

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE

UNIVERSITE SAAD DAHLEB DE BLIDA
Faculté des Sciences de l'Ingénieur
Département D'Aéronautique



**Mémoire de fin d'études en vue de l'obtention
du diplôme d'ingénieur d'état en Aéronautique**

**Option : Installation
THEME**

**Etude de Réseau d'Antenne Linéaire pour
Estimation et Poursuite de la DOA basé sur
le LMS et le RLS**

Réalisé par :

M. KHEDROUCHE

Ramzi

M. DAHMANI

Abdelkader

Encadré par :

M^{lle}. DOUDOU.F

ANNEE UNIVERSITAIRE 2005 /2006

Résumé

Dans cette thèse dont le titre est intitulé, étude de réseau d'antenne linéaire pour estimation et poursuite de la DOA basé sur le LMS et le RLS, on a fait une étude mathématique de ces deux algorithmes et leurs différentes procédures, puis on les a appliqués à un réseau d'antenne à fin d'obtenir une antenne intelligente, capable de fonctionner de manière autonome et capable de poursuivre les variations temporelles que ce soit pour les signaux ou pour la DOA, et qui permet de réaliser le formage du lobe directif avec un temps de calcul et un coût relativement faible.

En fin on a simulé sur MATLAB, les différents types de ces algorithmes pour obtenir la meilleure conception des antennes à pointage, avec applications dans le domaine de réseau adaptatif.



REMERCIEMENTS

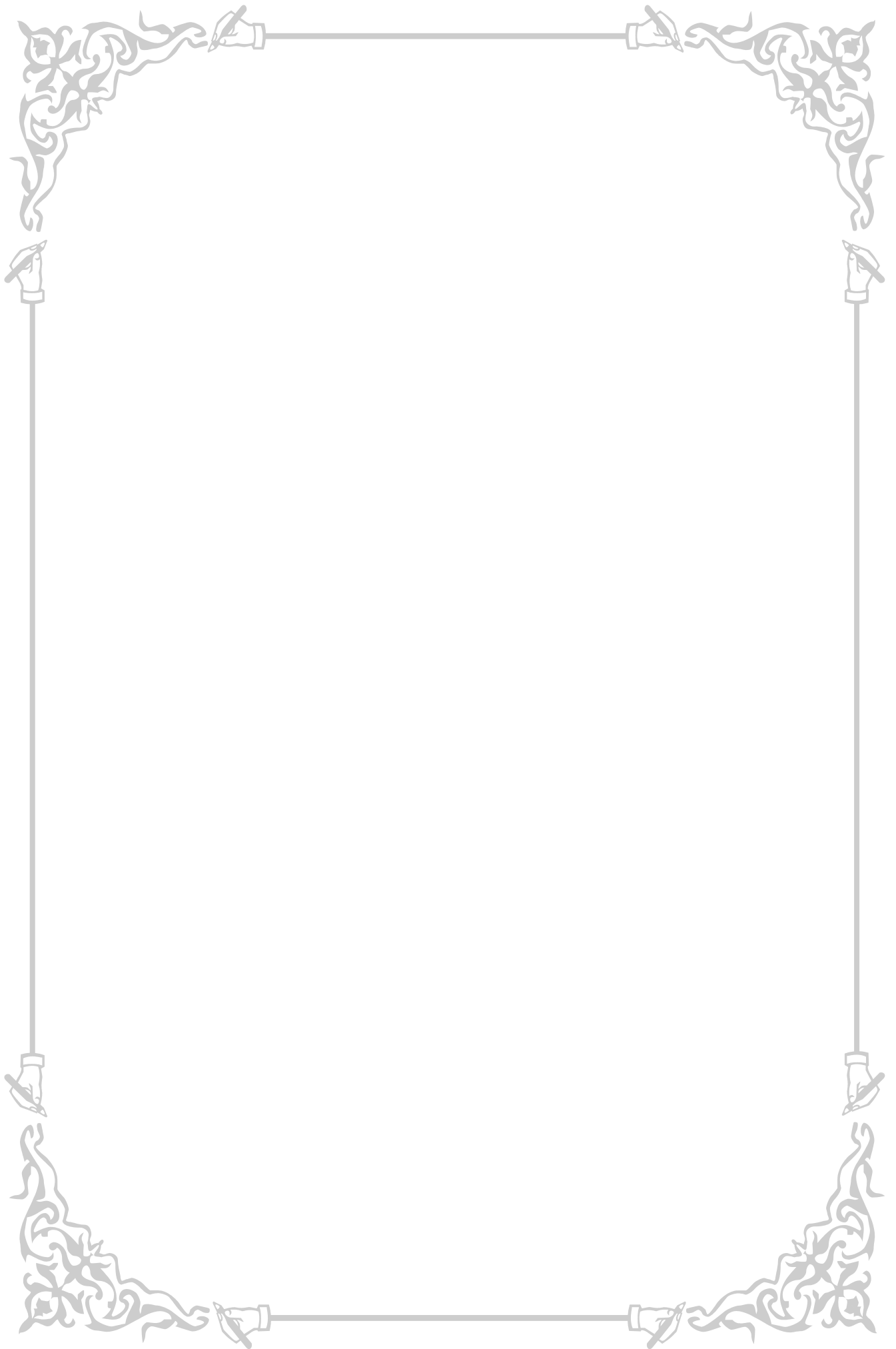
*Nous remercions tout d'abord dieu tout puissant tout clément qui
nous a aidé à élaborer ce modeste travail
et d'aboutir à ce niveau d'études.*

*Nous exprimons notre profonde et respectueuse gratitude
et reconnaissance :*

A notre promotrice M^{elle}.DOUDOU FAIZA

A Monsieur DJALLEL.

*Nous exprimons nos remerciements les plus sincères à l'ensemble
des enseignants du département d'Aéronautique.*









Dédicace

*Je dédie ce modeste travail, synonyme de
concrétisation de tous mes efforts fournis pendant ces
cinq dernières années :*

A ma très chère Mère.

