1.

Simulation 3, Figure 2.24

A est issu d'une distribution normale $N(1,0.2)$. B est issu d'une distribution normale $N(2,0.2)$.

On utilise 10 000 occurrences de X qui est constitué de 9500 occurrences de A et 500 occurrences de B réparties aléatoirement.

On mesure les variances de $X^{FiltreMoy}$ calculé avec la méthode de la moyenne sur 30 points et de $X^{FiltreRBF}$ calculé avec la méthode du maximum de vraisemblance sur 30 points en fonction de la variance des gaussiennes du RBFN

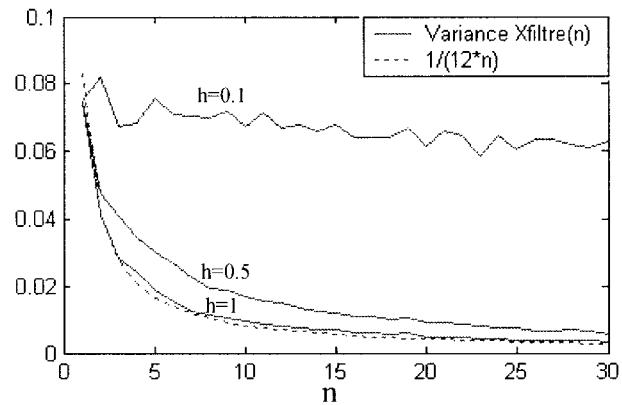


Figure 2.22 Simulation 1

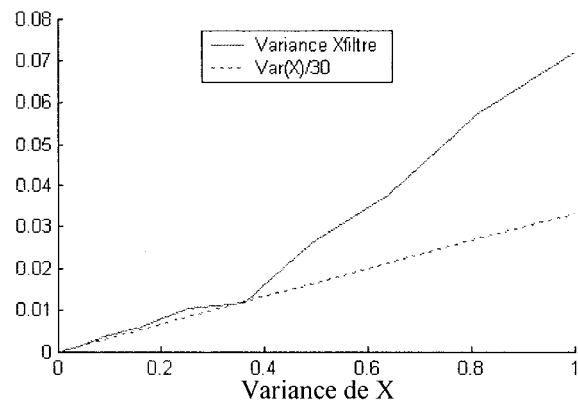


Figure 2.23 Simulation 2

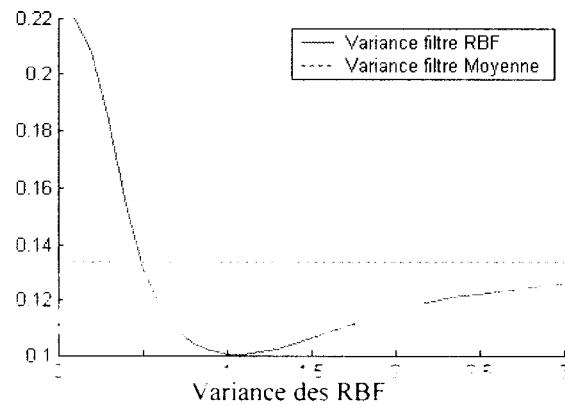


Figure 2.24 Simulation 3

Analyse des résultats :

n est le nombre de points utilisés dans le filtre.

h est la variance des gaussiennes du filtre RBFN.

La variance d'une distribution A est notée $\text{var}(A)$.

- Si les échantillons X_i sont identiquement distribués (simulation 1 et 2) alors quand h est grande, $\text{var}(X^{\text{filtre}})$ tend vers $\text{var}(X^{\text{moyenne}})$ par valeur supérieure.
Valeur typique : quand $h > 5 * \text{var}(X)$, alors $\text{var}(X^{\text{filtre}}) = \text{var}(X)/n$.

Conséquence : si les échantillons X_i sont identiquement distribués, il est préférable d'utiliser un filtre par moyenne car on n'a pas le soucis du choix de h.

- Si les échantillons X_i ne sont pas identiquement distribués (simulation 3) alors il existe une variance h optimale pour laquelle $\text{var}(X^{\text{filtre}})$ est inférieure à $\text{var}(X^{\text{moyenne}})$.

Conséquence : si les échantillons X_i ne sont pas identiquement distribués, il est préférable d'utiliser un filtre RBFN avec une variance h optimale

Nous venons de présenter et d'analyser les différents outils disponibles dans la littérature pour mener à bien le projet. Au Chapitre 3 nous allons décrire comment nous avons utilisé ces outils en expliquant la démarche que nous avons utilisée pour atteindre nos objectifs.

CHAPITRE III

CONCEPTION

La conception de notre système de DOA a été motivée par une recherche continue de simplicité. La simplicité pour l'architecture du réseau d'antennes, se caractérise par l'utilisation du minimum d'antennes ce qui diminue le couplage entre chacune d'elles et donc favorise un comportement conforme aux hypothèses utilisées dans les modèles théoriques de réseaux d'antennes. La simplicité dans le combinateur a pour but conserver la puissance des signaux reçue le plus possible et donc de fournir au réseau de neurones des signaux avec un meilleur rapport signal sur bruit. Enfin la simplicité dans le réseau de neurones signifie un réseau avec le moins de neurones possible ce qui permet un apprentissage rapide et fiable ainsi qu'une implantation matérielle suscitant le moins de ressources.

Nous allons présenter dans ce chapitre la démarche que l'on a suivie pour atteindre nos objectifs et nous verrons comment notre recherche a abouti à l'élaboration de deux prototypes de système de DOA. Nous présentons les étapes de la conception dans l'ordre chronologique afin de faire ressortir la méthodologie utilisée. Premièrement, la section 3.1 présente la conception et le choix des réseaux d'antennes et des combinateurs de signaux associés. La section 3.2 constitue le cœur du projet puisqu'elle expose comment ont été conçus et entraînés les réseaux de neurones qui en finalité effectuent le travail de régression. Enfin la section 3.3 décrit comment nous avons abordé le problème du bruit inhérent aux signaux reçus. L'organigramme de la Figure 3.1 explique comment les différentes sections interagissent dans l'ensemble du projet.

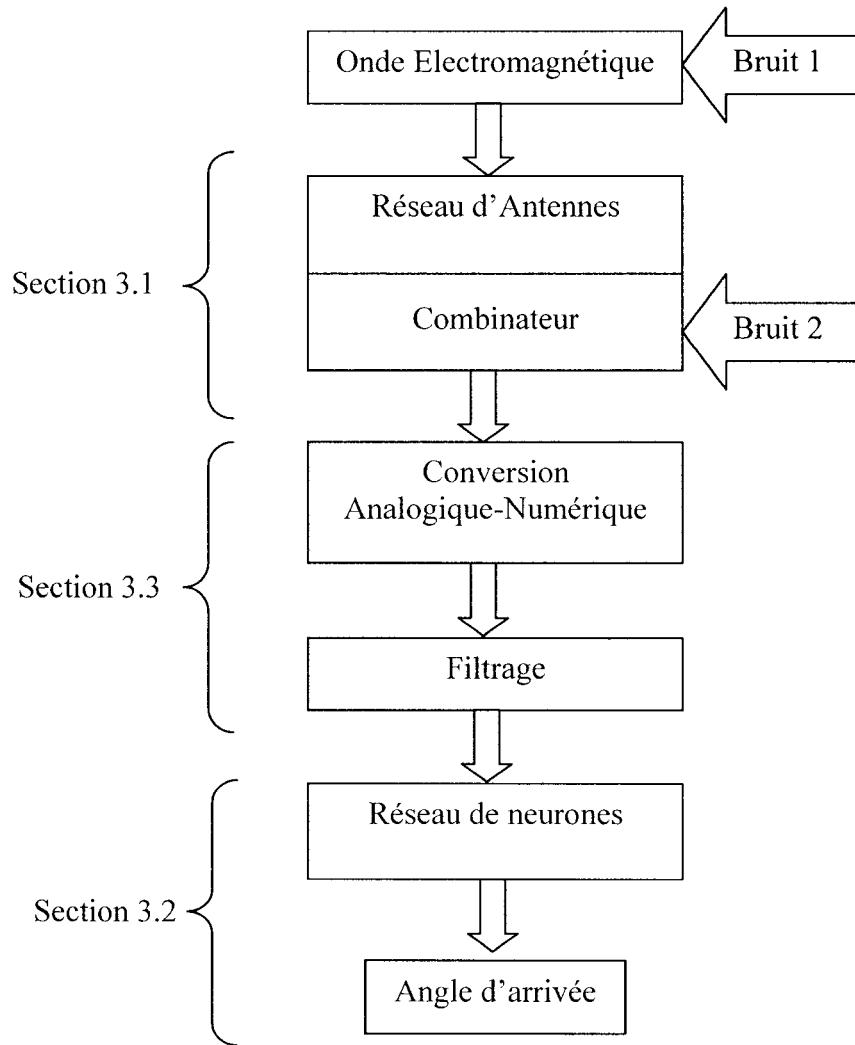


Figure 3.1 Organisation du chapitre 3

3.1 Réseaux d'antennes et combinateurs : une approche qualitative

Une des originalités de ce projet réside dans le fait que nous avons toujours fait nos choix en se posant une question : « qu'est ce qui nous permettra au final de donner la meilleure réponse ? » Bien que le projet puisse être séparé en différentes unités (Figure 3.1), nous avons néanmoins toujours essayé de garder une vision globale. Pour réaliser et optimiser le réseau d'antennes et le combinateur nous avons utilisé une approche qualitative qui reposait sur un questionnement : « quels sont a priori les signaux à fournir au réseau de neurones pour faire la régression la plus précise ? ». Nous avons travaillé en étroite collaboration avec le laboratoire PolyGrames, Groupe de Recherches Avancées en Micro-ondes et Électronique Spatiale de l'École Polytechnique qui est l'initiateur du projet et qui s'est chargé tout particulièrement de l'étude liée aux antennes : choix de l'antenne, réalisation de prototypes, simulation du réseau d'antennes comme décrit dans [Fonseca (03)].

Les seules orientations de départ fournies par le PolyGrames sont :

- Le réseau d'antennes est constitué de deux axes orthogonaux car cette géométrie bien connue et étudiée présente des qualités en terme de couplage.
- Le combinateur est fait à partir de coupleurs hybrides.
- L'objectif est que le système doit fournir une réponse à un degré près pour toute onde incidente située dans un espace délimité par un cône d'ouverture (demi-angle au sommet) de 45° .

3.1.1 Choix du cône d'arrivée

Bien que l'ouverture du cône soit prédéfini à 45° , il y a tout de même une infinité de façon de choisir son cône par rapport au réseau d'antennes, Figure 3.2.

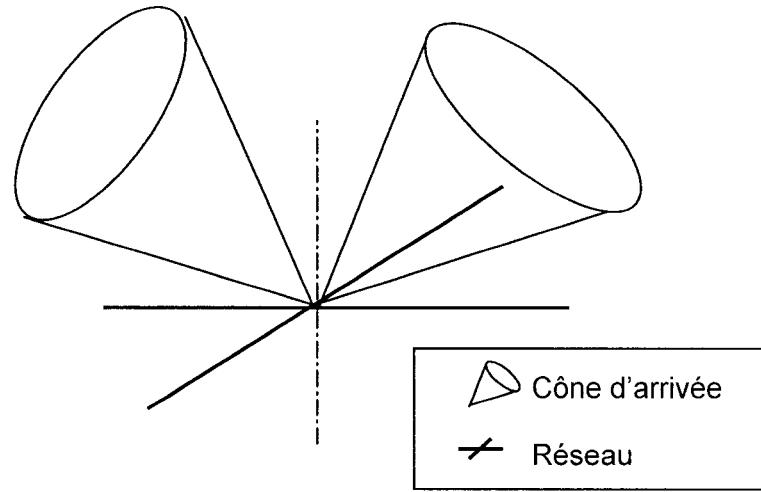


Figure 3.2 Différents cônes de direction d'arrivée

Pour simplifier les notations et pour respecter la symétrie du système, le système de coordonnées utilisé repère une direction au moyen des angles (θ_x, θ_y) formés avec les deux axes du réseau d'antennes comme indiqué sur la Figure 3.3.

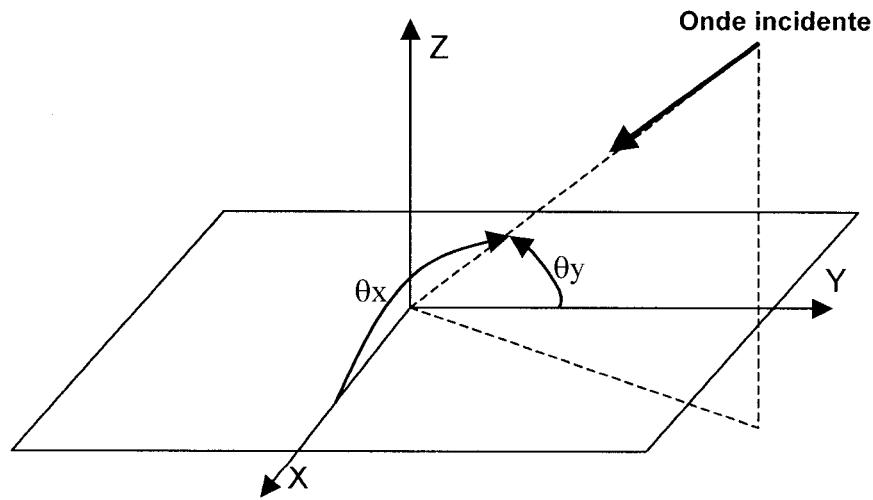


Figure 3.3 Système de coordonnées

Afin de découvrir l'intervalle en (θ_x, θ_y) optimal, nous repartons de la forme générale des signaux provenant du combinateur tels que nous les avons définis au Chapitre 1 :

$$X_i(\theta) = 1 + \cos(\varphi + \frac{2\pi * d_i}{\lambda} \cos(\theta_i)) \quad (3.1)$$

où

θ_i est l'angle d'arrivée par rapport à l'axe i ($= x$ ou y)

d_i est une distance entre deux antennes sur l'axe i ($= x$ ou y)

φ est une phase éventuellement introduite dans le combinateur

λ est la longueur d'onde

Comme il y a symétrie entre les axes, nous conduisons cette étude pour un axe seulement et pour éclaircir les notations nous remplacerons θ_i par θ . Nous savons que les signaux vont être utilisés pour retrouver l'angle θ d'arrivée. Mathématiquement, nous allons donc inverser la fonction $X(\theta)$. Théoriquement, pour que la fonction inverse soit la plus précise possible, c'est-à-dire qu'une petite variation de X entraîne une grande variation de θ , il faut que la dérivée de $X(\theta)$ soit maximum. Nous allons donc calculer les variations de $X(\theta)$ et en déduire l'intervalle dans lequel ces variations sont maximales, Figure 3.4.

Mathématiquement, nous recherchons θ_0 tel que :

$$\theta_0 = \arg \max_{\varphi} \int_{\varphi - \frac{\pi}{4}}^{\varphi + \frac{\pi}{4}} \left| \frac{\partial X(\theta)}{\partial \theta} \right| d\theta \quad (3.2)$$

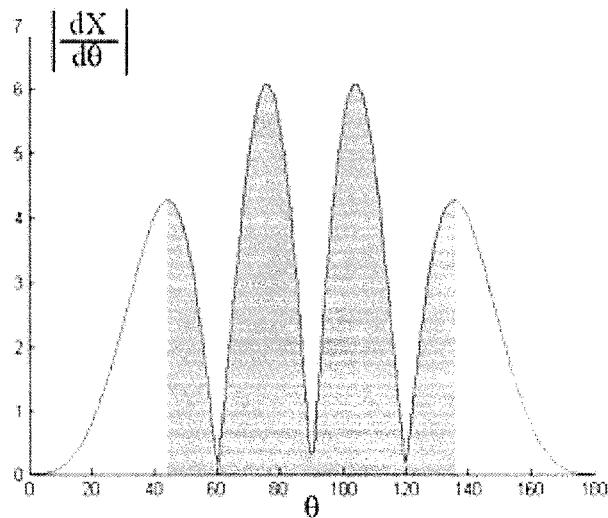


Figure 3.4 $\left|\frac{\partial X}{\partial \theta}\right|$ sur l'intervalle [0,180]

Graphiquement, on vérifie que l'intervalle optimal est [45, 135], Figure 3.5.

Dans toute la suite du projet, la zone d'intérêt pour les directions d'arrivée sera donc le cône vertical (par rapport au plan des antennes) avec une ouverture de $\frac{135 - 45}{2} = 45^\circ$.

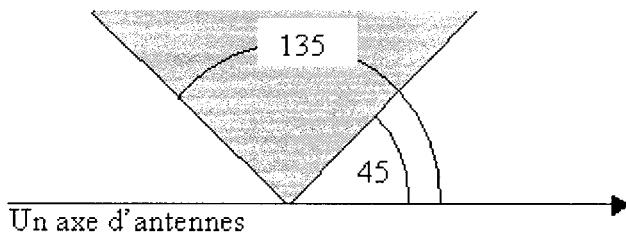


Figure 3.5 Intervalle optimal

3.1.2 Premier prototype

Le principe

Le point de départ pour optimiser les positions des antennes sur l'axe est encore une fois l'équation (3.1). Cependant, contrairement à l'étude faite à la section 2.2.1, nous allons utiliser une approche qualitative pour définir les paramètres d et φ de l'équation (3.1).

Notre approche est une réflexion sur : « quelle serait la forme des signaux à fournir à un réseau de neurones pour effectuer une régression dans les meilleures conditions ? »

Robustesse au bruit

Pour que la régression soit utilisable, il faut que le système fournit une réponse en présence de bruit. Il faudrait donc créer des signaux qui portent une information sur la direction d'arrivée qui ne soient pas très précise mais très fiable. Par exemple, une dichotomie de l'intervalle $[45,135]$ serait efficace pour remplir cette tâche. La Figure 3.6 est un exemple de signaux que l'on pourrait utiliser pour diviser l'intervalle $[45,135]$ en quatre sous-intervalles, dans toute la suite nous les appellerons signal 1 et signal 2.

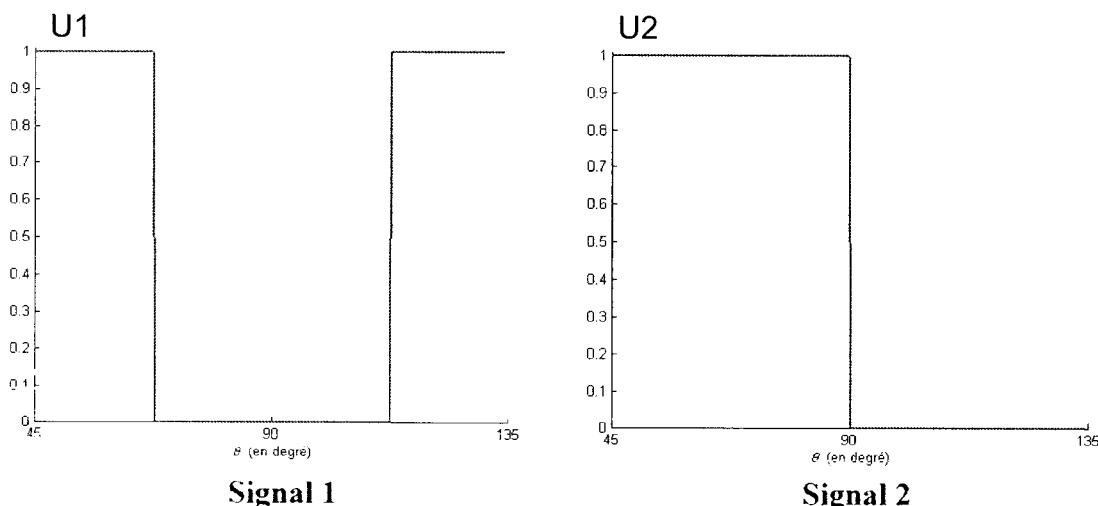


Figure 3.6 Signaux réalisant la division de l'intervalle en quatre sous-intervalles

Précision de la réponse

Une fois que l'intervalle d'étude est divisé en quatre sous-intervalles, il faut disposer de signaux qui varient beaucoup sur ces sous-intervalles. En effet, pour inverser une fonction il est préférable que la dérivée de X par rapport à θ soit grande. La Figure 3.7 montre deux signaux qui remplissent cette dernière condition. La partie sombre représente un des quatre sous-intervalles. Le signal 3 assure une variation maximale et le signal 4 assure un niveau de puissance supérieur à 0.5 sur le sous-intervalle.

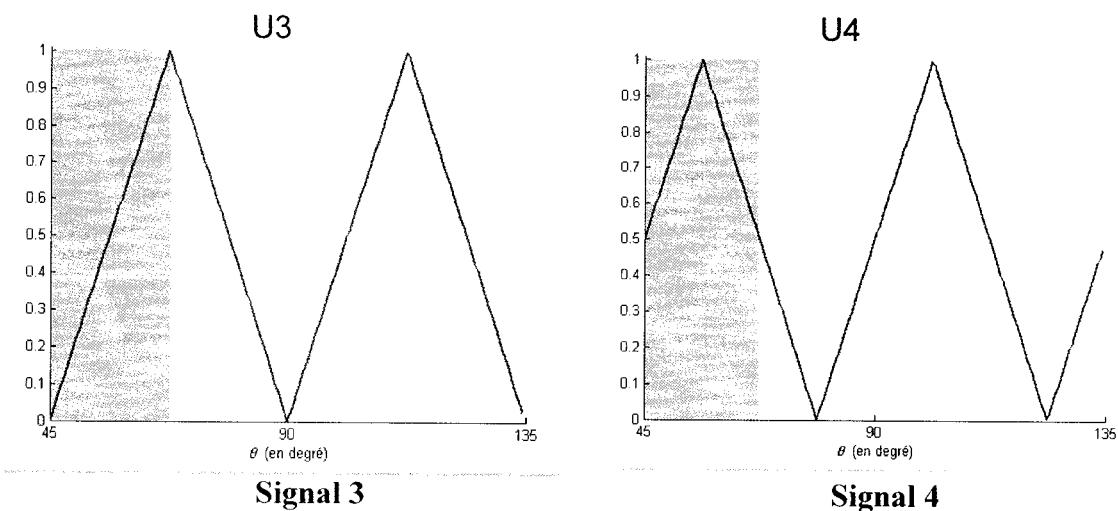


Figure 3.7 Signaux optimum pour une régression sur un sous-intervalle

La réalisation

Une fois que nous avons fixé les formes des signaux à atteindre il faut optimiser les valeurs des paramètres J et ϕ de l'équation (3.1) pour élaborer notre système. Pour chacun des signaux il faut minimiser la fonction f_i correspondante représentée à la Figure 3.8 et décrite par les équations (3.3) à (3.5).

$$f_1(d, \varphi) = - \int_{\frac{\pi}{4}}^{\frac{3\pi}{4}} |X(\theta) - u_1(\theta)| d\theta \quad f_2(d, \varphi) = - \int_{\frac{\pi}{4}}^{\frac{3\pi}{4}} |X(\theta) - u_2(\theta)| d\theta \quad (3.3)$$

$$f_3(d, \varphi) = \sum_{i=1}^4 \left| X\left(\frac{\pi}{4} + i \frac{\pi}{8}\right) - X\left(\frac{\pi}{4} + (i-1) \frac{\pi}{8}\right) \right| \quad (3.4)$$

$$f_4(d, \varphi) = \sum_{i=1}^4 \left| X\left(\frac{3\pi}{8} + i \frac{\pi}{8}\right) - X\left(\frac{3\pi}{8} + (i-1) \frac{\pi}{8}\right) \right| \quad (3.5)$$

où

u_1 (resp. u_2) est la fonction caractéristique du signal 1(resp. 2) désiré.

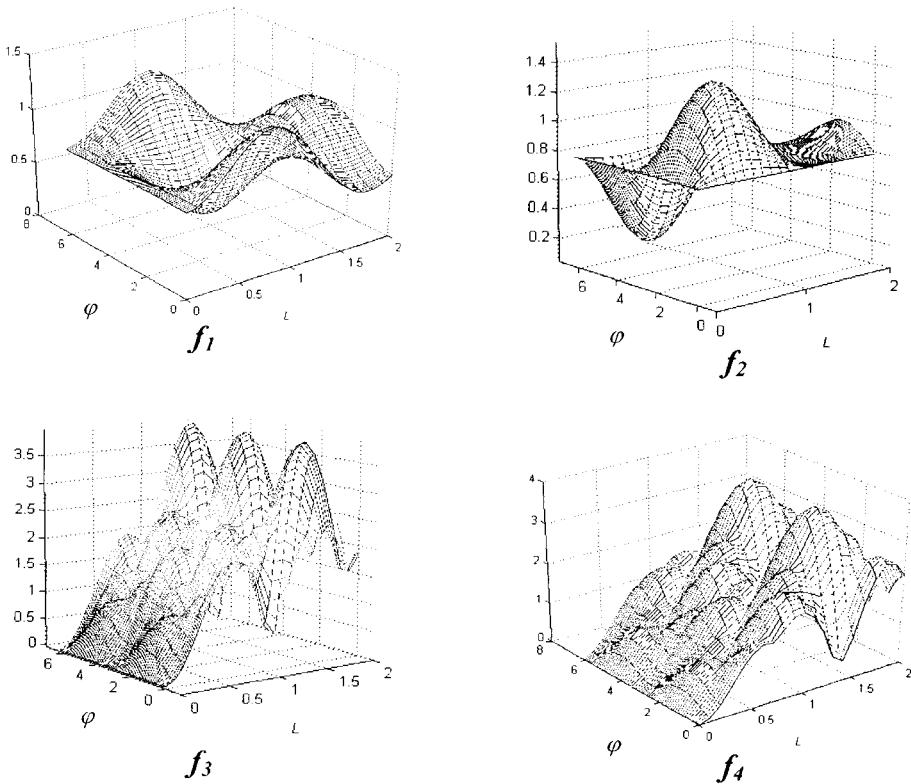
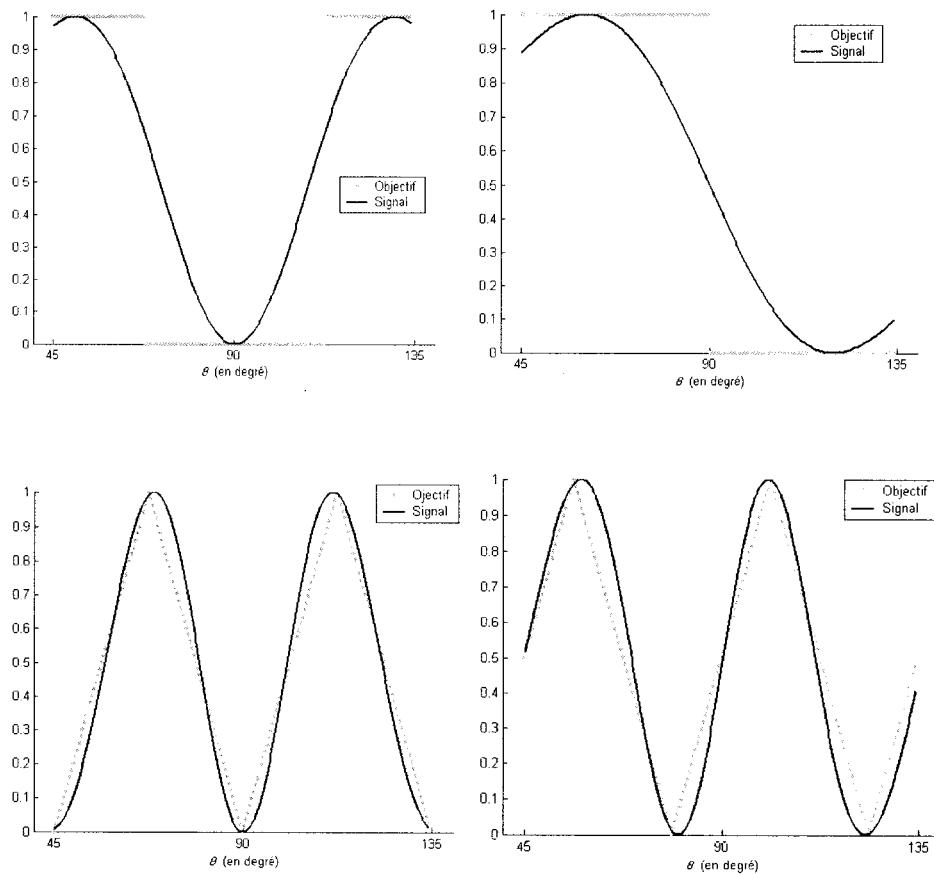


Figure 3.8 Variation des fonctions $f_i(L, \varphi)$

Le Tableau 3.1 rassemble les valeurs optimales des paramètres. La Figure 3.9 représente les quatre signaux effectivement obtenus avec les valeurs correspondantes de d et φ .

Tableau 3.1

Signaux	1	2	3	4
d (multiple de λ)	0.79	0.50	1.37	1.37
φ (en radians)	π	$-\frac{\pi}{2}$	π	$\frac{\pi}{2}$

**Figure 3.9 Signaux obtenus avec les valeurs optimales**

Si l'on veut réaliser ces quatre signaux, il faudrait a priori 4 antennes qui pourraient être disposées conformément à la Figure 3.10. Cependant, on remarque que la distance séparant les antennes A3 et A4 est de 0.58λ , ce qui est proche des 0.50λ utilisés pour le signal 2. De plus la simulation de la Figure 3.11 montre que le signal obtenu en prenant 0.58λ est très proche de celui obtenu avec 0.50λ .

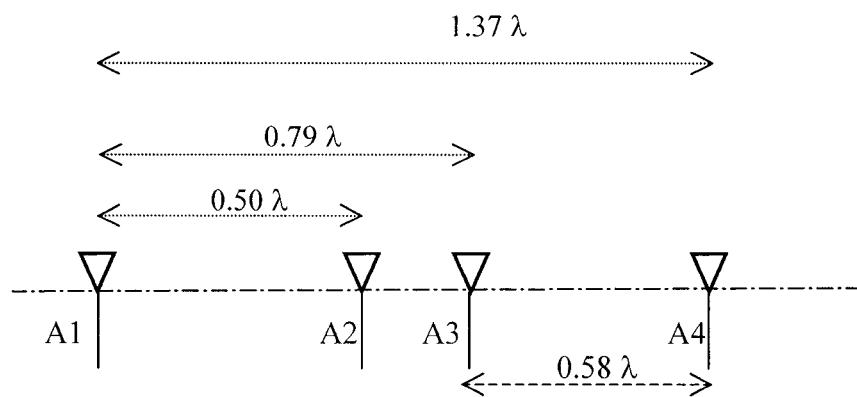


Figure 3.10 Réseau de quatre antennes avec les trois valeurs de d

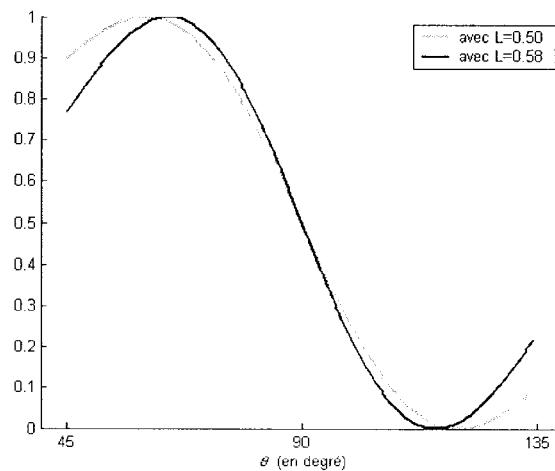


Figure 3.11 Signal 2 avec $d=0.58\lambda$ ou 0.50λ

Cette dernière considération nous a permis de supprimer une antenne tout en produisant les quatre signaux désirés. La topologie finale du réseau et du combinateur est présentée à la Figure 3.12. Notre système se compose donc de trois antennes et de 4 coupleurs hybrides (section 2.2) qui génèrent les signaux X_i avec le déphasage désiré.

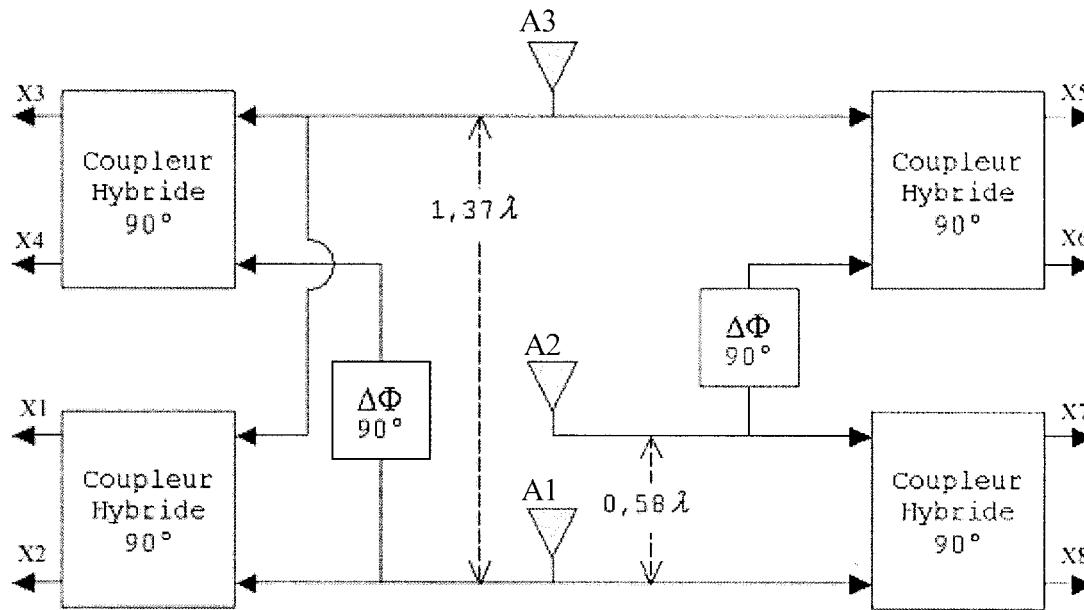


Figure 3.12 Réseau d'antennes et combinateur du prototype 1

Il est intéressant de noter que l'architecture du réseau d'antennes que nous avons obtenue par des considérations qualitatives est très proche de celle que nous nous avions obtenue à la section 2.1 en suivant la méthode proposée par [Gavish (1995)], Figure 3.13. Cependant, en introduisant une dissymétrie dans le réseau d'antennes, le prototype 1 augmente la diversité des informations.

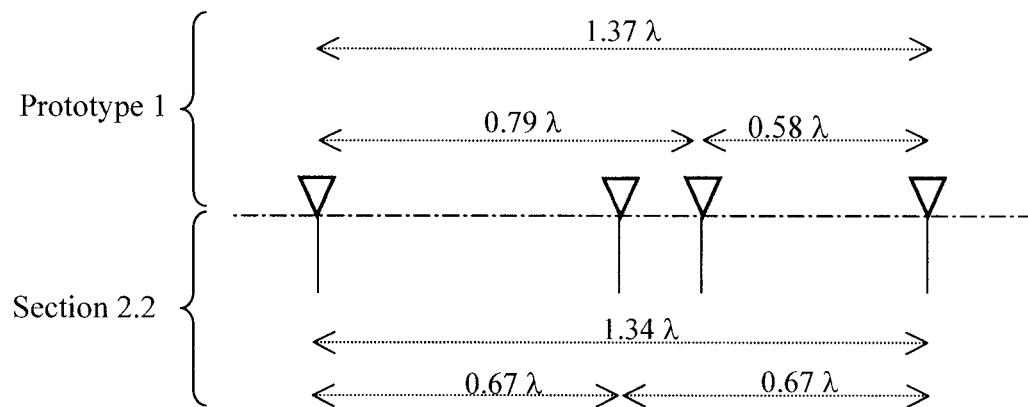


Figure 3.13 Comparaison des architectures suivant la méthode utilisée

On remarque que le système génère huit signaux, alors que jusqu'à présent nous n'avions mentionné que quatre signaux. Cela s'explique par le fait que les coupleurs hybrides produisent deux signaux déphasés l'un par rapport à l'autre de $\pi/2$. Ainsi notre système produit les quatre signaux désirés (X_2, X_4, X_6, X_7) et quatre signaux supplémentaires décalés (X_1, X_3, X_5, X_8).

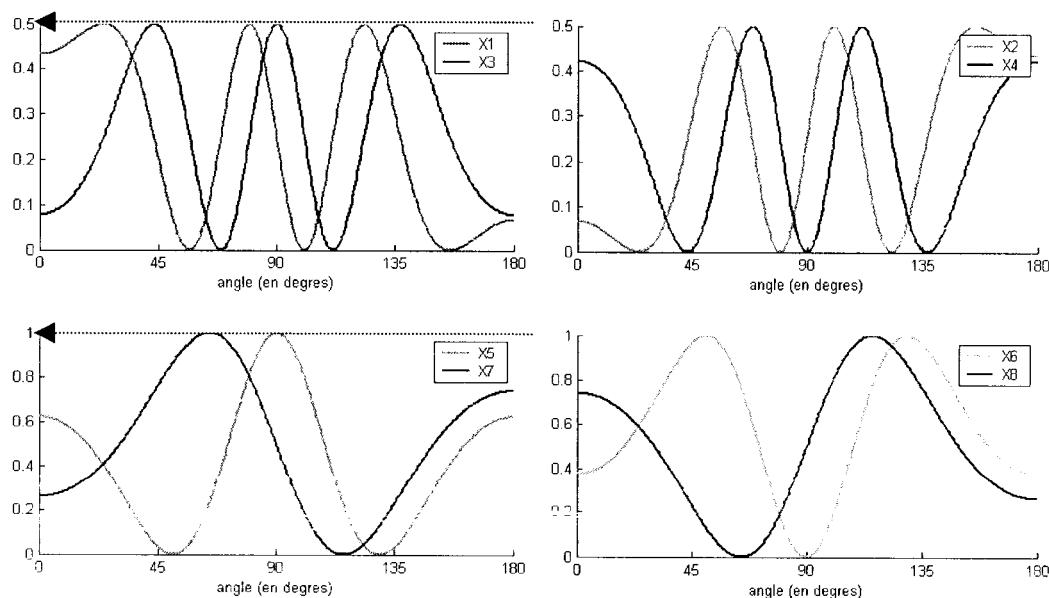


Figure 3.14 Signaux de sortie du prototype 1

La Figure 3.14, présente les huit sorties du prototype 1. On peut remarquer sur cette figure que les sorties X₁, X₂, X₃ et X₄ ont un niveau de puissance inférieur à celui des signaux X₅, X₆, X₇, X₈. Ceci s'explique par le fait que ces signaux sont divisés une fois de plus dans le combinateur comme on peut le voir sur la Figure 3.12.

Ce premier prototype a donc été conçu avec deux lignes directrices : la première est que tout le système est optimisé en se préoccupant de la finalité, qui est de retrouver l'angle θ à partir de $X(\theta)$, et la deuxième est que le système doit être robuste à la présence de bruit et c'est pour cela que nous avons intégré une redondance d'information.

3.1.3 Deuxième prototype

Le principe

Comme les signaux X₁, X₂, X₃ et X₄ du prototype 1 ont un niveau de puissance inférieur aux quatre autres, ils sont plus sensibles à la présence de bruit. D'autre part, le minimum théorique pour repérer la direction d'arrivée est deux antennes et la présence de la troisième antenne augmente le couplage du réseau et donc diminue la précision de chaque signal. Il serait donc intéressant de considérer un autre système qui aurait seulement deux antennes et aurait moins de signaux, mais avec des niveaux de puissance plus élevés et plus homogènes. C'est avec cette dernière considération que nous avons conçu le prototype 2. Nelson Fonseca, [Fonseca (03)], a proposé d'optimiser une architecture de combinateur basée sur les matrices de Blass. Comme pour le prototype 1 ce sont des considérations qualitatives qui ont guidé notre étude. Nous voulons à présent réaliser la régression à partir de 3 signaux issus de 2 antennes. Qualitativement il faudrait former deux signaux de « dichotomie » et un signal de « précision » comme illustré sur la Figure 3.15

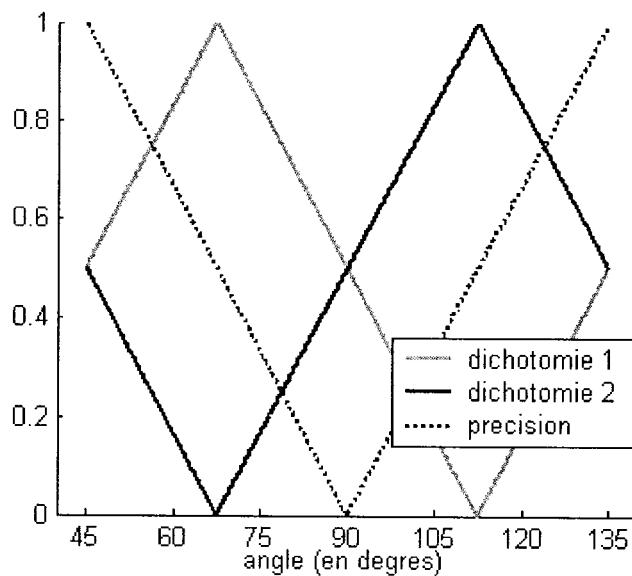


Figure 3.15 Signaux d'objectif du prototype 2

La Réalisation

Afin de générer les 3 signaux d'objectif, il faut chercher les valeurs optimales de la distance entre les deux antennes d , des déphasages φ_1 et φ_2 ainsi que des gains C_1 et C_2 introduits dans les coupleurs hybrides, Figure 3.16. Jusqu'à présent les coupleurs hybrides que nous utilisions n'avait pas de dissymétrie de gain, mais cette particularité nous permet à présent de diviser les puissances des antennes de telle sorte que chaque signal aux portes du combinateur ait une puissance équivalente.