A mon chère Père

A ma grand-mère.

A mes frères.

A mes sœurs.

A mon binôme khadrach badrach.

A tout mes Amis :

*Djalal, Morad, Fathi, kader, Sid ahmed, Morsli, Mansour, Houari,
Benamer, Benoumar, Lebbad, Messaoud, Mourad, Mouf, Nabil, ,
Salim Kabranne, Seddik, Tounsi, Zaki, Zahir, Hassene, el
mareg, benslama.*

DAHMANI.AEK

Dédicace

Après aimer et aider, remercier est assurément le troisième plus beau verbe dans toutes les langues.

A ma très chère mère ma raison de vivre, symboles de courage et de sacrifices, Que dieu la garde.

A la mémoire de mon père.

A mes oncles.

A ma arrière grande mère.

A ma future femme et petite famille.

A mon binôme Kadandou.houwa.

A toute les familles Khedrouche et Bouziad.

A tout mes Amis :

Adel, Hamza, Riad, Merouane, Hamido, Larsen, Farid, Khalel, Lebbad, Besslama, El mareg, Mouf, Fethi, Seddik, Tounsi, Hacenne, Zikos, Zahir, Fares. El haba, Hadji, Benaoui, djilali, Anis, El aiid , ja3fer , kamatchou.

KHEDROUCHE. R

Table des matières

Remerciements	I
Dédicaces	II
Résumé	III
Notations	IV
Abréviations et Acronymes	VI
Table des figures	VII
Liste des tableaux	X
Introduction	1
Chapitre N°01 : Traitement numérique du signal	
1.1. Généralité.....	2
1.1.1. Introduction.....	2
1.1.2. Définition de la stationnarité.....	2
1.1.2.1. Définition de la stationnarité.....	2
1.1.2.2. Définition d'un signal du second ordre.....	3
1.1.2.3. Définition de la stationnarité à l'ordre 2.....	3
1.1.2.4. Définition de l'ergodicité d'un signal stationnaire à l'ordre 2.....	3
1.2. Analyse Spectrale.....	4
1.2.1. Problématique de l'Analyse spectrale.....	4
1.2.2. L'estimation spectrale non paramétrique.....	4
1.2.2.1. Périodogramme, Corrélogramme.....	4
1.2.2.2. Méthode du minimum de variance (méthode de Capon).....	5
1.2.3. L'estimation spectrale par décomposition harmonique.....	6
1.2.3.1. Méthode de Pisarenko.....	6
1.2.4. L'estimation spectrale paramétrique.....	7
1.2.4.1. Le modèle auto régressif - modèle AR -.....	7
1.2.4.2. Le modèle à moyenne ajustée - modèle MA -.....	9
1.2.4.3. Le modèle auto régressif à moyenne ajustée - modèle ARMA-.....	10
1.3. Définition de quelques estimateurs statistiques.....	10
1.4. Régression linéaire.....	11
1.4.1. Mesure, modèle et écart.....	11
1.4.2. Minimisation de l'écart quadratique.....	12
1.4.3. Équations de la régression linéaire.....	13
1.5. Conclusion.....	13
Chapitre N°02 : Filtrage adaptatif	
2.1. Introduction.....	14
2.1.1. Objectifs et déroulement du travail.....	14
2.1.2. Classification des filtres.....	14
2.1.3. Choix de l'algorithme.....	15

2.2. Filtrage linéaire optimal Filtrage de Wiener.....	15
2.2.1. Relations d'orthogonalité - Equation de Wiener-Hopf.....	16
2.2.2. Application.....	17
2.2.2.1 Identification de Systèmes.....	18
2.2.2.2. Prédiction.....	18
2.2.2.3. Modélisation inverse (égalisation, déconvolution).....	18
2.2.2.4. Annulation d'interférences (annulation d'écho, de bruit).....	19
2.3. Algorithme du gradient.....	19
2.3.1. Convergence de l'algorithme du gradient.....	20
2.3.1.1. Conditions de convergence pour un pas μ constant.....	21
2.3.1.2. Rapidité de convergence.....	21
2.3.1.3. Pas optimal.....	22
2.3.1.4. Pas optimal et conditionnement de la matrice d'autocorrélation.....	23
2.3.2. Autre présentation de la méthode du gradient.....	24
2.4. Algorithme du gradient stochastique.....	24
2.4.1. Convergence et stabilité.....	26
2.4.1.1. Comportement de l'erreur.....	26
2.4.1.2. Convergence de l'algorithme à pas variable.....	27
2.4.2. Poursuite – Adaptativité.....	28
2.4.3. Algorithme du signe.....	29
2.4.4. Algorithme LMS normalisé.....	29
2.5. Formulation du filtrage de Wiener dans le cadre des moindres carrés.....	31
2.5.1. Critère des moindres carrés pour un bloc de données de N points $(u_0, u_1, \dots, u_{N-1})$	31
2.5.1.1. Théorème de projection.....	33
2.5.2. Solution de l'équation normale.....	33
2.6. Algorithme des moindres carrés récursifs.....	34
2.6.1. Construction de l'algorithme des MCR.....	34
2.6.1.1. Lemmes d'inversion matriciels.....	35
2.6.2. Problèmes d'initialisation.....	37
2.6.3. Version adaptative – pondération exponentielle.....	38
2.6.4. Equation récursive sur le critère somme quadratique.....	38
2.6.5. Convergence de l'algorithme, en moyenne.....	39
2.6.6. Erreur quadratique moyenne.....	40
2.6.7. Algorithmes sous optimaux.....	40
2.7. Conclusion.....	40

Chapitre N°03 : Les antennes intelligentes

3.1. Rangée d'antennes linéaires.....	41
3.1.1. Introduction aux Antennes.....	41
3.1.2. Caractéristiques d'une antenne.....	41
3.1.2.1. Principe de réciprocité.....	41
3.1.2.2. Antenne isotrope.....	41
3.1.2.3. Directivité.....	42
3.1.2.4. Gain en puissance et rendement.....	42
3.1.4. Émission et réception d'antenne.....	43

3.1.4.1. Émission.....	43
3.1.4.1.1. Définition.....	43
3.1.4.1.2. Représentation.....	43
3.1.4.1.3. Caractéristiques principales.....	44
3.1.4.1.4 Difficultés.....	45
3.1.4.2. Réception.....	45
3.1.4.2. 1. Définition.....	45
3.1.4.2. 2. Représentation.....	45
3.1.4.2. 3. Exemples d'application.....	45
3.1.4.2. 4. Caractéristiques principales.....	45
3.1.4.2. 5. Difficultés.....	46
3.1.4.3. Émission et réception.....	46
3.1.4.3.1. Définition.....	46
3.1.4.3.2. Représentation.....	46
3.1.4.3.3. Caractéristiques principales.....	47
3.1.4.3.4. Difficultés.....	47
3.1.5. Réseaux linéaire.....	47
3.1.5.1. Réseaux linéaires dispersifs.....	47
3.1.5.1.1. Principe.....	47
3.1.5.1.2. Balayage angulaire standard.....	48
3.1.5.1.3. Augmentation du balayage angulaire.....	49
3.1.6. Antenne à balayage électronique.....	50
3.1.6. 1. Définition.....	50
3.1.6. 2. Représentation.....	50
3.1.6. 3. Exemples d'application.....	50
3.1.7. Etude théorique de rangée d'antennes linéaire.....	51
3.1.7. 1. Introduction.....	51
3.1.7. 1. 1. Généralités.....	51
3.1.7. 1. 2. Déphasage De translation.....	52
3.1.7. 2. Rangées unidimensionnelles.....	53
3.1.7. 2.1. Facteur de rangée.....	53
3.1.7. 2.2. La transformée en Z.....	54
3.1.7. 3. Rangées unidimensionnelles uniformes.....	55
3.1.7.4. Direction De Rangée.....	56
3.1.7.5. La comparaison entre les traitements matriciels des rangées et le traitement de signa numérique.....	58
3.2. Les antennes intelligentes.....	59
3.2.1. Définition des antennes intelligentes	59
3.2.2. La Conception d'antenne intelligente	59
3.2.3. Avantages des antennes intelligentes	60
3.2.4. Rangée D'Antenne Adaptative	60
3.2.5. Structure d'une Rangée D'Antenne Adaptative	61
3.2.6. Les algorithmes adaptatifs	61
3.3. Formation de voies optimales (statistically optimum beamforming).....	62
3.3.1. Application de filtre adaptif au rangée d'antenne linéaire et annulation des interférenc	62

3.3.2. Beamforming Adaptif.....	67
3.3.2.2. Algorithmes Temporels De Référence.....	67
3.3.2.1. Beamforming Conventionnel.....	67
3.3.2.2.1. <i>Beamforming LMS</i>	68
3.3.2.2.2. <i>Beamforming RLS</i>	70
3.3.2.2. La complexité informatique de l'algorithme de RLS.....	72
3.4. Conclusion.....	72

Chapitre N°04 : Simulation

4.1. Introduction.....	73
4.2. Simulations et commentaires.....	73
4.3. Organigramme et interface de simulation.....	74
4.4. Simulation et commentaires pour AOA (DOA) constante.....	76
4.4.1. Simulation n°01.....	77
4.4.2. Simulation n°02.....	103
4.4.2. 1. Partie 1.....	103
4.4.2. 2. Partie 2.....	109
4.5. Simulation et commentaires pour AOA (DOA) variable.....	113
4.5. 1. Simulation n°03.....	114
4.5. 1. 1. Partie1.....	114
4.5. 1. 2. Partie2.....	116
4.6. Conclusion.....	117
Conclusion et perspective.....	118

Table des figures

Figure.1.1 : évolution du signal.....	8
Figure.1.2 : Régression linéaire.....	12
Figure.2.1 : filtrage optimal.....	15
Figure.2.2 : Identification de Systèmes.....	18
Figure.2.3 : Prédiction.....	18
Figure.2.4 : Modélisation inverse.....	18
Figure.2.5 : Annulation d'interférences.....	19
Figure.2.6 : Dérivée de l'erreur quadratique.....	20
Figure.2.7 : Théorème de projection.....	33
Figure.3.1 : Représentation d'une antenne d'émission seule.....	43
Figure .3.2: Représentation d'une antenne de réception seule.....	45
Figure .3.3: Représentation d'une antenne d'émission et de réception.....	46
Figure.3.4 : Réseau linéaire de 3 dipôles.....	47
Figure.3.5 : Réseau linéaire dispersif : description.....	48
Figure .3.6 : Réseau linéaire dispersif avec ligne à méandres.....	49
Figure .3.7 : Représentation d'une antenne à balayage électronique.....	50
Figure .3.8 : Configurations typiques de rangée.....	51
Figure .3.9 : Antennes traduites.....	52
Figure.3.10 : Rangée unidimensionnelle.....	53
Figure.3.11 : Facteur de rangée et modèle angulaire de la rangée 8-element uniforme.....	56
Figure.3.12 : Balayage par traduction.....	57
Figure.3.13 : Modèle de faisceau d'exemple d'une rangée d'antenne adaptative.....	60
Figure.3.14 : Schéma représentatif d'une antenne adaptative.....	61
Figure.3.15 : Rangée d'antennes linéaire reçoit plusieurs ondes planes.....	62
Figure.3.16 : Rangée d'antennes linéaire reçoit une seule onde plane.....	63
Figure.3.17 : Schéma synoptique de récupération de l'enveloppe complexe.....	64
Figure.3.18 : Schéma fonctionnel De Beamforming Adaptatif.....	67
Figure 4.1 : Organigramme de simulation du programme (sur Matlab).....	74
Figure 4.2 : Interface de simulation du programme (sur Matlab).....	75
Figure 4.3 : Initialisation des données pour AOA constant.....	76
Figure 4.4 : Choix de l'algorithme.....	77
Figure.4.5 : poursuite de la phase par LMS.....	78
Figure.4.6 : Poursuite de la phase par RLS sans facteur d'oublie.....	78
Figure.4.7 : Poursuite de la phase par RLS à facteur d'oublie constant.....	79
Figure.4.8 : Poursuite de la phase par RLS à facteur d'oublie variable.....	79
Figure.4.9 : Poursuite de la phase par RLS à facteur d'oublie a trace constante.....	80
Figure.4.10 : poursuite de la magnitude par LMS.....	81
Figure.4.11 : Poursuite de la magnitude par RLS sans facteur d'oublie.....	81
Figure.4.12 : Poursuite de la magnitude par RLS à facteur d'oublie constant.....	82
Figure.4.13 : Poursuite de la magnitude par RLS à facteur d'oublie variable.....	82
Figure.4.14 : Poursuite d la magnitude par RLS à facteur d'oublie a trace constante.....	83
Figure.4.15 : Evolution de l'erreur quadratique par LMS.....	84
Figure.4.16 : Evolution de l'erreur quadratique par RLS sans facteur d'oublie.....	84
Figure.4.17 : Evolution de l'erreur quadratique par RLS à facteur d'oublie constant.....	85
Figure.4.18 : Evolution de la l'erreur quadratique par RLS à facteur d'oublie variable.....	85
Figure.4.19 : Evolution de l'erreur quadratique par RLS à facteur d'oublie a trace constante.....	86
Figure.4.20 : réponse du filtre LMS.....	87