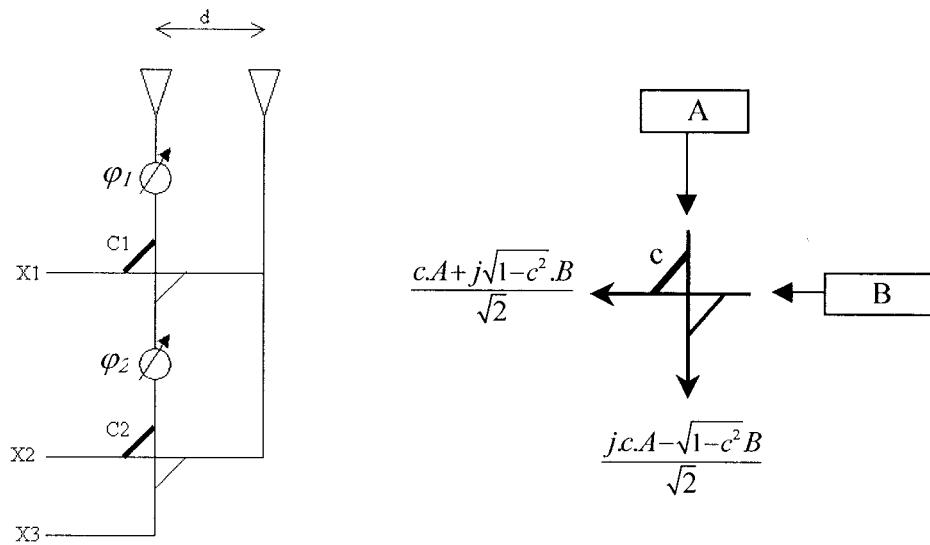


Figure 3.16 Géométrie à priori du réseau et gain d'un coupleur hybride

Les valeurs optimales des paramètres, [Fonseca (03)], sont consignées dans le Tableau 3.2 et le prototype 2 est schématisé à la Figure 3.17.

Tableau 3.2

d	φ_1 (degrés)	φ_2 (degrés)	C₁	C₂
0.58 λ	30	90	1/√3	1/√2

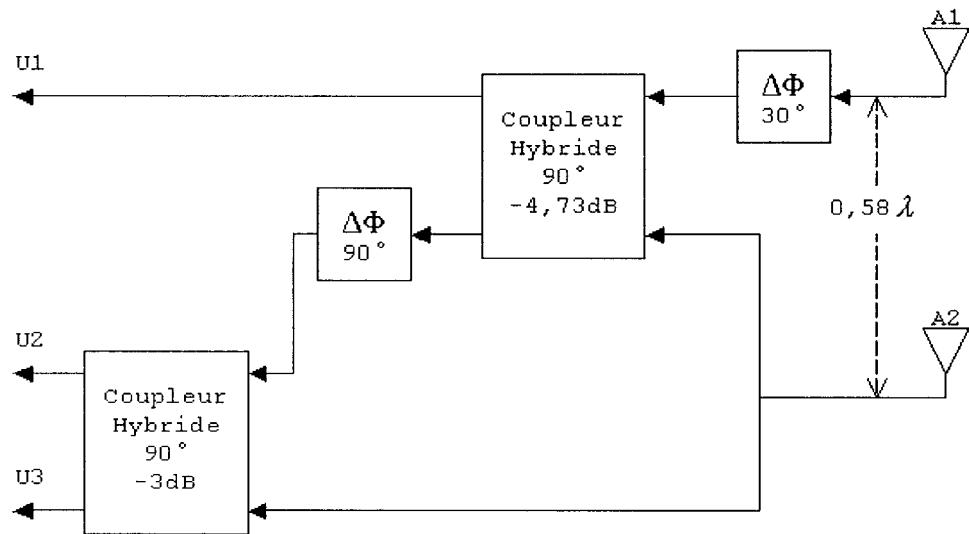


Figure 3.17 Réseau d'antennes et combinateur pour le prototype 2

La Figure 3.18 présentent les signaux U_1 , U_2 et U_3 issus du combinateur. On peut remarquer que les signaux obtenus ont tous le même niveau de puissance et que cette puissance est plus élevée que pour les signaux du prototype 1.

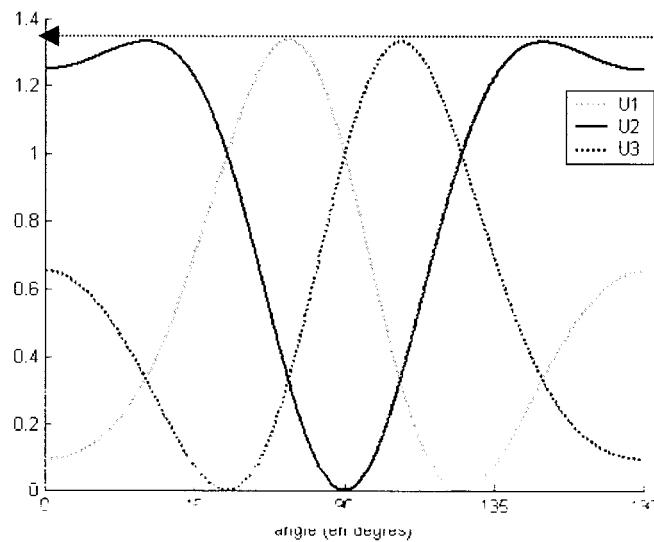


Figure 3.18 Signaux du prototype 2

Les deux prototypes sont donc le fruit d'une même démarche qui consiste à optimiser les paramètres du système à partir de considérations qualitatives sur la forme des signaux à fournir au réseau de neurones. En revanche, les différences entre le prototype 1 et le prototype 2 proviennent de la manière dont on compte s'affranchir des incertitudes liées au bruit : par une redondance d'informations pour le prototype 1 et par la recherche de signaux homogènes et robustes pour le prototype 2.

3.2 Optimisation des réseaux de neurones

Le point de départ pour l'élaboration du réseau de neurones a été le choix du type de réseau que nous allions utiliser. C'est le perceptron multicouche, MLP, qui a été retenu pour deux raisons: premièrement le MLP est un régresseur universel, [Haykin, chap4], et donc il est théoriquement capable d'inverser la fonction $X(\theta)$ si celle-ci est inversible. Deuxièmement, les MLP, une fois entraînés, ont la particularité de donner une réponse très rapidement car les calculs nécessaires sont majoritairement des additions et des multiplications ce qui est très facilement programmable sur des microprocesseurs de type DSP. Ainsi les MLP semblent être *a priori* tout à fait adaptés à notre projet.

3.2.1 Résolution à une ou deux dimensions

La résolution du problème de direction d'arrivée a été menée jusqu'à présent en séparant le problème en deux : la recherche de θ_x et la recherche de θ_y , Figure 3.3. Pour l'élaboration du réseau de neurones, nous pouvons garder cette séparation en deux ou aborder le problème de façon globale.

Méthode 1D

Dans la méthode 1D, nous traitons les données provenant de chaque axe de façon indépendante. Nous entraînerons donc un réseau pour chaque axe et il donnera une réponse à une dimension correspondant à l'angle d'arrivée par rapport à son axe. La réponse finale est le couple (θ_x, θ_y) provenant des deux axes de façon indépendante.

Méthode 2D

Dans la méthode 2D, nous traitons ensemble les données provenant des deux axes. Nous entraînerons donc un seul réseau qui prend en entrée les signaux des deux axes et donnera en sortie le couple (θ_x, θ_y) correspondant.

Normalisation

Pour les deux méthodes, les signaux que l'on va fournir aux réseaux de neurones ne sont pas les sorties du combinateur telles quelles. En effet, les signaux de puissance issus du combinateur vont passer à travers un convertisseur analogique numérique, Figure 3.1, et donc le réseau de neurones ne traitera que des données numériques. Pour convertir un signal analogique en un signal numérique sur n bits, il faut se fixer une valeur de référence i. e. la constante de normalisation qui fixe la plus grande valeur binaire représentable. Dans le cas présent, nous prenons le plus grand des signaux comme référence. Mathématiquement, les entrées disponibles pour le réseau de neurones ont donc la forme décrite par l'équation (3.6). Bien entendu, dépendamment de l'angle d'arrivée, le signal qui sert de référence est différent.

$$[X_1, \dots, X_k]_{\text{numérique}} = \frac{[X_1, \dots, X_k]_{\text{combinateur}}}{\max_{i=1..k} X_i_{\text{combinateur}}} \quad (3.6)$$

où

k est le nombre d'entrées pour le réseau de neurones. Les valeurs de k sont répertoriées dans le Tableau 3.3.

Tableau 3.3

	Méthode 1D	Méthode 2D
Prototype 1	8	16
Prototype 2	3	6

Les Figure 3.19 et Figure 3.20 représentent les signaux de l'axe X après normalisation pour la méthode 1D en fonction de θ_x pour les prototypes 1 et 2. On peut remarquer que la normalisation due à la conversion analogique numérique a pour effet d'accentuer la concordance des signaux avec notre objectif de dichotomie de l'intervalle [45,135].

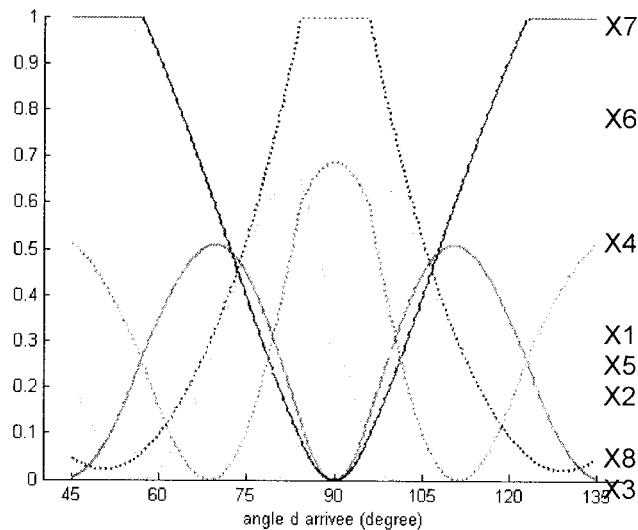


Figure 3.19 Entrées normalisées pour la méthode 1D pour le prototype 1

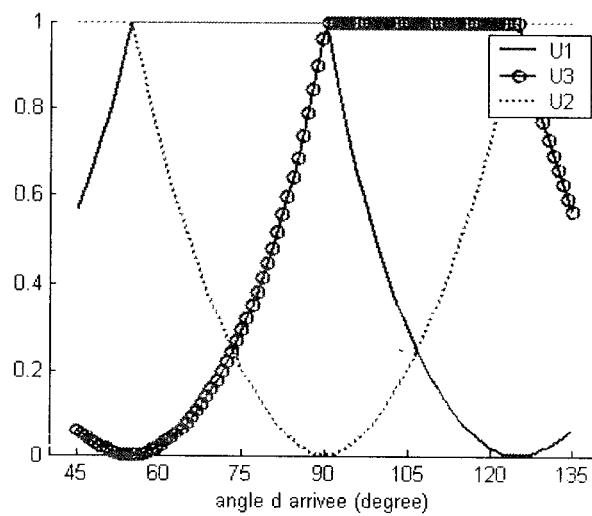


Figure 3.20 Entrées normalisées pour la méthode 1D pour le prototype 2

Si les antennes étaient totalement isotropes, les signaux de l'axe X (respectivement Y) seraient totalement indépendants de θ_y (respectivement θ_x) et la méthode 2D serait une pure redondance de la méthode 1D. Dans la réalité, les antennes ont un gain spatial qui dépend de θ_x et de θ_y et donc, par la méthode 2D, on peut se servir de ce gain comme source d'information pour la direction d'arrivée. Avec un vrai réseau d'antennes, la méthode 2D devrait donc avoir de meilleurs résultats. Cependant, on peut objecter plusieurs points :

- Pour la méthode 2D l'entraînement doit se faire sur n^2 points au lieu de $2n$ points pour la méthode 1D.
- Plus les réseaux sont grands, plus l'optimisation des poids synaptiques est délicate.
- Comme le gain n'est pas uniforme, les signaux portent une information bidimensionnelle. Cependant, après la conversion analogique-numérique, les signaux perdent une partie de cette information. En effet nous avons simulé les sorties de l'axe X des prototypes 1 et 2 en fonction de (θ_x, θ_y) en attribuant des gains non uniformes aux antennes et nous avons pu constater que la dépendance en θ_y diminue après la normalisation. La Figure 3.21 représente la sortie X4 du prototype 1 avant normalisation, puis en normalisant pour la méthode 2D, et enfin en normalisant pour la méthode 1D.

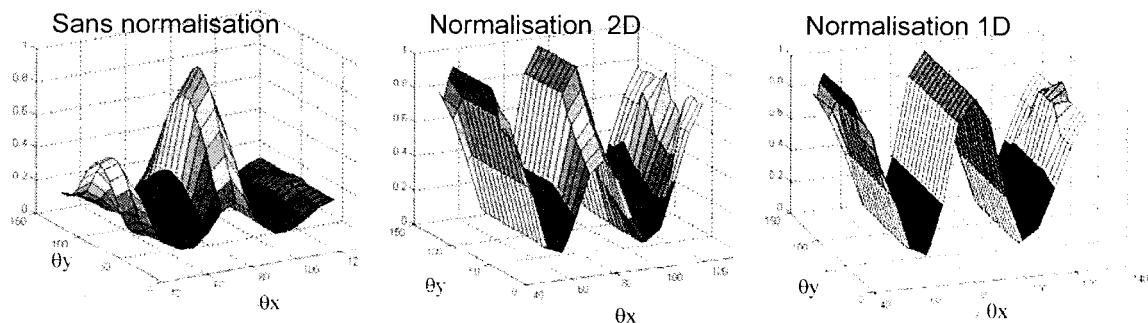


Figure 3.21 Effet de la normalisation sur les signaux

A priori, les deux méthodes présentent des avantages et des inconvénients. Pour la suite nous développerons en parallèle ces deux méthodes avant de pouvoir décider laquelle est la meilleure.

3.2.2 Caractéristiques générales

Comme les entrées du réseau de neurones sont les signaux numérisés provenant du combinateur, la forme des entrées est, dépendamment du prototype et de la méthode, un vecteur normalisé de 8, 3,16 ou 6 signaux. Comme le réseau de neurones doit donner en réponse l'angle d'arrivée, la valeur des sorties est comprise entre $\pi/4$ et $3\pi/4$. De par la nature 2π -périodique de la sortie, il faut également que la fonction coût, qui mesure l'erreur de sortie, soit 2π -périodique. La fonction coût doit être nulle en 0 et 2π et doit être maximale en π , Figure 3.22

Pour un échantillon $\{x_i\}_{i=1\dots n}$ et un objectif $\{t_i\}_{i=1\dots n}$, notre fonction coût L est :

$$L = \frac{1}{n} \sum_{i=1}^n \sin^2\left(\frac{x_i - t_i}{2}\right) \quad (3.7)$$

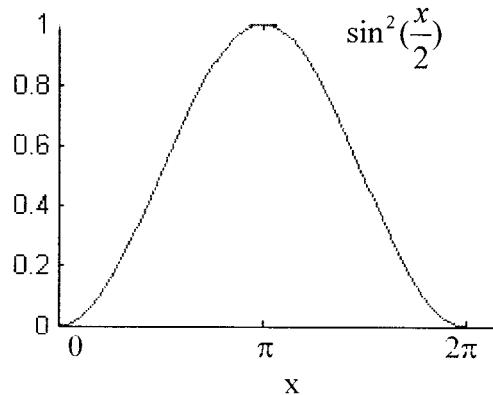


Figure 3.22 Fonction coût L

Pour étudier la faisabilité du système, nous simulons les signaux provenant de chaque prototype pour des angles d'arrivée différents et nous entraînons le réseau de neurones avec ces échantillons. L'algorithme d'apprentissage que nous allons mettre en œuvre est la rétro-propagation de l'erreur avec l'approximation de Levenberg Marquardt comme décrit à la section 2.3.2. Nous utiliserons pour cela les fonctions programmées de la Neural Network Toolbox de Matlab R12. Les différents paramètres du réseau de neurones (nombre de couches et nombre de neurones par couche) ainsi que le nombre d'époques d'entraînements sera optimisé par *validation croisée*.

Choix de l'époque par validation croisée

On dit que l'algorithme a effectué une époque quand tous les points de l'échantillon d'entraînement ont été pris en compte pour modifier les poids synaptiques. Le nombre d'époques est souvent associé au problème de généralisation d'un réseau de neurones. En effet, si l'algorithme effectue un très grand nombre d'époques, le réseau de neurones aura une erreur très faible sur l'échantillon d'entraînement mais comme l'échantillon d'entraînement n'est généralement pas exactement représentatif de tous les cas possibles, alors le réseau aura une grande erreur sur des points de *validation* pris en dehors de l'échantillon d'entraînement. Cette erreur s'appelle l'erreur de généralisation. Si l'on étudie l'erreur d'entraînement et l'erreur de généralisation en fonction des époques on constate souvent que les courbes ont l'allure présentée à la Figure 3.23. En effet à chaque époque l'algorithme améliore l'erreur du réseau sur l'échantillon d'entraînement et ce faisant, il améliore l'erreur du réseau sur l'échantillon de validation. Cependant à partir d'une certaine époque, le réseau se spécialise trop pour l'échantillon d'entraînement et son erreur de généralisation augmente (phénomène d'*overfitting*). La validation croisée consiste à arrêter l'entraînement quand l'erreur de validation est minimum.

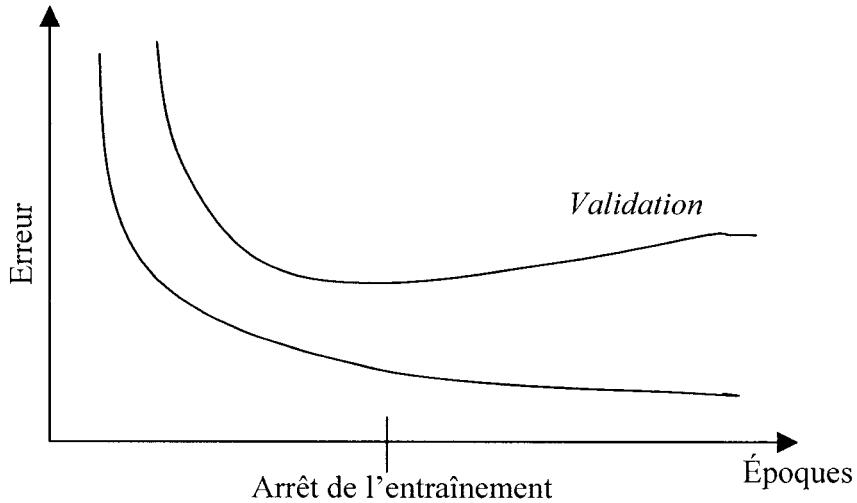


Figure 3.23 Arrêt par validation croisée

Choix de la taille du réseau, *dilemme biais/variance*

La taille du réseau influence également la qualité du résultat obtenu. Dans le cas des MLP, plus le réseau possède de neurones sur sa couche cachée, plus son pouvoir expressif est grand c'est-à-dire que la variété de fonctions qu'il peut exprimer est très grande. Ceci n'est pas toujours un avantage. Une fois de plus, comme l'apprentissage se fait sur un échantillon fini, si le réseau de neurones a un pouvoir expressif trop grand, il peut construire une fonction qui passe par tous les points de l'échantillon ce qui n'est pas forcément la meilleure fonction. Pour illustrer ce phénomène, nous allons construire 10 couples (x, y) tels que :

$$y = x^2 + \text{bruit} \quad (3.8)$$

A partir de ces couples, nous allons entraîner 2 MLP, l'un avec 3 neurones sur la couche cachée et l'autre avec 10 neurones. La Figure 3.24 présente les interpolations obtenues avec les deux MLP.

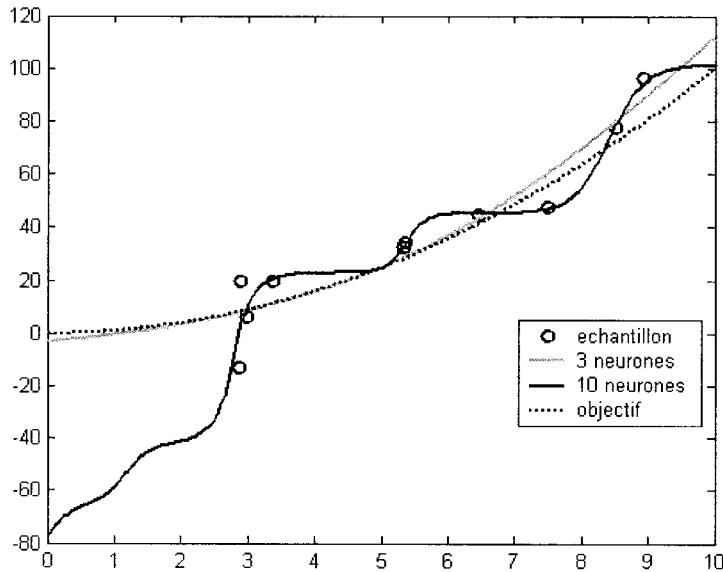


Figure 3.24 Pouvoir expressif d'un MLP

On constate que le MLP avec 3 neurones, même s'il a une erreur d'entraînement importante, est plus proche de notre objectif. Il est donc important que la taille du réseau corresponde à la classe de fonctions dans laquelle se trouve notre objectif. Ce problème est souvent nommé le dilemme biais-variance ; un réseau avec une taille trop faible ne pourra pas interpoler suffisamment de points de l'échantillon ce qui entraînera un biais sur la réponse finale. En revanche un réseau avec une taille trop grande pourra interpoler tous les points de l'échantillon et la réponse variera donc en fonction du choix de l'échantillon.

L'avantage quand on sait comment simuler les données, comme dans notre projet, c'est que l'on peut fabriquer autant d'échantillons que l'on veut. Ainsi en augmentant le nombre de points d'entraînement et le nombre de points de validation, on fait diminuer la variance de la réponse. La variance diminuant, nous pouvons évaluer la plus petite taille de réseau pouvant résoudre notre problème de régression.

3.2.3 Régression pour le prototype 1

Pour que les simulations soient le plus proche de la réalité, nous avons modélisé le gain de chaque antenne. L'antenne élémentaire utilisée est une spirale d'Archimède à un tour, [Nakano (92)]. A cause de la dissymétrie de l'antenne, la direction du maximum de rayonnement de l'antenne est décalée d'environ 10 degrés par rapport à la verticale, Figure 3.25.

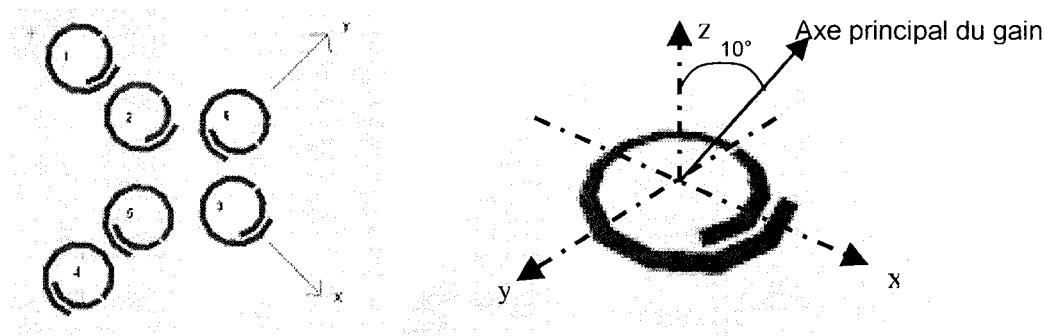


Figure 3.25 Prototype 1: réseau et antenne élémentaire

A partir de mesures effectuées par Nelson Fonseca, [Fonseca (03)], le modèle de gain proposé est le suivant :

pour une direction $[x,y,z]$ donnée,

$$G(x,y,z) = \frac{([axe]' \cdot [x,y,z])^2}{\|[x,y,z]\|^2} \quad (3.9)$$

où

$[axe]$ est la direction de l'axe principal du gain.

Comme les positions des trois antennes ne sont pas équivalentes au sein du réseau, Figure 3.25, on fixe des gains minimums différents pour chaque antenne d'un axe. La Figure 3.26 représente les gains dans le demi-espace supérieur au-dessus de chaque antenne pour des angles d'arrivée compris entre 45° et 135° . L'antenne A1 reçoit un gain minimum de 0.5, les antennes A2 et A3 reçoivent un gain minimum de 0.4.

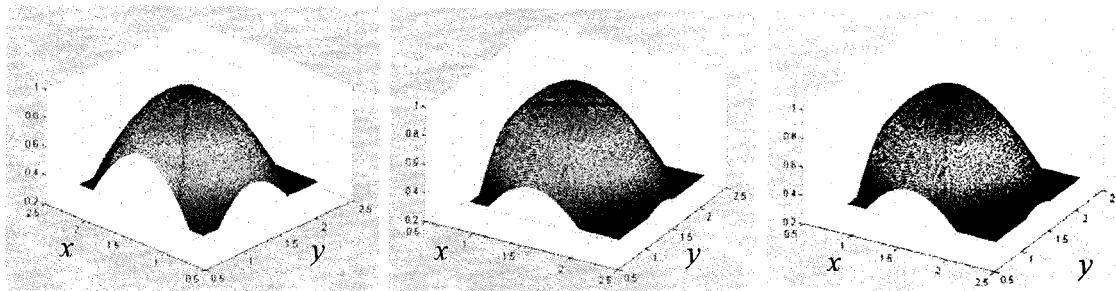


Figure 3.26 Gains des antennes dans le demi-espace supérieur

Avec ces gains, les sorties X_{1..X8} vont être modifiées. En effet pour un θ_x donné, la valeur des sorties varie suivant l'angle θ_y , mais on constate qu'après normalisation, les effets du gain spatial ne sont pas trop importants. La Figure 3.27 représente 4 sorties qui ont été évaluées pour différents angles d'arrivée (θ_x, θ_y) mais l'on ne représente que les variations par rapport à θ_x .

Enfin pour fabriquer nos échantillons de simulation nous allons par la suite rajouter du bruit à ces signaux. Ce bruit est rajouté à la sortie du combinateur, avant la conversion analogique numérique et nous le prenons de type gaussien.

En résumé :

Les signaux qui proviennent de chaque antenne sont de la forme :

$$\begin{aligned} A_1 &= G_1(\theta_x, \theta_y) \\ A_2 &= G_2(\theta_x, \theta_y).A_1 \cdot \exp(-j2\pi * 0.58 * \cos(\theta_x)) \\ A_3 &= G_3(\theta_x, \theta_y).A_1 \cdot \exp(-j2\pi * 1.37 * \cos(\theta_x)) \end{aligned} \quad (3.10)$$

Puis par exemple X_1 à la sortie du combinateur :

$$X_1 = \frac{1}{2} \left\| \frac{A_3}{2} + \frac{A_1}{2} \cdot \exp(-j\frac{\pi}{2}) \right\|^2 + bruit \quad (3.11)$$

Enfin $[X_1..X_8]$ est normalisé (3.6).

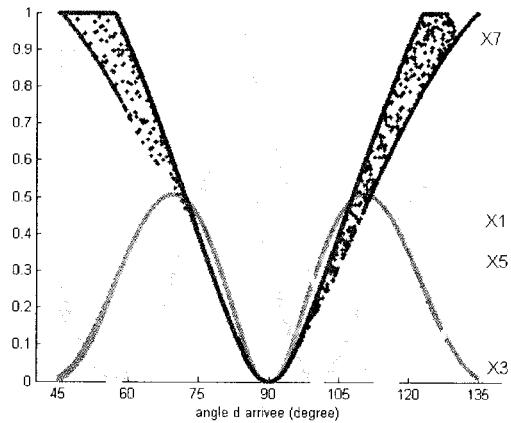


Figure 3.27 Sorties X1,X3,X5,X7 en fonction de θ_x

Conditions de simulations :

Dépendamment de la méthode, k est égal au nombre de signaux par vecteur soit 8 ou 16.

Echantillon d'entraînement : 500 vecteurs $[X_1 \dots X_k]$ correspondant à 500 directions d'arrivée prises de façon aléatoire dans le cône d'arrivée.

Echantillon test : 500 vecteurs $[X_1 \dots X_k]$ correspondant à 500 directions d'arrivée prises de façon aléatoire dans le cône d'arrivée.

Algorithme d'apprentissage : rétro-propagation de l'erreur avec l'approximation de Levenberg-Marquardt.

Choix de l'époque : Validation croisée.

Afin de vérifier que l'on peut théoriquement retrouver l'angle d'incidence à partir des entrées $[X_1 \dots X_8]$ ou $[X_1 \dots X_{16}]$ nous allons simuler ces données sans bruit.

Pour découvrir la taille du réseau optimale dépendamment des méthodes 1D et 2D, nous allons entraîner différents réseaux et choisir celui qui parvient au meilleur résultat avec la taille la plus petite. La Figure 3.28 présente l'erreur quadratique moyenne (RMSE en radians) sur l'échantillon test en fonction du nombre de neurones sur la couche cachée.

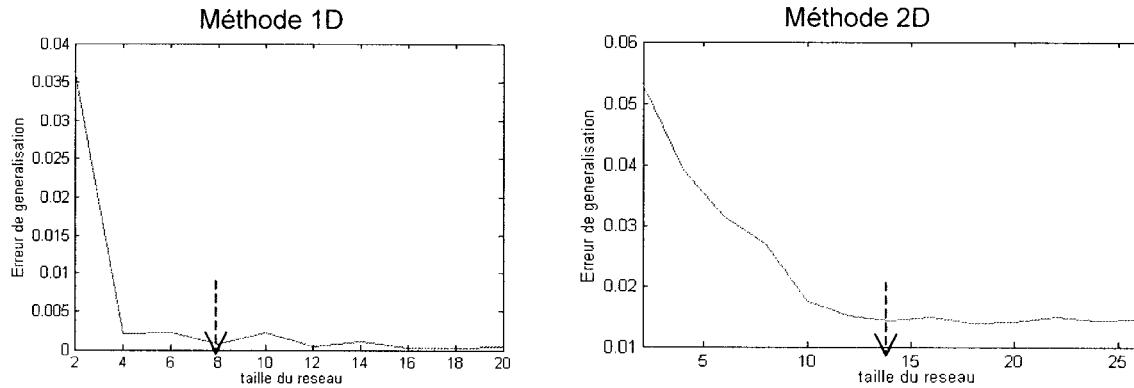


Figure 3.28 Estimation de la taille du réseau pour résoudre le problème

Les réseaux que nous allons donc utiliser pour résoudre le problème lié au prototype 1 avec les méthodes 1D ou 2D sont les MLP décrits à la Figure 3.29.

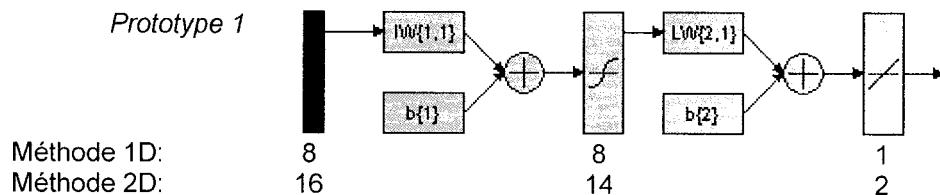


Figure 3.29 MLP pour la régression à partir des signaux du prototype 1

Nous avons testé chacun des MLP décrits précédemment pour les deux méthodes et la réponse des réseaux est représentée à la Figure 3.30.

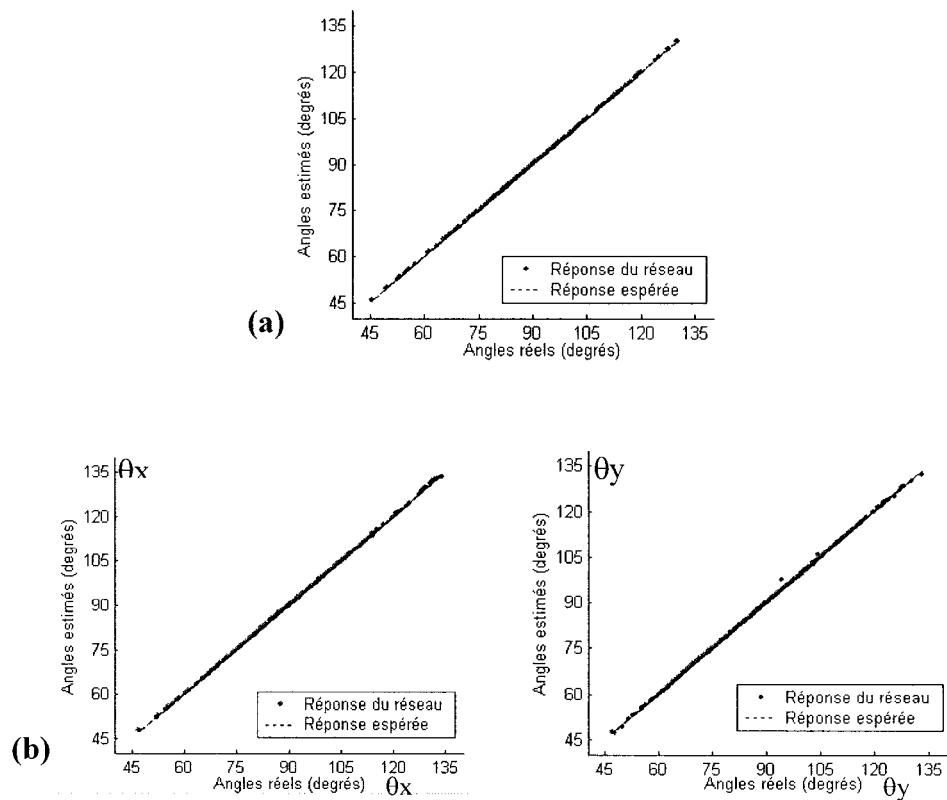


Figure 3.30 Test des réseaux avec des signaux non bruités, (a) avec la méthode 1D, (b) avec la méthode 2D

3.2.4 Régression pour le prototype 2

Pour ce nouveau prototype, Nelson Fonseca, [Fonseca (03)] a retenu une *antenne patch* comme antenne élémentaire, Figure 3.31. Le modèle de gain adopté est semblable à celui utilisé précédemment excepté que l'axe principal du gain est vertical.

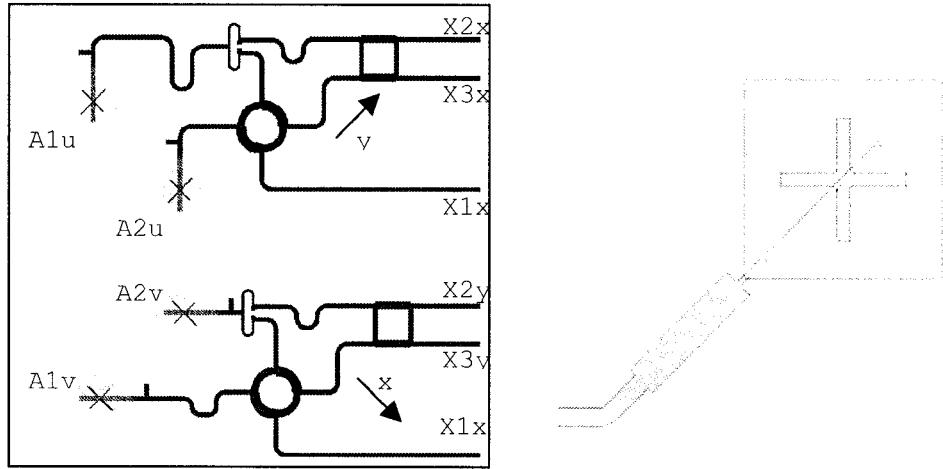


Figure 3.31 Prototype 2 à partir de 4 antennes Patch ; Antenne Patch

Dans les deux axes d'antennes, le gain minimum de l'antenne A1 est fixé à 0.5 et celui de l'antenne A2 est fixé à 0.4. Comme le gain dépend à la fois de θ_x et de θ_y , il est normal que pour un θ_x donné il y ait plusieurs signaux de sortie $[X_1, X_2, X_3]$ dépendamment de θ_y . Cependant on remarque que la perturbation introduite par le fait que le gain ne soit pas uniforme est encore plus faible que pour le prototype 1, Figure 3.32. Enfin pour fabriquer nos échantillons de simulation nous allons rajouter le même bruit qu'au prototype 1 à ces signaux.

En résumé,

- les signaux qui proviennent de chaque antenne sont de la forme :

$$\begin{aligned} A_1 &= G_1(\theta_x, \theta_y) \\ A_2 &= G_2(\theta_x, \theta_y).A_1 \cdot \exp(j2\pi * 0.58 * \cos(\theta_x)) \end{aligned} \quad (3.12)$$

- puis par exemple X_1 à la sortie du combinateur :

$$X_1 = \left\| j\sqrt{\frac{1}{3}} \cdot A_1 \cdot \exp(-j\frac{\pi}{60}) - \sqrt{\frac{2}{3}} \cdot \frac{A_2}{\sqrt{2}} \right\|^2 + \text{bruit} \quad (3.13)$$

- enfin $[X_1..X_3]$ est normalisé.

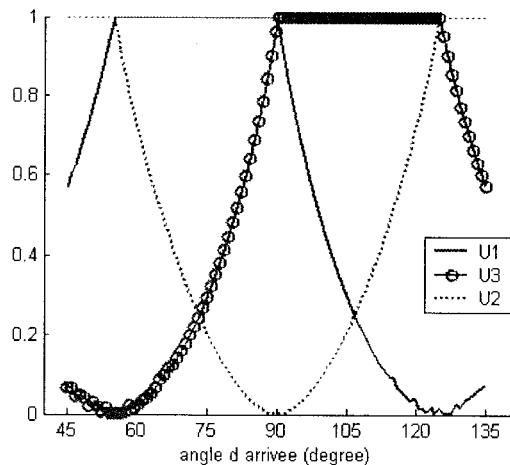


Figure 3.32 Sorties du prototype 2 avec des gains dépendants de θ_x et θ_y

Conditions de simulations :

Dépendamment de la méthode, le nombre k de signaux par vecteur est égal à 3 ou 6.

Echantillon d'entraînement : 500 vecteurs $[X_1..X_k]$ correspondant à 500 directions d'arrivée prises de façon aléatoire dans le cône d'arrivée.

Echantillon test : 500 vecteurs $[X_1..X_k]$ correspondant à 500 directions d'arrivée prises de façon aléatoire dans le cône d'arrivée.

Algorithme d'apprentissage : rétro-propagation de l'erreur avec l'approximation de Levenberg-Marquardt.

Choix de l'époque : Validation croisée.

Afin de vérifier que l'on peut théoriquement retrouver l'angle d'incidence à partir des entrées $[X_1..X_3]$ ou $[X_1..X_6]$, nous allons simuler ces données sans bruit. Encore une fois, pour déceler la taille du réseau optimale dépendamment des méthodes 1D et 2D, nous allons entraîner différents réseaux et choisir celui qui parvient au meilleur résultat avec la taille la plus petite. La Figure 3.33 présente l'erreur quadratique moyenne, RMS, (en radians) sur l'échantillon test en fonction du nombre de neurones sur la couche cachée.

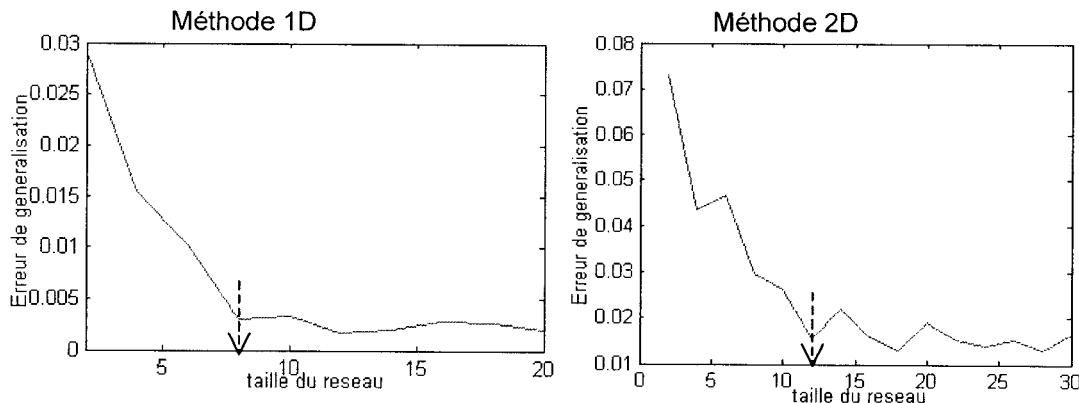


Figure 3.33 Estimation de la taille du réseau pour résoudre le problème

Les réseaux que nous allons donc utiliser pour résoudre le problème lié au prototype 2 avec les méthodes 1D ou 2D sont les MLP décrits à la Figure 3.34.

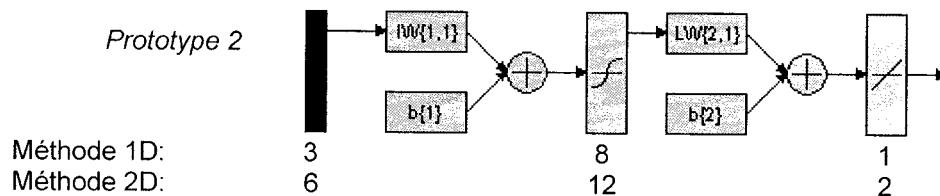


Figure 3.34 MLP pour la régression à partir des signaux du prototype 2

Nous avons testé chacun des MLP décrits précédemment pour les deux méthodes et la réponse des réseaux est représentée à la Figure 3.35.