Figure.4.21 : réponse du filtre RLS a facteur d'oublie constant.....	87
Figure.4.22 : réponse du filtre RLS facteur d'oublie variable.....	88
Figure.4.23 : réponse du filtre RLS a trace constante.....	88
Figure.4.24 : évolution des coefficient complexe du filtre LMS.....	89
Figure.4.25 : évolution des coefficient complexe du filtre RLS sans oublie.....	90
Figure.4.26 : évolution des coefficient complexe du filtre RLS a facteur d'oublie constant.....	90
Figure.4.27 : évolution des coefficient complexe du filtre RLS a facteur d'oublie variable.....	91
Figure.4.28 : évolution des coefficient complexe du filtre RLS a trace constante.....	91
Figure.4.29 : réponse des l'antennes au filtre LMS.....	92
Figure.4.30 : réponse des l'antennes au filtre RLS sans oublie.....	93
Figure.4.31 : réponse des antennes au filtre RLS a facteur d'oublie constant.....	93
Figure.4.32 : réponse des antennes au filtre RLS a facteur d'oublie variable.....	94
Figure.4.33 : réponse des antennes au filtre RLS a trace constante.....	94
Figure.4.34 : évolution du facteur d'oublie variable.....	95
Figure.4.35 : évolution du gain d'adaptation pour RLS sans oublie.....	96
Figure.4.36 : évolution du gain d'adaptation pour RLS a facteur d'oublie constant.....	96
Figure.4.37 : évolution du gain d'adaptation pour RLS a facteur d'oublie variable.....	97
Figure.4.38 : évolution du gain d'adaptation pour RLS a trace constante.....	97
Figure.4.39 : évolution de trace pour RLS sans facteur d'oublie.....	98
Figure.4.40 : évolution de la trace pour RLS a trace constante.....	99
Figure.4.41 : tracé des lobes pour une adaptation LMS.....	100
Figure.4.42 : tracé des lobes pour une adaptation LMS.....	100
Figure.4.43 : tracé des lobes pour une adaptation RLS a facteur d'oublie constant.....	101
Figure.4.44 : tracé des lobes pour une adaptation RLS a facteur d'oublie variable.....	101
Figure.4.45 : tracé des lobes pour une adaptation RLS a trace constante.....	102
Figure.4.46 : poursuite de la phase par LMS.....	103
Figure.4.47 : poursuite de la phase par RLS a facteur d'oublie variable.....	104
Figure.4.48 : poursuite de la magnitude par LMS.....	104
Figure.4.49 : poursuite de la magnitude par RLS a facteur d'oublie variable.....	105
Figure.4.50 : évolution de l'erreur par LMS.....	105
Figure.4.51 : évolution de l'erreur par RLS a facteur d'oublie variable.....	106
Figure.4.52 : évolution des coefficient complexe du filtre LMS.....	106
Figure.4.53 : évolution des coefficient complexe du filtre RLS a facteur d'oublie variable.....	107
Figure.4.54 : tracé des lobes pour une adaptation LMS.....	107
Figure.4.55 : tracé des lobes pour une adaptation RLS a facteur d'oublie variable.....	108
Figure.4.56 : poursuite de la phase par LMS.....	109
Figure.4.57 : poursuite de la phase par RLS a facteur d'oublie variable.....	110
Figure.4.58 : évolution de l'erreur par LMS.....	110
Figure.4.59 : évolution de l'erreur par RLS a facteur d'oublie variable.....	111
Figure.4.60 : tracé des lobes pour une adaptation LMS.....	111
Figure.4.61 : tracé des lobes pour une adaptation RLS a facteur d'oublie variable.....	112
Figure.4.62 : Initialisation des données pour AOA variable.....	113
Figure.4.63 : tracé super poser du balayage des lobes pour une adaptation RLS a trace constante....	114
Figure.4.64 : tracé du balayage des lobes à chaque variation pour une adaptation RLS a trace constante.....	115
Figure.4.65 : tracé super poser du balayage des lobes pour une adaptation RLS a trace constante.....	116
Figure.4.66 : tracé du balayage des lobes a chaque variation pour une adaptation RLS a trace constante.....	117

Liste des tableaux

Tableau.3.1 : La comparaison entre les traitements matriciels des rangées et le traitement de signal Numérique.....	58
Tableau.3.2 : La complexité informatique de l'algorithme de RLS.....	72
Tableau.4.1 : initialisation de la simulation n°01.....	77
Tableau.4.2 : valeurs des angles vrais et estimés pour la simulation n°02.....	102
Tableau.4.3 : initialisation de la simulation n°02(partie1).....	103
Tableau.4.4 : valeurs des angles vrais et estimés pour la simulation n°02(partie1).....	108
Tableau.4.5 : initialisation de la simulation n°02 (partie2).....	109
Tableau.4.6 : valeurs des angles vrais et estimés pour la simulation n°02(partie2).....	112
Tableau.4.7 : initialisation de la simulation n°03(partie1).....	114
Tableau.4.8 : valeurs des angles vrais et estimés pour la simulation n°03(partie1).....	115
Tableau.4.9 : initialisation de la simulation n°03(partie2).....	116
Tableau.4.10 : valeurs des angles vrais et estimés pour la simulation n°03(partie2).....	117

Introduction

Dans le cadre de notre étude d'utilisation de rangée d'antenne linéaire pour l'estimation et la poursuite de l'angle d'arrivée constant ou variable, à base du LMS et RLS, on a commencé par une présentation du traitement numérique du signal et l'analyse spectrale, puis une étude mathématique des filtres adaptés et une étude des rangées d'antenne et de l'application des filtres adaptés, et en fin on a élaboré un programme qui simule cette étude.

Dans le premier chapitre on a montré l'avantage des systèmes numériques sur leurs homologues analogiques, ainsi qu'une présentation de quelques méthodes d'analyse spectrale et des critères influant sur le choix de leur application qui sont la longueur et la stationnarité des signaux, mais pour des signaux à temps réel on a vu qu'il faut avoir recours au filtrage adaptatif au lieu du filtrage classique.

Dans la première partie du deuxième chapitre nous avons exposé l'approche statistique du problème (filtrage de Wiener), et à cause du fait qu'il soit inadéquat pour les signaux non stationnaires on a présenté la technique adaptative, puis en remplaçant les grandeurs statistiques impliquées par des valeurs instantanées on a abouti à l'algorithme du gradient stochastique (LMS), et en reformulant le problème en termes de moindres carrés on a présenté l'algorithme des moindres carrés récursifs (RLS), avec une étude de leur capacité de convergence, stabilité et poursuite des variations des signaux non stationnaires, et l'élimination du bruit superposé au signal désiré.

Dans le troisième chapitre on a étudié le réseau d'antenne linéaire, le balayage de son lobe résultant, puis on a appliqué les algorithmes adaptatifs sur notre réseau, ce qui nous a conduit à l'antenne intelligente, et cela nous a permis d'estimer et de poursuivre notre AOA.

Dans le dernier chapitre consacré à la simulation des résultats on a pu voir les performances des différentes méthodes d'adaptation avec des résultats qui ont appuyé notre étude théorique.

Chapitre N°01

Traitement numerique du signal

1.1. Généralité :

1.1.1. Introduction :

Le traitement (numérique) du signal consiste en un ensemble de théories et de méthodes, relativement indépendantes du signal traité, permettant de créer, d'analyser, de modifier, de classifier, et finalement de reconnaître les signaux.

Il s'agit donc d'une science appliquée, puisque le signal numérique n'existe pour ainsi dire pas dans la nature. Il est une invention de l'homme, qui a pour but principal de permettre une manipulation aisée de signaux analogiques à l'aide de calculateurs numériques. Ses applications sont nombreuses dans des domaines aussi variés que les télécommunications, le traitement du son musical, le traitement de la parole, le radar, le sonar, l'étude des vibrations sismiques, le contrôle non destructif, l'ingénierie biomédicale, l'imagerie, l'économie (avec l'étude des séries chronologiques), etc.

On peut estimer que cette science de l'ingénieur a réellement vu le jour dans les années 60, avec la découverte d'algorithmes rapides de calcul de la transformée de Fourier discrète par Cooley et Tukey. Les ouvrages de base ont été publiés entre la fin des années 70 et le début des années 90. Ce n'est que récemment, cependant, que les applications pratiques se sont multipliées, avec l'avènement des technologies numériques (RNIS-CD-GSM-DVD-MP3, etc.) et des processeurs spécialisés en traitement du signal (DSP : digital signal processor).

Les systèmes numériques possèdent sur leurs homologues analogiques un ensemble d'avantages décisifs :

- *Simplicité.*
- *Possibilités de traitement accrues.*
- *Robustesse aux bruits.*
- *Précision et stabilité.*
- *Flexibilité.*

Dans un grand nombre de systèmes numériques, le traitement est défini par un logiciel chargé en mémoire. Il est dès lors très facile de modifier ce traitement, sans devoir modifier la machine qui le réalise. On pense par exemple aux modems numériques actuels, qui peuvent s'adapter facilement aux normes futures par simple programmation.