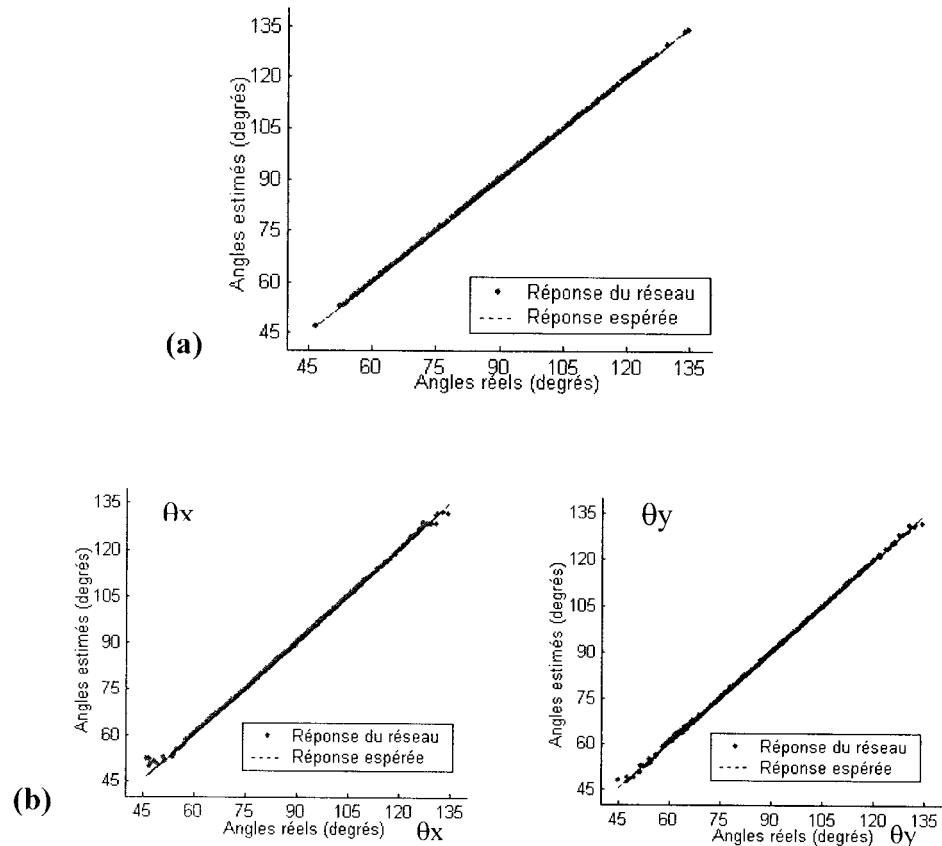


Figure 3.35 Test des réseaux avec des signaux non bruités, (a) avec la méthode 1D, (b) avec la méthode 2D

A ce stade de l'étude, nous disposons donc de quatre systèmes de repérage de la direction d'arrivée dépendamment de la méthode que nous utilisons pour traiter les signaux et du choix du prototype. Dans la section 3.3 nous expliquerons comment nous avons choisi notre système final.

3.3 Choix du système : fiabilité et précision

3.3.1 Fiabilité de la réponse

Les réseaux que nous avons optimisés jusqu'à présent, sont supposés fournir une réponse pour des angles variant dans [45, 135]. Que ce passerait-il si un vaisseau arrivait sur notre système avec un angle d'arrivée en dehors de [45, 135] ? Quelle serait la réponse du réseau ? Comment faire confiance à la réponse donnée par le système ? Ces quelques questions sont le point de départ de l'étude de fiabilité qui suit.

Tout d'abord, nous allons simuler des signaux pour les deux prototypes pour des angles d'arrivée dans [0,180] sans ajouter de bruit. Quelle que soit la méthode (1D ou 2D), comme les réseaux sont entraînés pour donner une réponse dans [45,135], la réponse est fausse dès lors que l'angle d'arrivée réel est en dehors de [45,135], Figure 3.36.

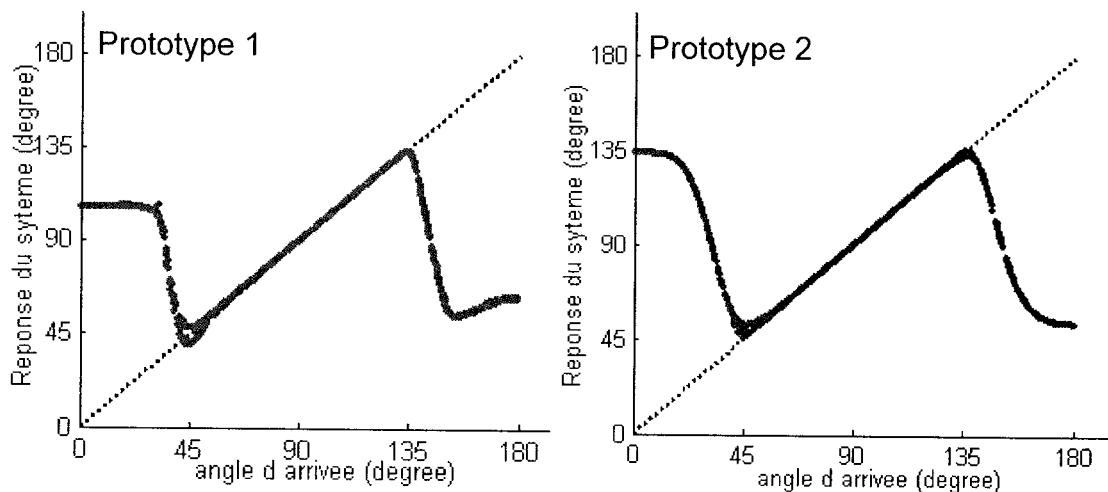


Figure 3.36 Réponses des réseaux pour des angles d'arrivée dans [0,180]

Pour être sûr que la réponse fournie par le système est correcte, nous allons créer un système de contrôle, ou réseau de fiabilité, qui sera simplement chargé de déterminer si

l'angle d'arrivée est en dedans ou en dehors de notre cône d'étude et donc de dire si l'angle estimé est fiable ou non.

Si on divise le problème en deux axes, ce filtre de fiabilité consiste donc à classifier les signaux en deux :

- la classe +1 si l'angle d'arrivée est dans [45,135]
- la classe 0 si l'angle d'arrivée est en dehors de [45,135]

Méthode 1D

Tout d'abord, nous allons vérifier que les signaux de chaque prototype permettent de faire cette classification avec un bruit nul. Pour cela nous allons choisir les réseaux de taille minimum pour chacun des problèmes, Figure 3.37. Il est intéressant de noter qu'un simple perceptron suffit pour classifier les signaux du prototype 2. Cela signifie que les 3 signaux sont linéairement séparables. Géométriquement, cela signifie qu'il existe un plan qui sépare les vecteurs $[X_1, X_2, X_3]$ provenant d'un angle dans [45,135] et les vecteurs $[X_1, X_2, X_3]$ provenant d'un angle en dehors de [45,135], Figure 3.38.

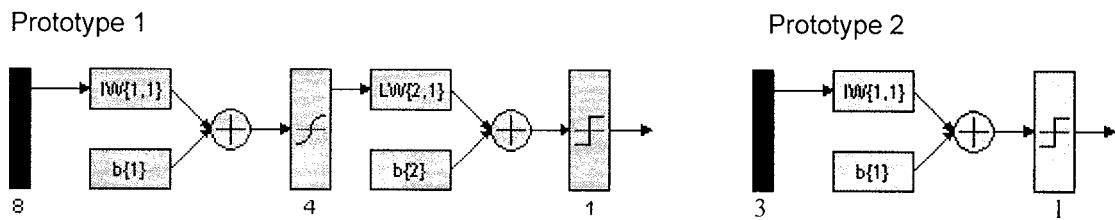


Figure 3.37 Réseaux pour la classification des signaux avec la méthode 1D

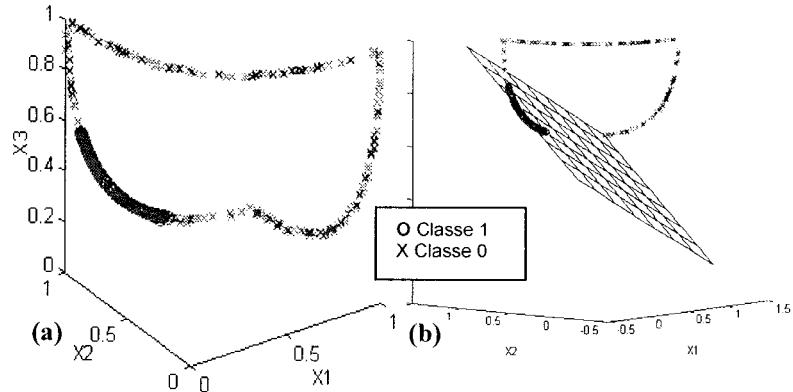


Figure 3.38 (a) Répartition géométrique de $[X_1, X_2, X_3]$ quand θ varie dans $[0, 180]$
(b) Plan de classification

Pour trouver le plan optimal, il est possible d'utiliser la méthode des Support Vector Machine. Cependant, comme il y a continuité des données quand on fait varier l'angle d'arrivée, la marge séparant les deux classes sera nécessairement nulle. La sensibilité au bruit sera donc très importante pour les données proche de la frontière.

Méthode 2D

On peut noter que pour qu'un signal soit classifié dans le cône d'arrivée, il faut que le réseau de fiabilité de l'axe X et le réseau de fiabilité de l'axe Y aient classé le signal +1. Cependant cette condition n'est pas suffisante. En effet le cône d'étude est contenu mais n'est pas égal à l'espace défini par $(\theta x, \theta y) \in [45, 135] \times [45, 135]$, Figure 3.39.

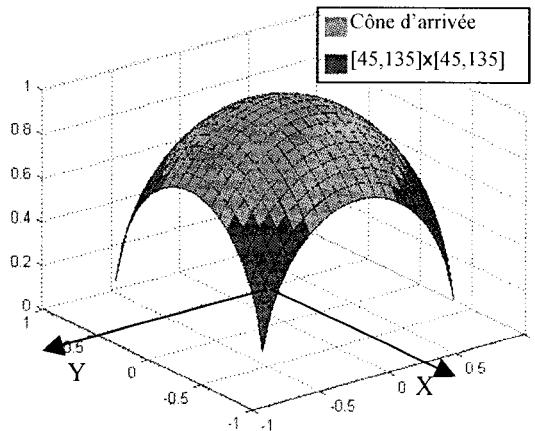


Figure 3.39 Cône d'étude

Pour que la classification porte bien sur le cône d'étude prédéfini, nous allons traiter le problème de la fiabilité en tenant compte des signaux provenant des deux axes à la fois. Dans ce nouveau contexte, après optimisation, les réseaux de taille minimum pour résoudre le problème lié aux prototypes 1 et 2 ont 16 (resp. 6) entrées et 10 (resp. 4) neurones sur la couche cachée comme l'indique la Figure 3.40.

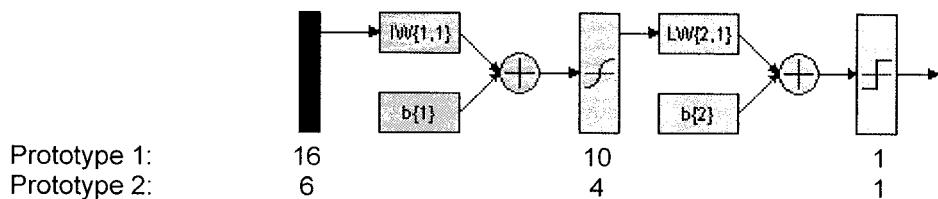


Figure 3.40 MLP de classification, méthode 2D

Dans toute la suite le problème de la classification ne sera abordé qu'avec la méthode 2D. La Figure 3.41 présente la réponse des réseaux de classification pour des signaux arrivant dans le demi-espace supérieur au plan des antennes. Les résultats de la classification sont identiques pour les prototypes 1 ou 2.

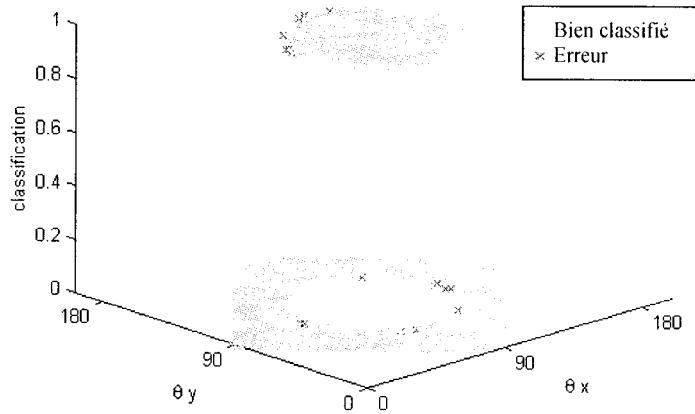


Figure 3.41 Classification des signaux par rapport au cône d'étude

3.3.2 Robustesse de la réponse face au bruit

Par convention pour le prototype 2, quand nous dirons que le rapport signal sur bruit est de 12 dB (par exemple), cela signifie que nous ajoutons autant de bruit que nous l'avions fait pour le prototype 1. En fait comme les signaux du prototype 2 sont plus puissants, à bruit égal le rapport signal sur bruit est meilleur, comme l'exprime l'équation (3.15). Mais comme on veut comparer les deux prototypes, il faut les examiner dans les mêmes conditions.

D'après la Figure 3.14 et la Figure 3.18, si on appelle P1 (respectivement P2) la puissance maximale des signaux du prototype 1 (respectivement 2), alors

$$\frac{P_2}{P_1} = 1.36 \quad (3.14)$$

En conséquence de (3.14), à bruit égal, si l'on exprime les RSB en décibel,

$$\text{RSB}_{\text{proto2}} = \text{RSB}_{\text{proto1}} + 1.3 \text{ dB} \quad (3.15)$$

Pour comparer les systèmes, le niveau de bruit, RSB, de référence sera le $\text{RSB}_{\text{proto1}}$.

Si l'on ajoute du bruit, il y aura des points mal classifiés c'est-à-dire qu'il y aura des signaux avec un angle d'arrivée dans le cône d'étude qui seront classifiés dans la classe 0 et des signaux avec un angle d'arrivée en dehors du cône d'étude qui seront classifiés dans la classe 1. Cette dernière erreur est plus grave car cela implique que le système donnera une réponse qui est fausse. On peut étudier l'erreur commise pour chaque prototype en fonction du Rapport Signal sur Bruit, Figure 3.42. On peut tirer deux conclusions de cette dernière étude : la première est que le système issu du prototype 2 est plus performant que celui issu du prototype 1 et la seconde est qu'il va falloir augmenter au moyen d'un filtre le rapport signal bruit si l'on veut une réponse suffisamment fiable pour un RSB de 10 dB.

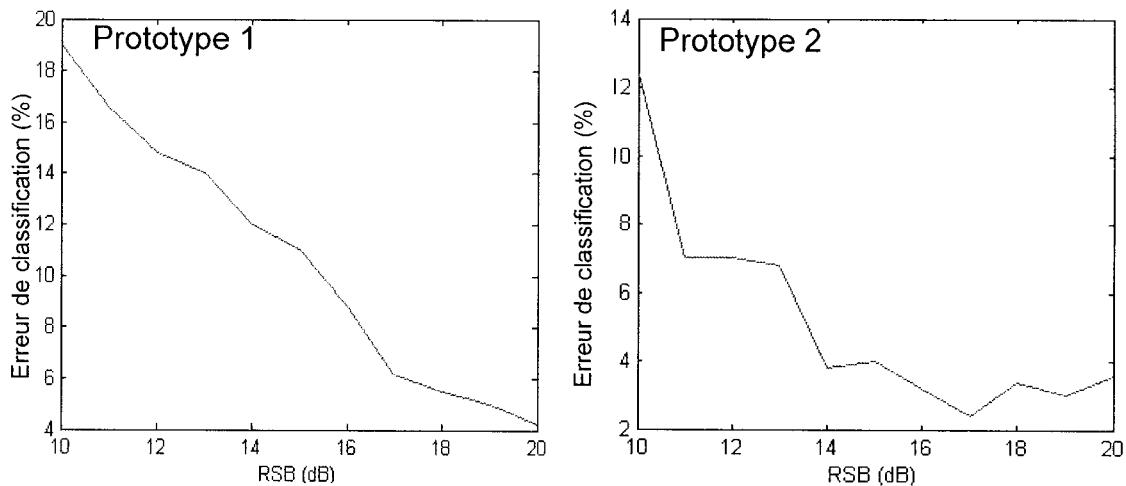


Figure 3.42 Erreur de classification

Le système de repérage de la direction d'arrivée que nous concevons est initialement prévu pour être embarqué à bord d'un micro-satellite de l'Agence Spatiale Canadienne. L'étude a été menée en supposant que l'altitude du micro-satellite se situe aux environs de 1000 km de la surface de la terre, sa période de révolution est de l'ordre de 1 heure 30 minutes. Le faisceau que le système devra détecter sera émis par un

satellite géostationnaire, par exemple MSat, ce qui signifie que l'angle d'incidence du faisceau sur le micro-satellite variera d'un degré toutes les 15 secondes, Figure 3.43.

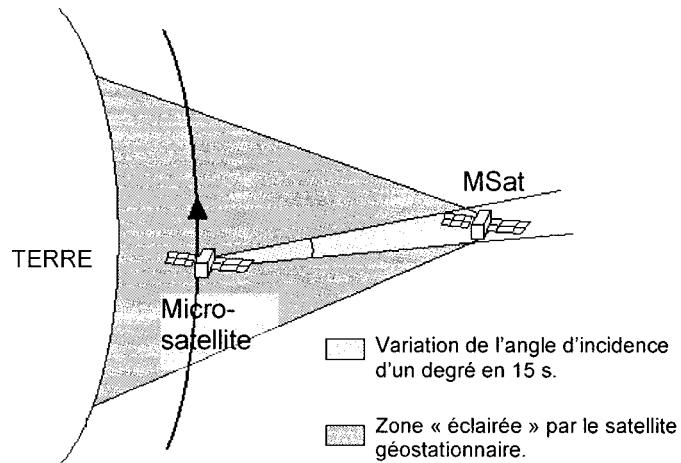


Figure 3.43 Système embarqué sur un micro-satellite. (non à l'échelle)

Comme notre objectif est d'avoir une résolution angulaire à un degré près, nous avons 15 secondes pour estimer la direction d'arrivée. Durant ce laps de temps il est possible de faire plusieurs mesures et de les filtrer afin que la réponse finale soit moins bruitée. On appelle durée d'une mesure, le temps nécessaire pour obtenir la valeur numérique des signaux provenant des deux axes d'antennes. Nous avons évalué de façon expérimentale que cette durée n'excède pas 0.5 s. Ainsi il est possible d'effectuer 30 mesures avant que le système donne la direction d'arrivée. Comme nous l'avons vu à la section 2.4 nous disposons de différents outils pour augmenter le RSB des signaux dépendamment de la nature du bruit et le choix du filtre dépend donc de l'environnement dans lequel notre système va être implanté : sur terre, il y a de multiples sources de perturbations, comme le passage d'un véhicule aux environs des antennes par exemple, en revanche dans l'espace il est peu probable que des objets viennent momentanément perturber le signal. Nous ferons l'hypothèse que le bruit que nous voulons éliminer est principalement d'origine thermique et est donc assimilé à un bruit blanc.

3.3.3 Comparaisons des différents systèmes

Conditions de simulation :

Pour chacun des systèmes (prototype 1 et 2, méthode 1D et 2D), les données sont générées comme précédemment. La convention pour le RSB est la même que précédemment c'est-à-dire que le niveau de bruit est désigné par RSB_{proto} . Quand nous ajoutons du bruit, nous le faisons pour les données d'entraînement et pour les données de test, comme si les données provenaient d'un système «en activité».

Les MLP utilisés sont ceux décrits aux Figure 3.29 et Figure 3.34 et pour chaque niveau de bruit, on fait l'apprentissage du réseau :

Echantillon d'entraînement : 500*30 vecteurs $[X_1 \dots X_k]$ correspondants à 30 vecteurs pour chacune des 500 directions d'arrivée prises de façon aléatoire dans le cône d'arrivée. Le filtre fait la moyenne des 30 vecteurs et ce sont les données filtrées qui sont envoyées au réseau de neurones.

Echantillon test : 200*30 vecteurs $[X_1 \dots X_k]$ (respectivement $[X_1 \dots X_k]$) correspondants à 30 vecteurs pour chacune des 200 directions d'arrivée prises de façon aléatoire dans le cône d'arrivée. Le filtre fait la moyenne des 30 vecteurs et ce sont les données filtrées qui sont envoyées au réseau de neurones.

Algorithme d'apprentissage : rétro-propagation de l'erreur avec l'approximation de Levenberg-Marquardt.

Choix de l'époque : Validation croisée.

On rappelle que pour la méthode 1D on entraîne indépendamment deux réseaux pour les deux axes et que pour la méthode 2D on entraîne qu'un seul réseau pour les deux axes. La Figure 3.44 illustre le choix de l'époque par validation croisée pour les MLP liées au prototype 1 pour un RSB de 12 dB. On ne représente pas toutes les courbes d'apprentissage car quelles que soient les réseaux et le bruit, l'allure des courbes est la même.

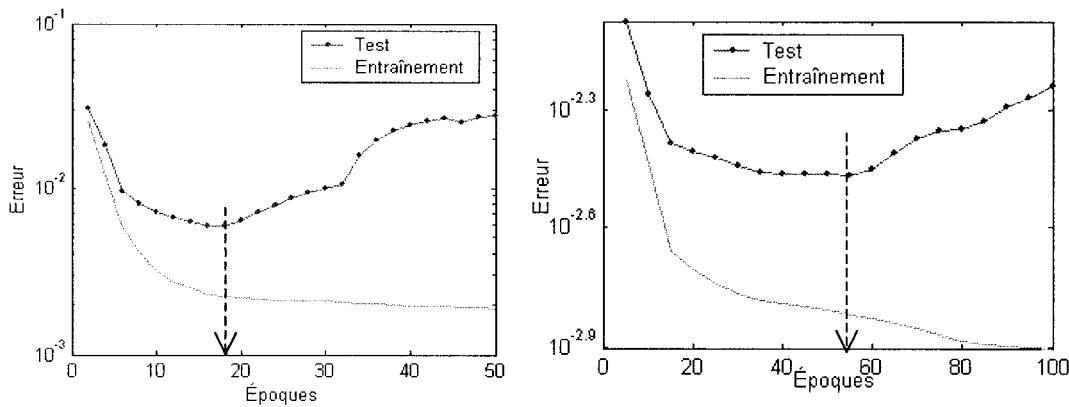


Figure 3.44 Choix de l'époque. (a) pour le réseau de la méthode 1D. (b) pour le réseau de la méthode 2D

Précision de la réponse :

Pour mesurer la précision, on génère 900×30 vecteurs correspondant à 900 directions d'arrivée dans le cône d'étude. On mesure l'erreur commise entre la réponse (θ_x, θ_y) et la direction réelle d'arrivée, Figure 3.45. On calcule enfin l'erreur quadratique moyenne, RMS, commise à l'intérieur des cônes avec différentes ouvertures angulaires.

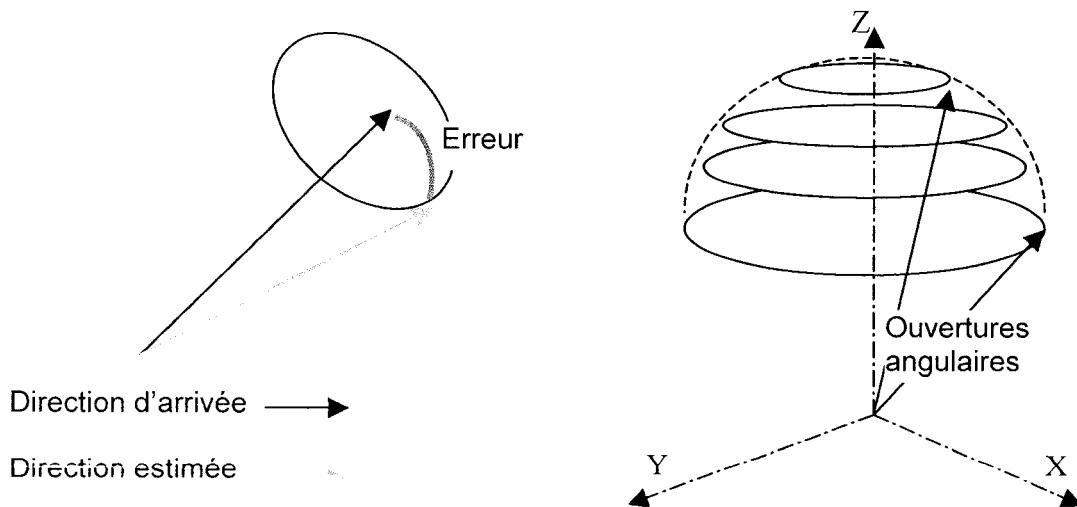


Figure 3.45 Calcul de l'erreur. Ouvertures angulaires du cône

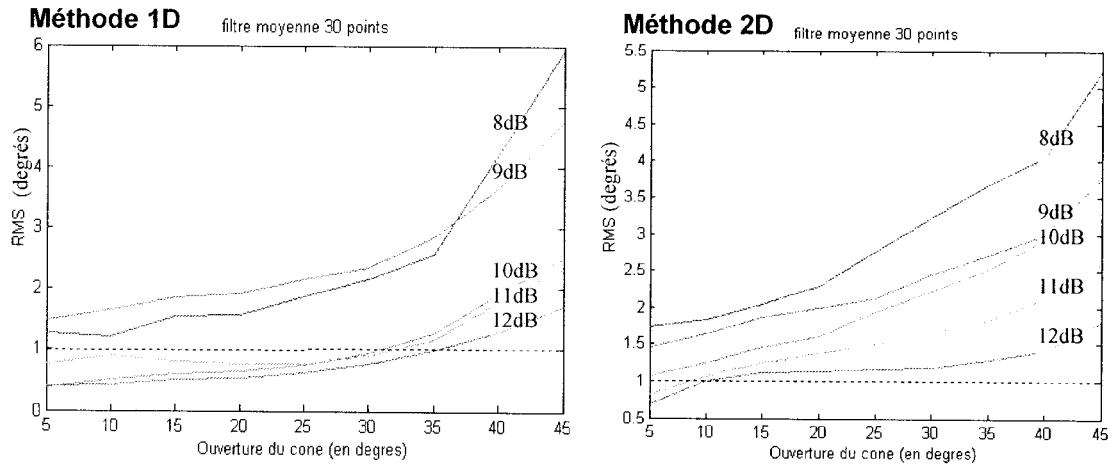


Figure 3.46 Performance du prototype 1 sous différentes conditions de bruit

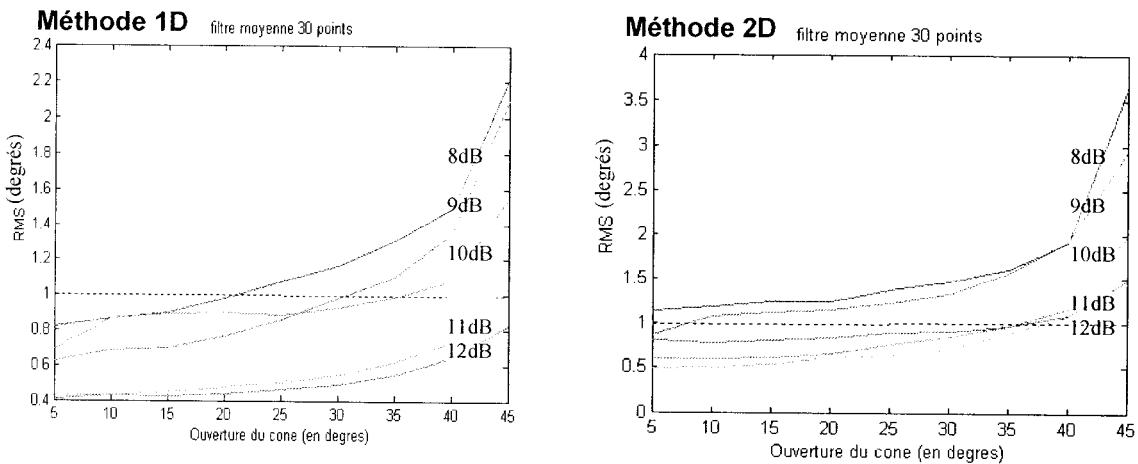


Figure 3.47 Performance du prototype 2 sous différentes conditions de bruit

Fiabilité de la réponse :

Nous allons répéter la même procédure pour tester les réseaux de fiabilité. Cependant le filtrage est effectué sur 50 points. En effet sur ce dernier filtre, nous pouvons nous permettre de prendre plus de points car la classification est moins exigeante sur la précision angulaire. Les MLP utilisés sont ceux décrits à la Figure 3.40.

Une erreur de type 1 correspond à un angle d'arrivée dans le cône classifié dans la classe 0 et une erreur de type 2 correspond à un angle d'arrivée hors du cône et classifié dans la classe 1.

Prototype 1 :

La Figure 3.48 (a) présente les taux d'erreurs en fonction du niveau de bruit.

La Figure 3.48 (b) présente les résultats de la classification pour un RSB=10dB

La Figure 3.50 (a) présente les erreurs de type 1 en fonction de l'ouverture du cône pour un RSB=10dB.

La Figure 3.50 (b) présente les erreurs de type 2 en fonction de l'ouverture du cône pour un RSB=10dB.

Prototype2 :

La Figure 3.49 (a) présente les taux d'erreurs en fonction du niveau de bruit.

La Figure 3.49 (b) présente les résultats de la classification pour un RSB=10dB

La Figure 3.51 (a) présente les erreurs de type 1 en fonction de l'ouverture du cône pour un RSB=10dB.

La Figure 3.51 (b) présente les erreurs de type 2 en fonction de l'ouverture du cône pour un RSB=10dB.

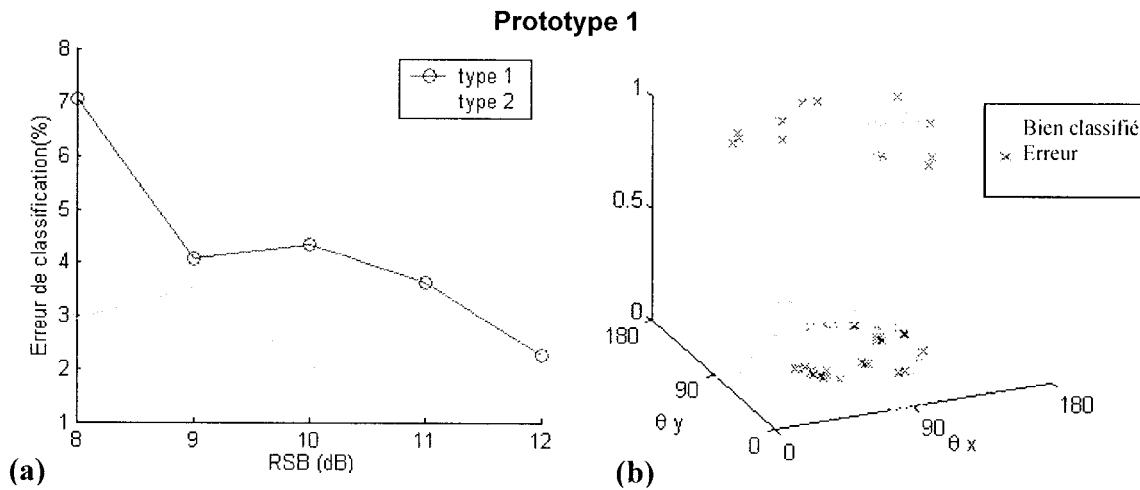


Figure 3.48 Erreur de classification (a) en fonction du niveau de bruit

(b) en fonction de la direction d'arrivée pour RSB=10dB

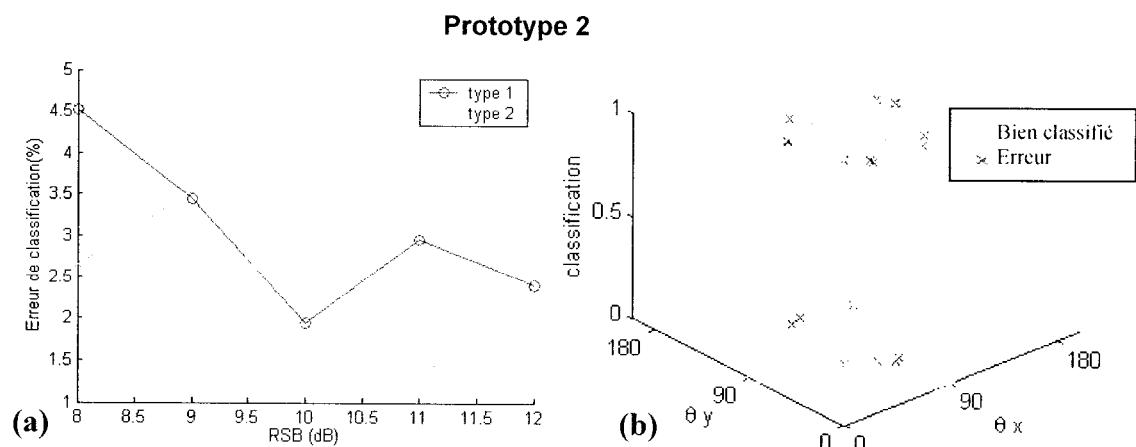


Figure 3.49 Erreur de classification (a) en fonction du niveau de bruit (b) en fonction de la direction d'arrivée pour RSB=10dB

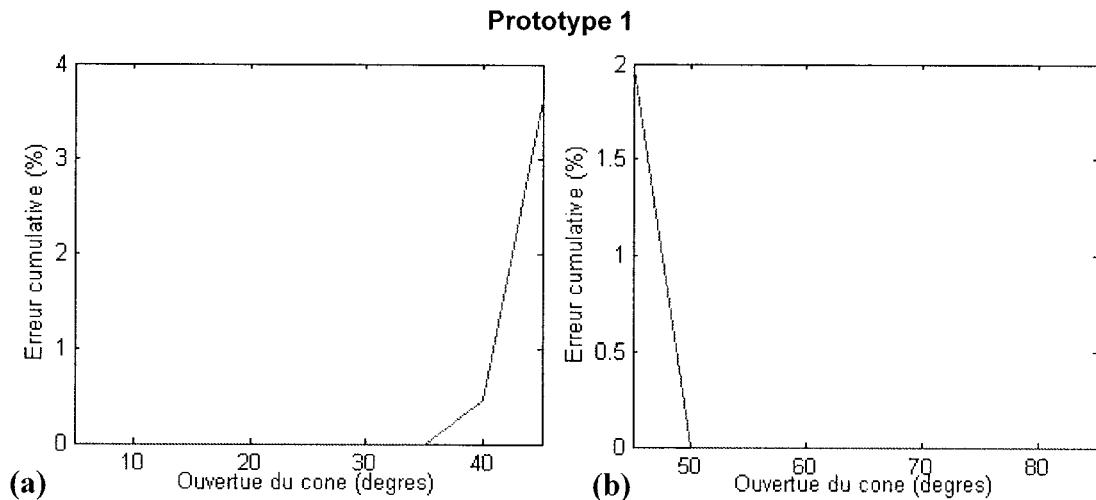


Figure 3.50 RSB=10dB. (a) Erreur de type 1. (b) Erreur de type 2.

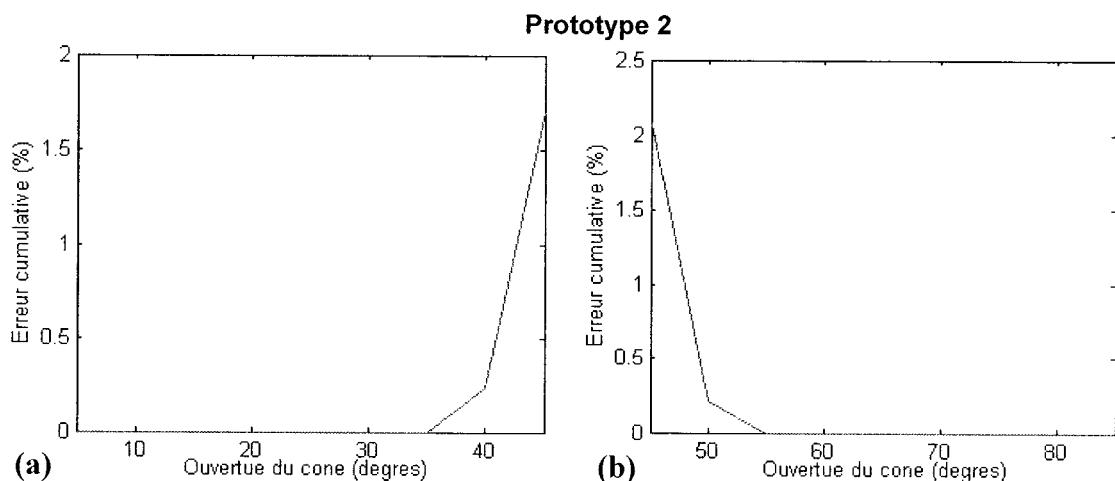


Figure 3.51 RSB=10dB. (a) Erreur de type 1. (b) Erreur de type 2.

Choix du Prototype :

Aussi bien en terme de précision qu'en terme de fiabilité, le prototype 2 est plus performant que le prototype 1. Pour un rapport signal sur bruit de 10 dB, le système créé à partir du prototype 2 offre une résolution angulaire avec moins de 1° d'erreur pour une ouverture de cône de 35° et une résolution à moins de 2° d'erreur pour tout le cône d'étude (ouverture de 45°). Ces bonnes performances sont fiables car les erreurs de classification sont négligeables (inférieures à 2%). Il est important que le système commette peu d'erreurs de type 2 car une telle erreur consiste à accepter une réponse fausse. Les erreurs de type 1 signifient que le système ne fournit pas la direction d'arrivée d'un faisceau qui arrive dans le cône d'étude.

Choix de la méthode :

Les résultats de chaque méthode sont intimement liés à notre modélisation du gain des antennes. A priori la méthode 1D obtient de meilleurs résultats que la méthode 2D mais il faut bien noter que nous avons fait un modèle de gain très régulier pour les trois antennes. Nous réservons donc le choix définitif de la méthode une fois que nous pourrons tester les deux méthodes avec des données expérimentales.

Il reste à présent à confronter ces résultats avec des données réelles et vérifier si nos hypothèses n'étaient ni trop optimistes ni trop pessimistes.

CHAPITRE IV

Validations expérimentales

Dans ce chapitre, nous ne considérerons que le système basé sur le prototype 2 comme décrit dans le Chapitre 3. En effet les simulations qui précédent nous indiquent que c'est celui-ci qui offre les meilleures performances pour la détection de l'angle d'incidence. Afin de valider les résultats et les bonnes performances du système, Nelson Fonseca, [Fonseca (03)] a réalisé le réseau d'antennes sur le modèle du prototype 2. Dans la section 4.1 nous décrivons brièvement les caractéristiques physiques du réseau et la méthode que nous avons employée pour collecter les données utiles aux réseaux de neurones. Dans la section 4.2, nous testons notre système avec les signaux obtenus de façon expérimentale. Enfin dans la section 4.3 nous dressons le bilan des résultats et du travail accompli.

4.1 Réalisation du réseau d'antennes, collecte des données

Le prototype a été réalisé pour une fréquence de 5.8 GHz. Les dimensions de l'antenne élémentaire ont été optimisées par Nelson Fonseca et sont répertoriées dans (4.1) et définies dans la Figure 4.1.

$$\left\{ \begin{array}{l} a = 12,19 \text{ mm} \\ L_{ax} = 7,06 \text{ mm} \\ L_{ay} = 8,59 \text{ mm} \\ w_a = 0,38 \text{ mm} \\ L_{ni} = 5,31 \text{ mm} \\ w_f = 1,29 \text{ mm} \end{array} \right. \quad (4.1)$$

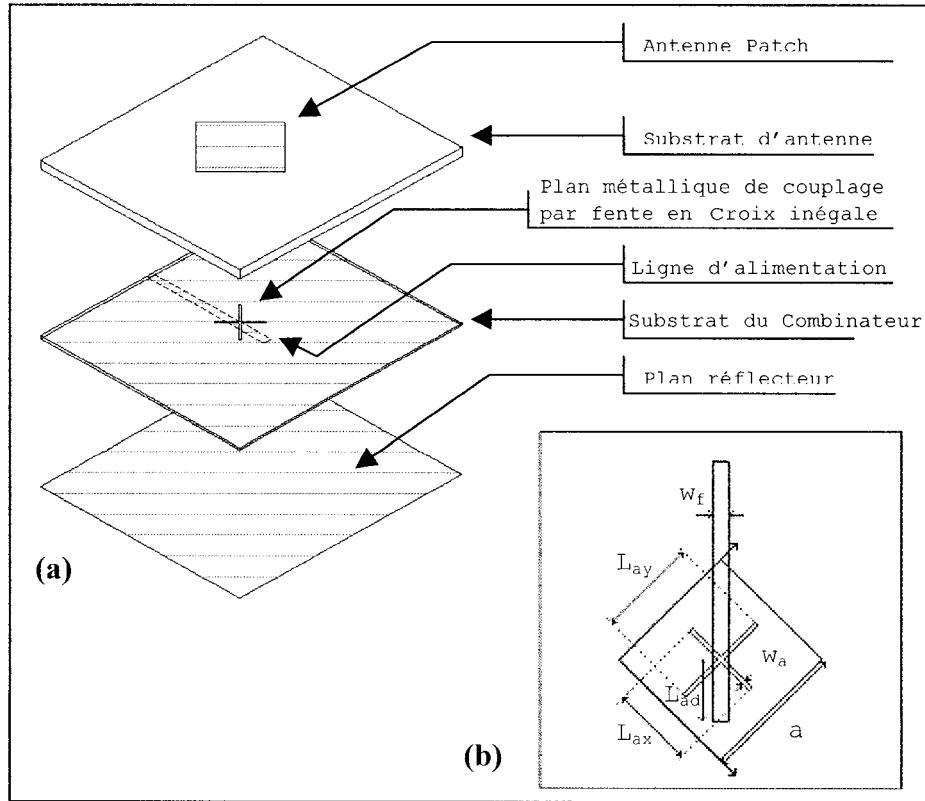


Figure 4.1 Antenne Patch carrée. (a) Vue éclatée. (b) Vue du dessous

A partir de quatre de ces antennes élémentaires, Nelson Fonseca a réalisé le réseau d'antennes et le combinateur avec les spécificités décrites au Chapitre 3. À la fréquence de 5.8 GHz, la distance entre deux antennes d'un même axe est de 30 mm ce qui en fait un réseau peu encombrant. Pour générer les 4 signaux sortant du combinateur, nous avons illuminé le réseau sous différents angles d'incidence. Pour cette dernière étape, l'ensemble constitué du réseau d'antennes et du combinateur, Figure 4.2(b), est monté sur un bras rotatif deux axes et l'antenne émettrice est fixe, Figure 4.2(a).