1.1.2. Définition de la stationnarité :

1.1.2.1. Définition de la stationnarité :

Un signal aléatoire est défini à chaque instant t par la loi de probabilité de son amplitude $X(t)$. Cette loi de probabilité peut s'exprimer par une densité de probabilité $P_X(x, t)$, définie de la manière suivante :

$$P_X(x, t) = \lim_{\Delta x \rightarrow 0} \frac{P_{rob} [x \leq X(t) \leq x + \Delta x]}{\Delta x} \quad (1.1)$$

Le signal est stationnaire (**ANNEXE 1**) si ses propriétés statistiques sont indépendantes du temps, c'est à dire, si sa densité de probabilité est indépendante du temps :

$$P_X(x, t) = P_X(x) \quad (1.2)$$

1.1.2.2. Définition d'un signal du second ordre :

Le signal sera dit du second ordre s'il possède un moment d'ordre 1 appelé valeur moyenne, qui est l'espérance mathématique de $X(t)$, notée $E\{X(t)\}$ et définie par :

$$E\{X(t)\} = \int_{-\infty}^{+\infty} x P_X(x, t) dx \quad (1.3)$$

Et un moment d'ordre 2, appelé fonction de covariance :

$$E\{X(t_1), X(t_2)\} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x_1 x_2 P_X(x_1, x_2; t_1, t_2) dx_1 dx_2 \quad (1.4)$$

Où $P_X(x_1, x_2; t_1, t_2)$ est la densité de probabilité du couple de variables aléatoires $\{X(t_1), X(t_2)\}$.

1.1.2.3. Définition de la stationnarité à l'ordre 2 :

Le caractère de stationnarité peut être limité aux moments du premier et du second ordre, on dit alors que le signal est stationnaire à l'ordre 2. On a alors :

$$E\{X(t)\} = \int_{-\infty}^{+\infty} x P_X(x) dx \quad (1.5)$$

$E\{X(t)\}$ Constante indépendante du temps.

Pour l'ordre 2, l'indépendance du temps s'écrit :

$$P_X(x_1, x_2; t_1, t_2) = P_X(x_1, x_2; 0, t_2 - t_1) = P_X(x_1, x_2; \dagger) ; \text{ ou } \dagger = t_2 - t_1 \quad (1.6)$$

Seul intervient l'écart entre les deux instants d'observation. On introduit alors la fonction d'autocorrélation $R_{XX}(\dagger)$ du signal aléatoire :

$$R_{XX}(\dagger) = E\{X(t) X(t - \dagger)\} \quad (1.7)$$

Le Signal aléatoire $X(t)$ possède aussi une moyenne temporelle m_T définie par :

$$m_T = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^{+T} x(t) dt \quad (1.8)$$

1.1.2.4. Définition de l'ergodicité d'un signal stationnaire à l'ordre 2 :

On dira que le signal est ergodique lorsque que l'on peut confondre la moyenne temporelle m_T avec la moyenne $m_1 = E\{X(t)\}$:

$$m_1 = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^{+T} x(t) dt \quad (1.9)$$

Et lorsque que l'on peut calculer la fonction d'autocorrélation de la même manière, c'est à dire :

$$R_{XX}(\dagger) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^{+T} x(t) x(t - \dagger) dt \quad (1.10)$$

Ce résultat a des conséquences pratiques très importantes car il va permettre d'accéder aux propriétés statistiques du signal à un instant donné à partir de l'observation de ce signal au cours du temps,

Les développements précédents ont essentiellement concerné des signaux à valeurs réelles, Ces signaux correspondent intuitivement et physiquement à la plupart des signaux rencontrés. Lorsque l'on considère une valeur de tension ou de courant et que l'on se place derrière un convertisseur analogique numérique, les signaux sont bien entendu réels, cependant dans bien des cas, en particulier en communications numériques, on sera amené à traiter des enveloppes complexes de signaux modulés (voir aussi la définition du signal analytique).

Il est donc nécessaire, afin de ne pas restreindre la généralité de la suite des algorithmes et méthodes présentées, de considérer dorénavant des signaux à valeurs complexes.

1.2. Analyse Spectrale :

1.2.1. Problématique de l'Analyse spectrale :

La représentation du signal sous la forme $X(t)$ est une démarche naturelle mais elle ne correspond pas forcément à la meilleure représentation physique des signaux rencontrés. En effet, l'individu ou les systèmes électroniques sont souvent plus sensibles à la puissance et à la fréquence des signaux et la représentation du signal sous la forme de sa répartition de puissance en fonction de la fréquence permet, dans bien des cas, d'extraire de manière plus immédiate l'information qui réside dans ce dernier. Le signal est alors représenté par une fonction $P(f)$ appelée densité spectrale de puissance. Le passage de $X(t)$ à $P(f)$ constitue l'Analyse Spectrale. Il existe deux grandes classes de méthodes pour

estimer la densité spectrale de puissance d'un signal $X(t)$:

- Non paramétrique.
- Paramétrique.
- L'estimation spectrale par décomposition harmonique.

1.2.2. L'estimation spectrale non paramétrique :

N'utilise aucune connaissance a priori sur le signal et part uniquement de l'observation de ce dernier.

1.2.2.1. Périodogramme, Corrélogramme :

En considérant un cas numérique, où l'observation du signal $X(t)$ se résume à N valeurs échantillonnées à la période T_e , la densité spectrale peut être estimée :

$$P_{per}(f) = \frac{1}{N \cdot T_e} \left| \sum_{n=0}^{N-1} X(n \cdot T_e) e^{-j2\pi f n T_e} \right|^2 \quad (1.11)$$

Cet estimateur de la densité spectrale de la densité spectrale de puissance du signal $X(n \cdot T_e)$ est appelé périodogramme. Il demande, pour être calculé, la mise au carré de la Transformée de Fourier du signal numérique $X(n \cdot T_e)$ sur N points. Jusqu'à la mise au point de l'algorithme de la FFT, la méthode la plus utilisée consistait à estimer N' valeurs de la fonction d'autocorrélation $R_{xx}(p)$ avec $N' < N$ et à calculer la Transformée de Fourier Discrète (TFD ou DFT en anglais) sur les N' points obtenus.

Cet estimateur de la densité spectrale de puissance, du aux travaux de Blackman et Tuckey, porte le nom de corrélogramme.

$$P_{cor}(f) = \sum_{p=-(N'-1)}^{N'-1} R_{XX}(p) e^{-j2f \cdot p \cdot f} \tag{1.12}$$

Le périodogramme est un estimateur de la densité spectrale de puissance qui est d'autant meilleur que le signal est observé sur une longue plage de stationnarité. L'algorithme de la Transformée de Fourier Rapide est bien connu et la plupart des processeurs de signaux sont vendus avec des routines de TFR optimisées. Cet estimateur est donc aisé à utiliser et c'est la raison pour laquelle c'est l'estimateur le plus employé aujourd'hui.