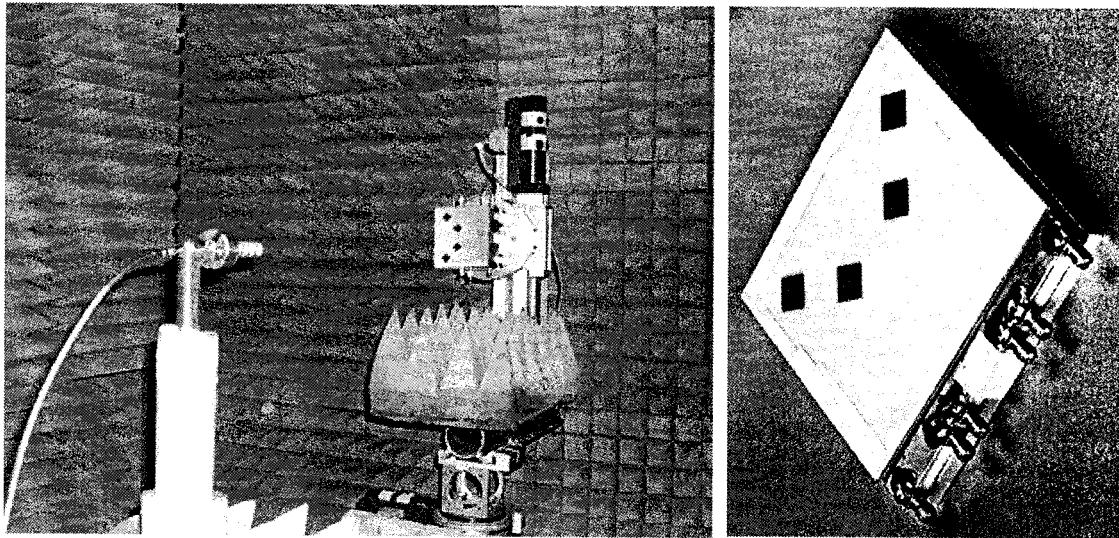


Figure 4.2 (a) Procédé expérimental. (b) Réseau d'antennes et combinateur

Au cours de cette expérience, nous avons collecté 32 400 mesures pour chaque sortie du combinateur. Ces 32 400 mesures couvrent le demi-espace supérieur au plan du réseau d'antennes. En coordonnées sphériques, ces mesures correspondent à des angles variant de 90° en élévation et de 360° en azimut ($90 \times 360 = 32\,400$).

Pour chaque sortie du combinateur, une mesure est donc un triplet $\{\theta_x, \theta_y, P\}$ où (θ_x, θ_y) est l'angle d'arrivée du faisceau par rapport aux axes du réseau d'antennes et P est la puissance du signal correspondant.

La Figure 4.3 représente la puissance des signaux à la sortie du combinateur pour l'axe X en fonction de θ_x . La différence entre l'expérience et la simulation provient essentiellement de la modélisation du gain. Nous constatons que le modèle de gain que nous avions adopté était trop pessimiste dans [45,135], car dans la réalité le gain varie très peu sur cet intervalle. En revanche pour des angles proches de 0° ou 180° , notre modèle semble optimiste car dans l'expérience le gain est quasi-nul pour ces valeurs extrêmes.

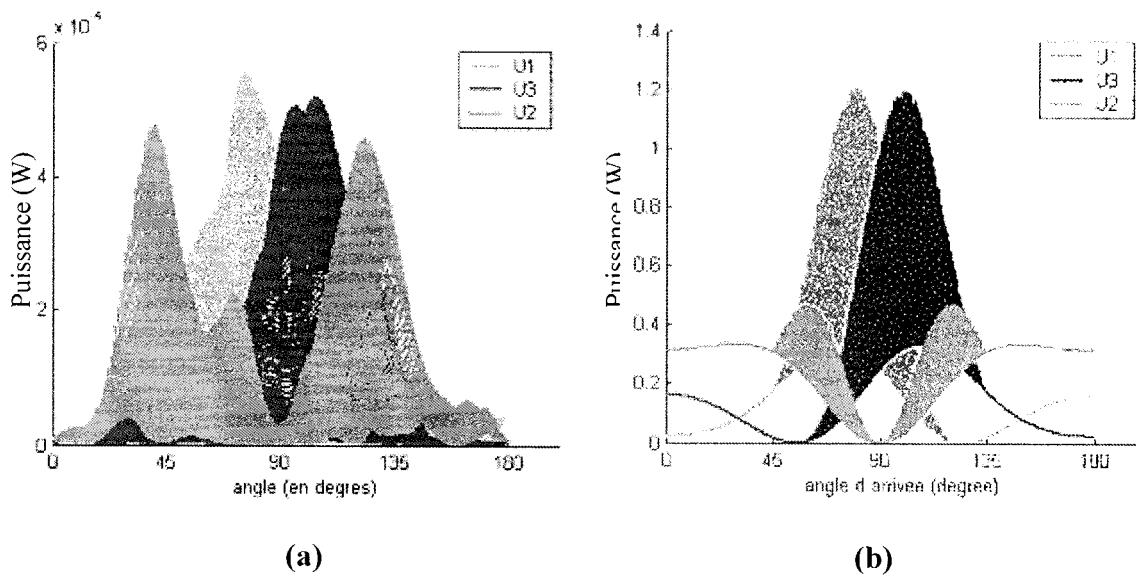


Figure 4.3 Puissances des signaux en fonction de l'angle d'incidence

(a) Expérimentales (b) Simulations théoriques

Les données que nous fournissons aux réseaux de neurones sont les signaux expérimentaux auxquels nous allons appliquer la normalisation des méthodes 1D et 2D décrites par l'équation (3.6). Pour les réseaux qui effectuent la régression, nous prendrons des données qui proviennent du cône d'étude, Figure 4.4 et Figure 4.5. Pour le réseau de fiabilité, nous prendrons des données qui proviennent de tout le demi-espace supérieur.

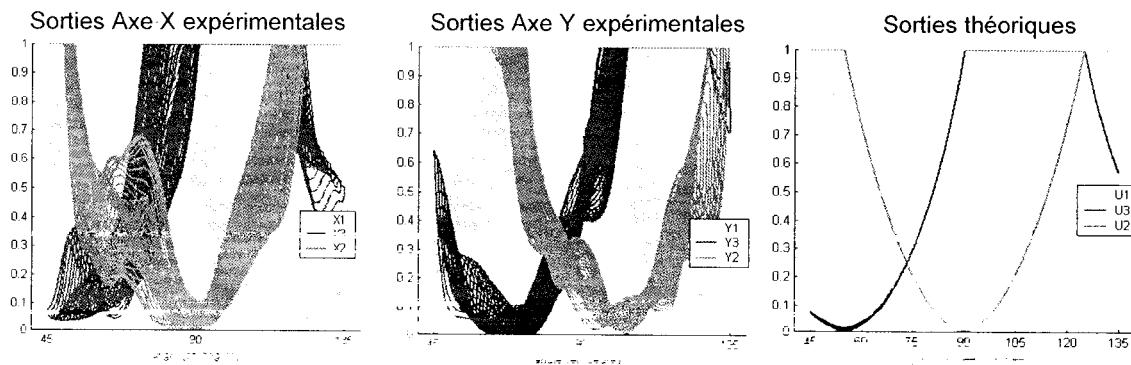


Figure 4.4 Données pour la méthode 1D dans le cône d'étude

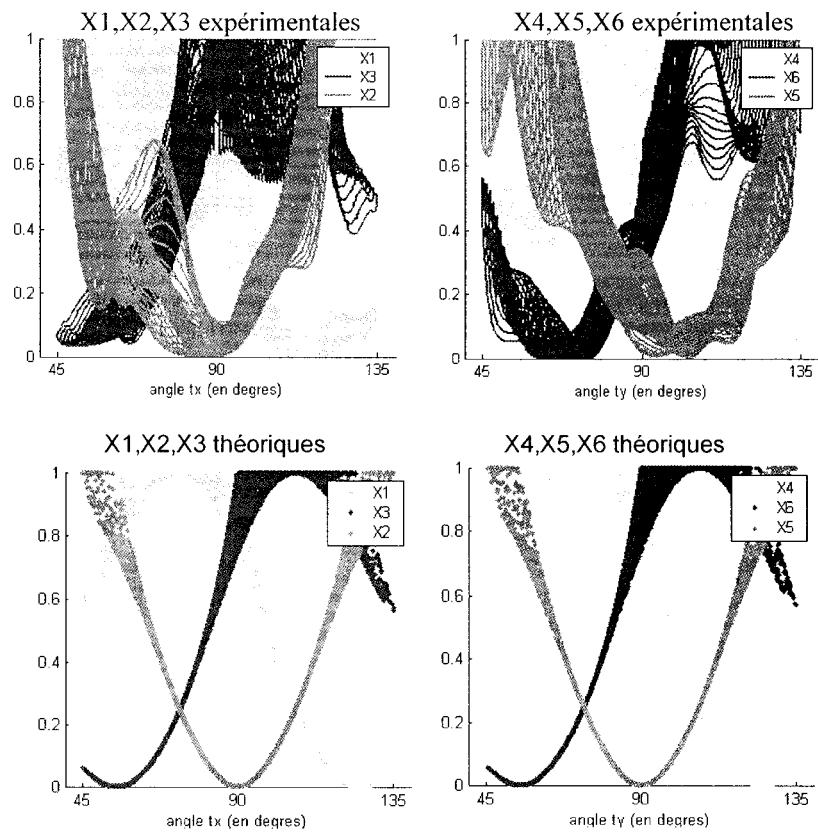


Figure 4.5 Données pour la méthode 2D dans le cône d'étude

Comme nous l'avions vu précédemment, les Figure 4.4 et Figure 4.5 montrent que l'information bi-dimensionnelle du gain des antennes est moins atténuée par la normalisation de la méthode 2D que par celle de la méthode 1D. Nous allons à présent tester la régression à partir des données expérimentales avec les méthodes 1D et 2D.

4.2 Apprentissage et choix du système global

Nous allons appliquer les procédures que nous avons mises en place lors des simulations pour le choix des réseaux et pour leur apprentissage.

4.2.1 Régression, méthode 1D

Choix de la taille du réseau : comme l'illustre la Figure 4.6, ce sont des MLP avec 8 neurones sur la couche cachée pour l'axe X et 12 pour l'axe Y qui ont été retenus. Les deux réseaux ont des tailles différentes car les signaux sont différents pour les deux axes, Figure 4.4.

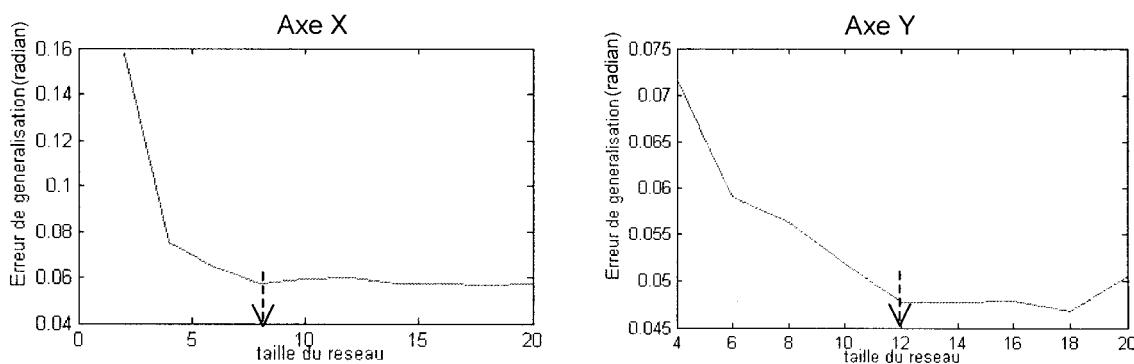


Figure 4.6 Erreur de régression en fonction du nombre de neurones cachés.

L'apprentissage : il se fait à partir de 1000 points d'entraînement et 500 points tests pris au hasard parmi les données expérimentales, indépendamment pour chaque axe. La Figure 4.7 présente les sorties des deux réseaux pour l'échantillon test.

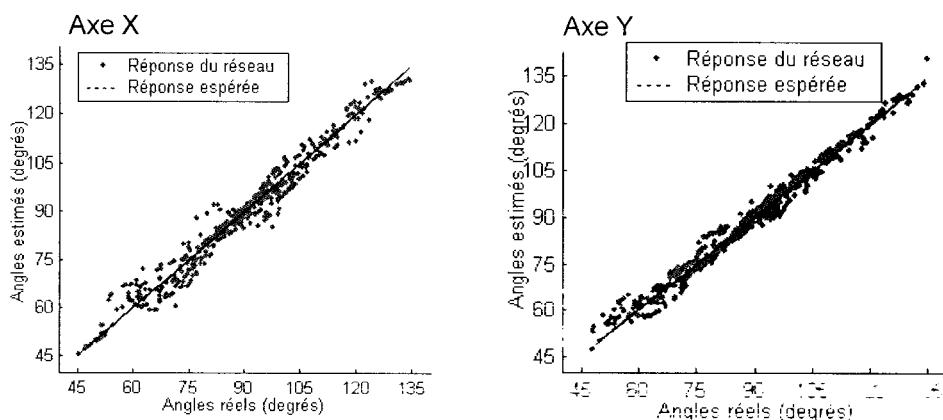


Figure 4.7 Test des réseaux pour les deux axes de façon indépendante

Performance : le système constitué des deux réseaux est testé sur toutes les données expérimentales prises dans le cône d'étude. La Figure 4.8 présente l'erreur quadratique moyenne sur l'estimation de la direction d'arrivée en fonction de l'ouverture du cône dans lequel arrivent les signaux.

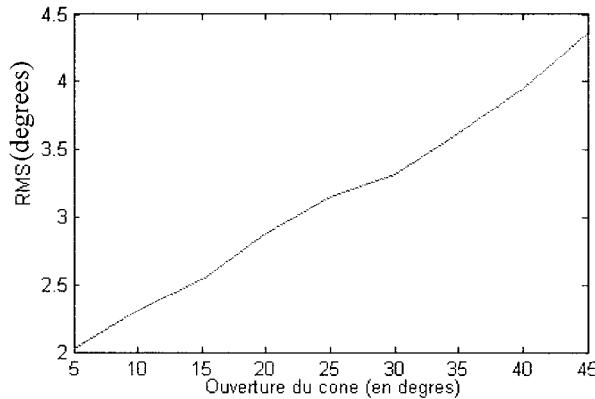


Figure 4.8 Performance de la méthode 1D

Cela indique qu'avec la méthode 1D nous ne pouvons pas créer une fonction de régression parfaite c'est-à-dire qu'il est possible que deux angles d'arrivée différents génèrent des signaux égaux. La non-bijectivité des signaux provient des variations spatiales des caractéristiques du gain des antennes.

4.2.2 Régression, méthode 2D

Choix de la taille du réseau : comme l'illustre la Figure 4.9, c'est un MLP avec 20 neurones sur la couche cachée qui a été retenu.

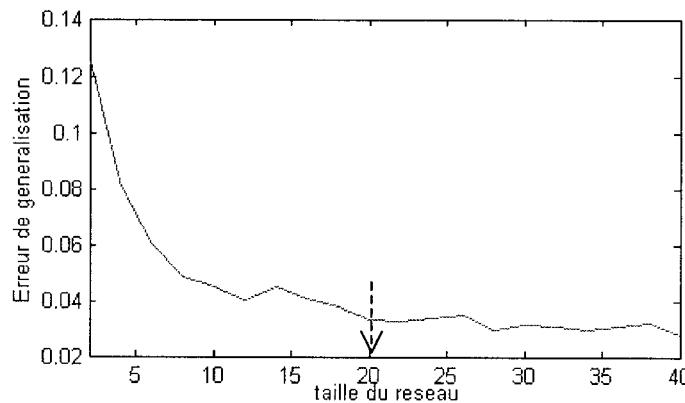


Figure 4.9 Erreur de régression en fonction du nombre de neurones cachés.

L'apprentissage : il se fait à partir de 1000 points d'entraînement et 500 points tests pris au hasard parmi les données expérimentales. La Figure 4.10 (a) illustre l'évolution de l'erreur en fonction des époques et les Figure 4.10 (b) et (c) présentent les sorties du réseau pour l'échantillon test.

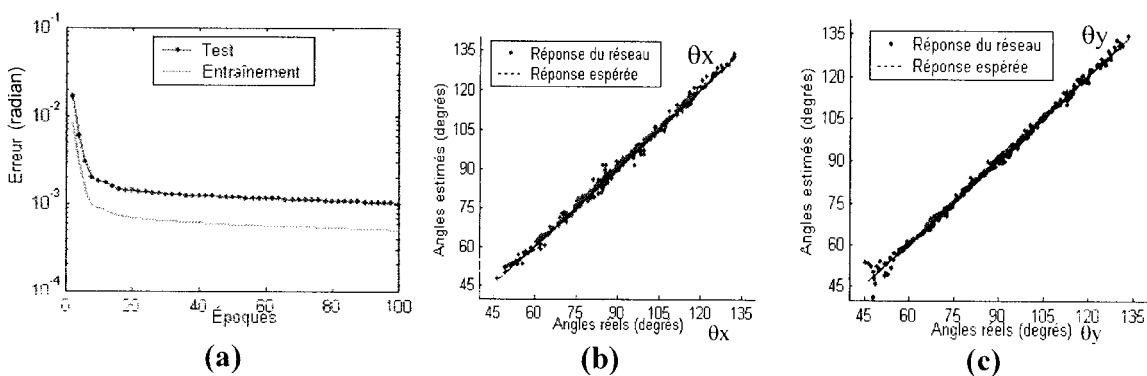


Figure 4.10 Cycles d'apprentissage : (a) Erreur. (b) et (c) Test du réseau

Performance : le réseau est testé sur toutes les données expérimentales prises dans le cône d'étude. La Figure 4.11 présente l'erreur quadratique moyenne sur l'estimation de la direction d'arrivée en fonction de l'ouverture du cône dans lequel arrivent les signaux.

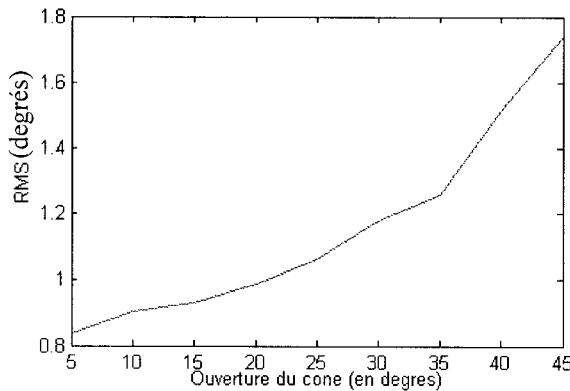


Figure 4.11 Performance de la méthode 2D

Tout comme pour la méthode 1D, avec la méthode 2D nous ne pouvons pas créer une fonction de régression parfaite. Cependant la méthode 2D offre de meilleurs résultats sur tout le cône d'étude que la méthode 1D. Comme les différences entre la théorie et l'expérience proviennent de la modélisation du gain, il était prévisible que la méthode 2D soit meilleure sur des données réelles car le réseau « apprend » la fonction de gain en deux dimensions et permet donc de mieux s'adapter aux anomalies des antennes.