1.2.2.2. Méthode du minimum de variance (méthode de Capon) :

Pour chaque fréquence f, cette méthode cherche un filtre adapté dont la réponse vaut 1 pour cette fréquence f et 0 partout ailleurs. Une fois ce filtre obtenu l'estimateur $P_{cap}(f)$ n'est autre que la puissance de sortie du filtre obtenu pour la fréquence f. La sortie $y(n)$ de ce filtre s'écrit :

$$y(n) = \sum_{i=0}^N h_i^* x(n-i) \tag{1.13}$$

Ou sous forme vectorielle :

$$y(n) = H^T \underline{X}(n) \tag{1.14}$$

Avec :

$$H = \begin{bmatrix} h_1 \\ h_2 \\ \bullet \\ \bullet \\ h_N \end{bmatrix} \quad \text{et} \quad X(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ \bullet \\ \bullet \\ x(n-N+1) \end{bmatrix} \quad \text{et} \quad F = \begin{bmatrix} 1 \\ e^{j2\pi f} \\ e^{j4\pi f} \\ \bullet \\ \bullet \\ e^{j2N\pi f} \end{bmatrix} \tag{1.15}$$

Le filtre H doit minimiser $E\{y(n)^2\}$ avec la contrainte :

$$H^T \bullet F = 1 \tag{1.16}$$

Ce qui s'écrit, en utilisant un multiplicateur de Lagrange α optimal est tel que :

$$E\left\{\left(H^T \underline{X}(n)\right)^2\right\} + \alpha(1 - H^T \bullet F) \quad \text{est minimal} \tag{1.17}$$

On annulant la dérivée par rapport a H :

$$H = \frac{\Gamma}{2} \underline{R}_{XX}^{-1} \bullet F \tag{1.18}$$

Avec $R_{XX} = E\{X(n) \underline{X}^T(n)\}$ matrice d'autocorrélation du signal $X(n)$.

On introduisant Γ dans l'expression de la contrainte :

$$\Gamma = \frac{2}{F^T R_{XX}^{-1} F} \quad (1.19)$$

D'où finalement :

$$H = \frac{R_{XX}^{-1} F}{F^T R_{XX}^{-1} F} \quad (1.20)$$

D'où :

$$P_{cap}(f) = E\{y(n)^2\} = \frac{1}{F^T R_{XX}^{-1} F} \quad (1.21)$$

En conclusion pour obtenir l'estimateur de Capon, il faut:

- Observer N échantillons du signal $X(n \cdot T_e)$
- Estimer la matrice d'autocorrélation et l'inverser
- Calculer $P_{cap}(f)$ pour chaque fréquence f .

Cette méthode souffre donc d'un coût de calcul supérieur aux estimateurs précédents. Il est possible de démontrer que cet estimateur à une variance minimale et c'est la raison pour laquelle il est souvent appelé estimateur du minimum de variance.

1.2.3. L'estimation spectrale par décomposition harmonique :

Entre ces deux méthodes il existe une troisième classe d'approches qui suppose que le signal est composé d'un certain nombre de raies spectrales dont il convient de trouver les fréquences et les puissances.

1.2.3.1. Méthode de Pisarenko:

Dans cette méthode, le signal $X(n)$ est supposé être constitué d'une somme de N sinusoides $S(n)$ et d'un bruit blanc additif $b(n)$.

$$X(n) = S(n) + b(n) \quad (1.22)$$

Sachant que toute sinusoïde pure $\sin(n\tilde{S})$ peut s'écrire:

$$\sin(n \cdot \tilde{S}) = 2 \cos(\tilde{\omega}) \sin(n-1) - \sin(n-2) \cdot \tilde{S}$$

D'où :

$$S(n) = -\sum_{m=1}^{2N} a_m S(n-m)$$

Alors :

$$X(n) = -\sum_{m=1}^{2N} a_m S(n-m) + b(n)$$

D'où :

$$\sum_{m=0}^{2N} a_m X(n-m) = \sum_{m=0}^{2N} a_m b(n-m) \quad (1.23)$$

Et :

$$\begin{aligned}
 X(n)^T A &= B(n)^T A \\
 R_{XX}(2N+1)A &= E\left\{ (S(n) + B(n)) B(n)^T \right\} A \quad (1.24)
 \end{aligned}$$

Or, le bruit est supposé blanc, de variance σ^2 et décorrélé du mélange de sinusoides, cette équation devient donc :

$$R_{XX}(2N+1)A = \sigma^2 A \quad (1.25)$$

On aboutie a la transformer en z :

$$\begin{aligned}
 S(n) &= -\sum_{m=1}^{2N} a_m S(n-m) \\
 S(Z) \left[1 - \sum_{m=1}^{2N} Z^{-m} \right] &= 0 \quad (1.26)
 \end{aligned}$$

Le vecteur A est donc le vecteur propre associé à la valeur propre σ^2 avec la contrainte $a_0 = 1$, les valeurs de Z pour lesquelles cette équation est vérifiée donnent les valeurs des fréquences présentes dans le mélange. Il faut donc extraire les racines du polynôme pour obtenir les N valeurs de fréquences, disposant des pulsations $\tilde{S}(i)$ et des coefficients d'autocorrélation $R_{XX}(i)$, on peut facilement déterminer $P(i)$ puissances respectives identifiées.