4.2.3 Classification

Choix de la taille du réseau : Comme l'illustre la Figure 4.12, c'est un MLP avec 8 neurones sur la couche cachée qui a été retenu.

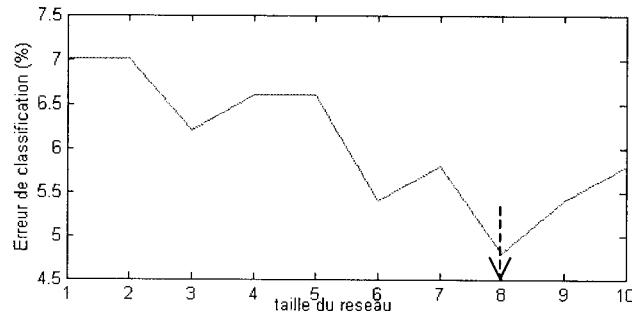


Figure 4.12 Erreur de classification (type 1 + type 2) en % en fonction du nombre de neurones sur la couche cachée.

A présent, nous allons mesurer la performance du système global sous différentes conditions de bruit. Le système global est constitué du réseau de neurones pour la régression avec la méthode 2D décrit à la section 4.2.2 et d'un réseau de fiabilité tel que décrit précédemment.

4.3 Performance du système

Nous allons transformer les données expérimentales en rajoutant des bruits gaussiens indépendants pour chacune des 6 sorties, avant de les normaliser. Puis nous allons filtrer chacune de ces données bruitées avec un filtre par moyenne sur 30 points. C'est avec ces dernières données que nous allons entraîner et tester le système.

Les réseaux de régression et de fiabilité sont entraînés avec 1000 points et testés sur l'ensemble des données. Au Chapitre 3, le rapport signal sur bruit était mesuré par rapports aux signaux du prototype 1 mais comme à présent nous ne travaillons qu'avec le prototype 2, le RSB correspond vraiment au rapport signal sur bruit des signaux du prototype 2. Pour comparer, à bruit égal :

$$\text{RSB}_{\text{proto1}} = \text{RSB}_{\text{proto2}} - 1.3 \text{ dB} \quad (4.2)$$

La Figure 4.13 présente l'erreur quadratique moyenne sur l'estimation de la direction d'arrivée en fonction de l'ouverture du cône dans lequel arrivent les signaux pour différents niveaux de bruit.

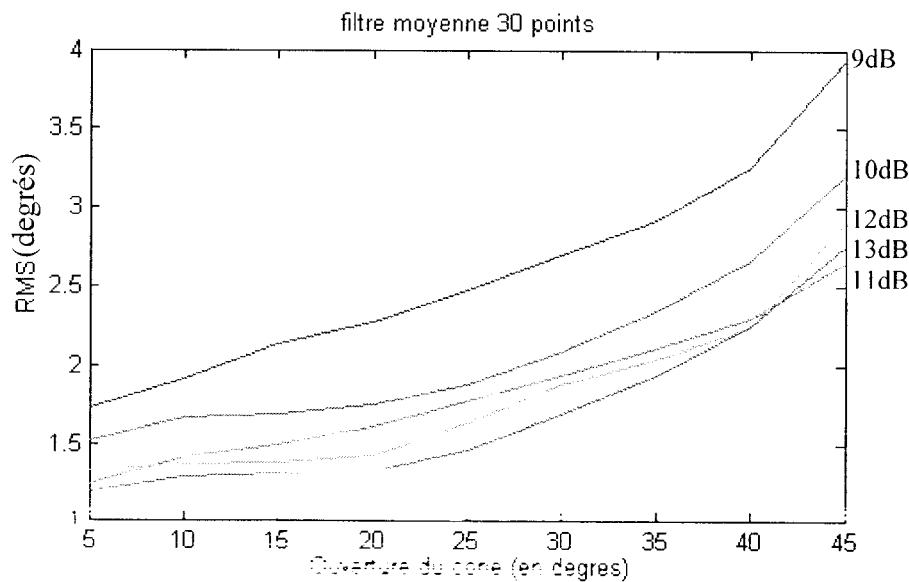


Figure 4.13 Précision du système pour différentes ouvertures du cône d'arrivée

La Figure 4.14 (a) présente les erreurs de classification de type 1 et 2 telles que décrites à la section 3.3. La Figure 4.14(b) représente la classification des données expérimentales en fonction des angles d'arrivée (θ_x , θ_y) pour un RSB de 10dB. La Figure 4.15 présente les erreurs de type 1 et 2 pour un RSB de 10dB en fonction de l'ouverture de cône dans lequel les signaux arrivent.

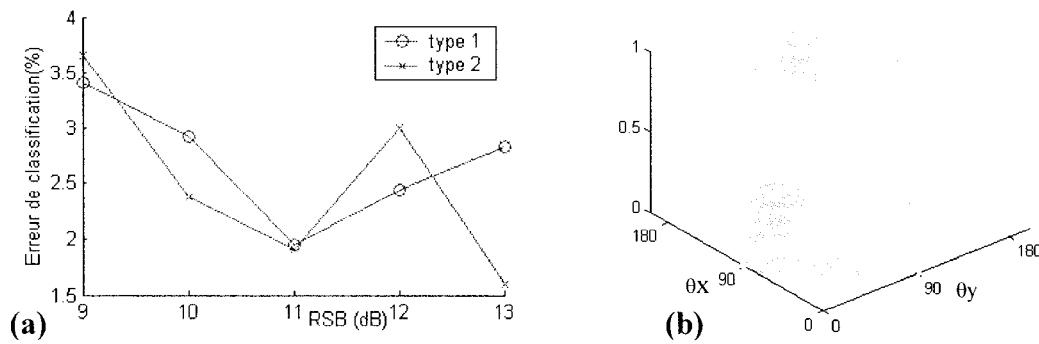


Figure 4.14 (a) Erreur de classification en fonction du bruit. **(b)** Classification pour RSB=10dB

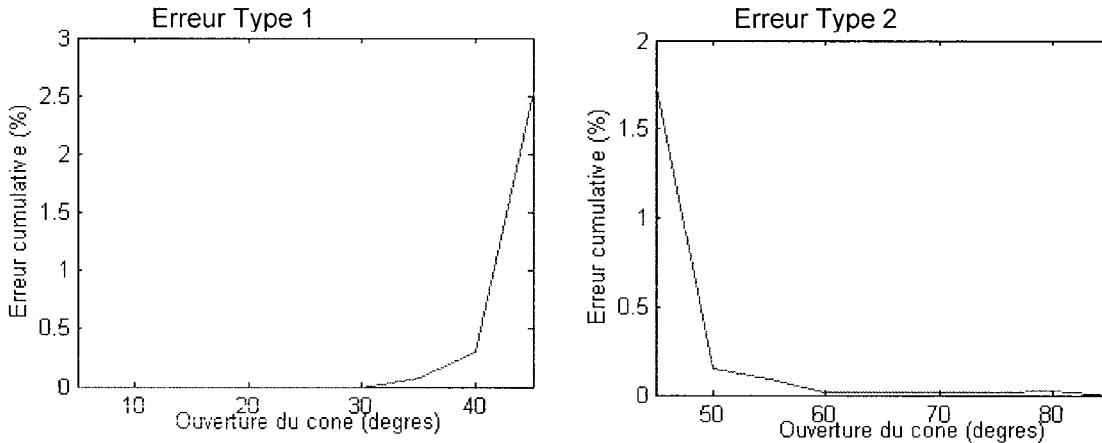


Figure 4.15 Répartition angulaire des erreurs pour un RSB=10 dB

CHAPITRE V

Conclusion et contributions

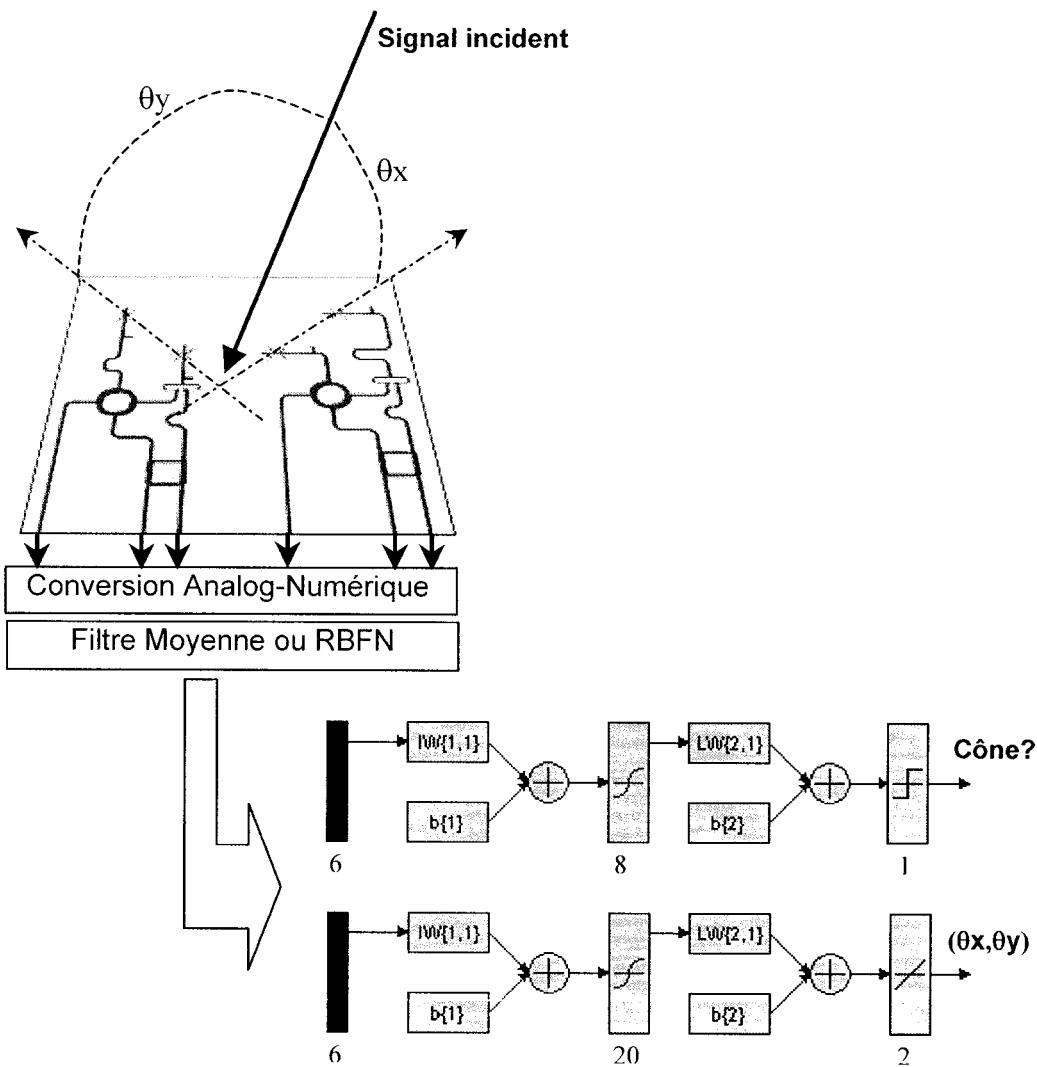


Figure 5.1 Système d'estimation de la direction d'arrivée d'un signal

La Figure 5.1 résume les caractéristiques du système que nous avons élaboré pour détecter l'angle d'arrivée d'un faisceau. Même si le système n'a pas rejoint nos objectifs qui étaient de détecter la direction à 1° près dans tout le cône d'étude, il demeure très performant et novateur.

Du point de vue de la conception, nous avons continuellement recherché les solutions les plus simples afin d'arriver à nos fins. La création du réseau d'antennes et du combinateur a été basée sur une analyse qualitative de l'information pertinente qu'il faudrait fournir aux réseaux de neurones. Cette analyse nous a conduit à un réseau de quatre antennes séparées de moins d'une longueur d'onde les unes des autres et générant 6 signaux. L'élaboration des réseaux de neurones a été guidée par une optimisation du dilemme biais/variance inhérent au choix de la taille a priori des réseaux à utiliser.

En ce qui concerne la performance, pour un RSB de 10 dB, ce système détecte la direction d'arrivée avec une erreur angulaire inférieure de 3° et sa précision atteint 1.5° pour des directions proches de la verticale. De plus, cette réponse est fiable puisque les signaux en dehors du cône d'étude sont détectés dans plus de 97% des cas et ce taux augmente quand la direction d'arrivée s'éloigne de la verticale.

Dans le futur, les performances pourraient être améliorées si l'on concevait un système de poursuite de faisceau. En effet, comme notre système donne une direction à moins de 4° près, nous pourrions imaginer un système mécanique qui orienterait le réseau vers cette direction. A l'étape suivante, le signal d'arrivée serait alors dans un cône de 3° autour de la verticale et le système pourrait, dans cette nouvelle configuration, fournir une réponse à 1.5° près.

Les principales contributions de notre travail sont les suivantes.

- Nous avons proposé une méthode audacieuse et performante pour la création de l'architecture du réseau d'antennes.
- Nous avons défini les critères qu'il faut prendre en compte pour choisir et réaliser un filtre qui augmente le rapport signal sur bruit

- Nous avons optimisé et entraîné des réseaux de neurones pour donner l'angle d'arrivée du signal et s'assurer que la direction d'arrivée était réellement dans le cône d'étude.
- Enfin nous avons validé le système à partir de données expérimentales provenant de notre réseau d'antennes.

RÉFÉRENCES

[Charpentier (1999)]

E.Chapentier, J.J. Laurin. “An implementation of a direction-finding Antenna for mobile communication using a Neural Network”.

IEEE Trans.Antennas Propagat., vol 47, pp 1152-1159, July 1999.

[Duda, Hart, Stork (1998)]

Richard O. Duda, Peter E. Hart and David G. Stork

Pattern Classification (2nd edition). Chapitre 4.3 et 6.8.

John Wiley & Sons, Inc.1998.

[Duda, Hart, Stork (Chap 6.9)]

Richard O. Duda, Peter E. Hart and David G. Stork

Pattern Classification (2nd edition).Chapitre 6.9.

John Wiley & Sons, Inc.1998.

[El Zooghby, (1997)]

A.H. El Zooghby, C.G. Christodoulou, M. Georgopoulos.

“Performance of a radial-basis function network for direction of arrival estimation with antenne arrays”.

IEEE Trans.Antennas Propagat., vol 45, pp. 1611-1617, November 1997.

[Fonseca (03)]

Nelson Fonseca, Jean-Jacques Laurin (directeur)

Maîtrise ès Sciences Appliquées de L’École Polytechnique de Montréal, « Étude, conception et modélisation d’antennes à polarisation circulaire. Application à un système de poursuite pour télécommunications par satellites en basse orbite » .

[Gavish (1995)]

M. Gavish, and A.J. Weiss, "Array Geometry Optimization for Ambiguity Resolution in Direction Finding",

Proceedings of the Eighteenth Convention of Electrical and Electronics Engineers in Israel, Tel Aviv, Israel, March 7-8, 1995.

[Goodwin (1976)]

"Ambiguity-resistant three and four channel interferometers",

Naval Res Lab Rep.8005 Septembre 1976

[Haykin, chap4]

Simon Haykin

Neural Network, a comprehensive foundation. 2nd edition. Chapter 4.

[Haykin, S. (1999)]

Haykin. Neural Networks: A Comprehensive Foundation, second edition. Prentice Hall, 1994.

[Hoyle (2002)]

V.A. Hoyle, G. Lachapelle, M.E. Cannon, C. Wang

Low-Cost GPS receivers and their feasibility for attitude determination.

The institute of navigation national technical meeting, San Diego, CA. January 2002.

[Huang (1991)]

J.Huang et A.C.Densmore, « Microstrip Yagi array antenna for mobile satellite vehicle application »,

IEEE Trans.Antennas Propagat., vol 39, pp. 1024-1030, July 1991.

[Johnson 1993]

Richard C. Johnson

Antenna Engineering Handbook, 3rd Edition

McGraw Hill, January 1993

[Laurin (02)]

Jean-Jacques Laurin, Michael Coudyser, Jean-Jules Brault, Yansheng Xu

A Direction Finding Antenna Based on Neural Networks for Space Applications

12^{èmes} Journées Internationales de Nice sur les Antennes

[Levenberg (1944)]

Kenneth Levenberg.

A method for the solution of certain non-linear problems in least squares.

Quarterly Journal of Applied Mathematics, II(2):164-168, 1944.

[Lo (1994)]

T.Lo,H. Leung, J. Litva

Radial basis function network for direction of arrivals estimation

Signal Processing letters, vol 1, pp.45-47, February 1994.

[Marquardt (1963)]

Donald W. Marquardt.

An algorithm for least-squares estimation of non-linear parameters.

Journal of the Society for Industrial and Applied Mathematics, 11(2):431-441, 1963.

[McCulloch, Pitts (1943)]

McCulloch, D. W. and Pitts, W. "A logical calculus of the ideas immanent in nervous activity". In *Bull. Math. Biophysics*, volume 5, pp.115-133. 1943.

[Nakano (92)]

H. Nakano, K. Nakayama, H. Mimaki, J. Yamauchi and K. Hirose
Single-arm spiral slot antenna fed by a triplate transmission line. Electronics Letters, vol. 28 No.22. (Oct 1992).

[Parzen (1962)]

Emanuel Parzen.
 "On estimation of a probability density function and mode".
Annals of Mathematical Statistics, pp. 1065-1076, 1962.

[Rumelhart, Hinton, Williams]

Rumelhart, D.E, G.E. Hinton, R.J Williams
 "Learning representation of back-propagation errors".
Nature (London), 1986, vol. 323, pp. 533-536.

[Rosenblatt(1962)]

F. Rosenblatt. *Principles of Neurodynamics*. Academic Press, 1962.

[Shieh (2000)]

Direction of arrival estimation based on phase differences using neural fuzzy network
 C-S. Shieh, C-T. Lin.
IEEE Trans. Antennas Propagat., vol 48, pp1115-1124, July 2000.

ANNEXE

Fonction S(d1,d2):

```

function y=esperance(d1,d2)

%%%%%
% dx et dt sont les pas d'intégration
%
%%%%%
dx=1/100;
y=0;

for i=0:(2/dx)

    x=sqrt(2)*i*dx-sqrt(2);
    s=0;

    if x<0
        u=-x-sqrt(2)/2;
        t2max=acos(u);
        t2=pi/4;
        dt2=1/500;
        while (t2<=t2max)
            t1=acos(cos(t2)+x);
            dt1=abs(acos(cos(t2)+x)-acos(cos(t2)+x+sqrt(2)*dx));
            s=s+dt1*dt2;
            t2=t2+dt2;
        end;

        s=s/(pi/2)/(pi/2);
    end;

    if x>=0
        u=x-sqrt(2)/2;
        t1max=acos(u);
        t1=pi/4;
        dt1=1/500;
        while (t1<=t1max)
            t2=acos(cos(t1)-x);
            dt2=abs(acos(cos(t1)-x)-acos(cos(t1)-x-sqrt(2)*dx));
            s=s+dt1*dt2;
            t1=t1+dt1;
        end;
    end;
end;

```

```

end;

s=s/(pi/2)/(pi/2);
end;

y=y+Q(d1,d2,x)*s;
end;

```

Fonction Q(d1,d2,x) :

```

function y=Q(d1,d2,x)
M=(([d1;d2;d2+d1]*x)-floor([d1;d2;d2+d1]*x-1/2)).^2;
y=sum(M);
end;

```

Taylor-Young :

William Henry Young (1863 - 1942).
 Brook Taylor (1685 - 1731).

Définition :

On appelle *polynôme de Taylor* de la fonction f à l'ordre n en x_0 le polynôme :

$$T_{n,f,x_0}(x) = \sum_{i=0}^n \frac{(x-x_0)^i}{i!} f^{(i)}(x_0)$$

Théorème de Taylor-Young :

Le nombre n est un entier strictement positif. Si la fonction f vérifie les deux conditions :

f est $n-1$ fois dérivable dans un voisinage v de x_0 ,

f est n fois dérivable en x_0 ,

alors, le polynôme de Taylor de f à l'ordre n en x_0 est une approximation polynomiale de f en x_0 :

$$T_{n,f,x_0}(x) = f(x_0) + \frac{f'(x_0)}{1!}(x-x_0) + \frac{f''(x_0)}{2!}(x-x_0)^2 + \dots + \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n$$

Loi Gaussienne

On la doit à Moivre qui, en 1738, a trouvé cette loi comme limite de la loi binomiale. On la retrouve comme modèle le plus courant pour les distributions d'erreurs de mesure autour d'une valeur « vraie ». Elle joue aussi un rôle important en terme de comportement asymptotique des autres lois de probabilités, comme le montre le théorème central limite. On dit qu'une variable aléatoire X suit une loi gaussienne de moyenne μ de variance σ^2 , appelée loi normale notée $N(\mu, \sigma^2)$ si

$$P(X \in [x, x+dx]) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{\sigma^2}\right) dx$$