En conclusion pour analyser un signal selon la méthode de Pisarenko, il faut:

- Observer N valeurs du signal $X(n \cdot T_e)$
- Calculer la matrice d'autocorrélation et en faire la décomposition en éléments propres,
- Détecter σ^2 et en déduire le nombre de sinusoides, blocage à σ^2 des valeurs propres.
- Extraire les racines d'un polynôme complexe de degré 2N,
- Enfin si l'on veut les puissances, résoudre un système réel de N équations à N inconnues.

La décomposition en éléments propres reste pour l'instant l'étape la plus délicate à réaliser de manière rapide et c'est le frein principal à l'emploi de cette méthode.

1.2.4. L'estimation spectrale paramétrique :

A la différence de l'estimation spectrale non paramétrique, qui ne faisait aucune hypothèse sur le signal observé $X(n)$, si ce n'est des propriétés de stationnarité à l'ordre 2, l'estimation spectrale paramétrique suppose que ce signal suit un modèle donné. Ce modèle comporte un certain nombre de paramètres qui sont adaptables en fonction du signal observé à partir duquel il est aisé d'obtenir la densité spectrale de puissance. Il existe principalement trois grandes classes de modèles:

1.2.4.1. Le modèle auto régressif - modèle AR - :

Le signal $X(n)$ est supposé être prédictible en fonction d'un certain nombre de ses valeurs antérieures.

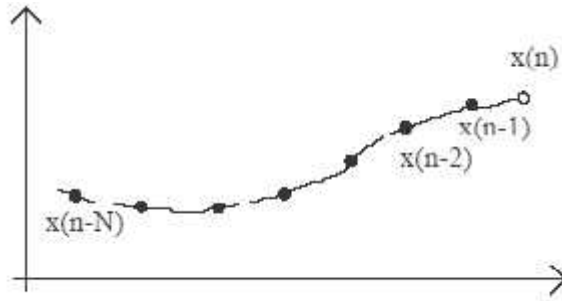


Figure.1.1- évolution du signal.

Il peut donc s'écrire:

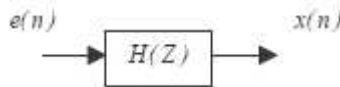
$$X(n) = \sum_{i=1}^N a_i X(n-i) + e(n) \tag{1.27}$$

Équation où les coefficients a_i $i = 1, \dots, N$ constituent les paramètres du modèle et où $e(n)$ est un bruit blanc décorrélé de $X(n)$ de variance σ^2 et qui représente l'erreur de prédiction.

La transformée en Z de cette équation donne alors:

$$X(Z) \left(1 - \sum_{i=1}^N a_i Z^{-i} \right) = E(Z) \tag{1.28}$$

Le signal $X(n)$ peut donc être vu comme le résultat du passage d'un bruit blanc $e(n)$ de variance σ^2 à travers un filtre de fonction de transfert $H(Z)$.



Avec :

$$H(Z) = \frac{1}{1 - \sum_{i=1}^N a_i Z^{-i}} \tag{1.29}$$

La connaissance des paramètres a_i permet donc de calculer la densité spectrale de puissance $P(f)$:

$$P_{ar}(f) = \frac{\sigma^2}{\left| 1 - \sum_{i=1}^N a_i e^{-j2\pi f i} \right|^2} \tag{1.30}$$

Il est possible d'optimiser un critère de minimisation d'erreur quadratique, c'est à dire, de chercher le jeu de paramètres a_i qui minimisent $E\{e(n)^2\}$ Ce qui peut s'écrire matriciellement de la manière suivante:

$$E\{e(n)^2\} = E\left\{ \left(x(n) - X_N^T(n-1) A_N \right)^2 \right\} \tag{1.31}$$

Après développement A_N optimal:

$$A_N = R_{XX}^{-1} r_N \quad (1.32)$$

Ou :

$$r_N = E\{X_N(n-1)X^*(n)\} \quad (1.33)$$

Il est donc possible de trouver le vecteur A_N en inversant la matrice R_{XX} puis de calculer l'énergie d'erreur de prédiction avant. Cependant, le coût de calcul d'une telle approche est alors en N^3 ce qui peut s'avérer gênant lorsque N est important. Il est possible de résoudre cette équation avec un coût de calcul proportionnel à N^2 en utilisant l'algorithme de Levinson.

En conclusion, la modélisation AR demande la mise à jour de N coefficients ai pour cela il faut:

- Observer au minimum N valeurs du signal
- Calculer les coefficients d'autocorrélation,
- Appliquer l'algorithme de Levinson (coût de calcul proportionnel à $O(N^2)$).

1.2.4.2. Le modèle à moyenne ajustée - modèle MA - :

Le signal $X(n)$ est supposé pouvoir s'écrire comme une combinaison linéaire d'échantillons décorrélés entre eux, ce qui peut se formaliser comme une combinaison linéaire d'échantillons d'un bruit blanc $e(n)$.

$$X(n) = \sum_{i=0}^M b_i e(n-i) \quad (1.34)$$

Le signal $X(n)$ peut donc être vu comme le résultat du passage d'un bruit blanc $e(n)$ à travers un filtre de fonction de transfert $H(Z)$, avec :

$$H(Z) = \sum_{i=0}^M b_i Z^{-i} \quad (1.35)$$

La densité spectrale de puissance du signal $X(n)$ s'écrit alors :

$$P_{ma}(f) = \left| \sum_{i=0}^M b_i e^{-j2\pi f i} \right|^2 \sigma^2 \quad (1.36)$$

Dans le cas du modèle AR, le vecteur A_N s'obtient en optimisant un critère d'erreur quadratique. Cette optimisation conduit de manière explicite à la résolution d'un système linéaire, Par contre, pour un modèle MA, l'erreur de prédiction s'écrit:

$$e(n) = x(n) - E_N^T(n-1) B_N \quad (1.37)$$

avec : $E_N^T(n-1) = [e(n-1), e(n-2), \dots, e(n-M)]$

Le vecteur E_N dépend de B_N et le système à résoudre devient non linéaire. L'optimisation exacte du critère est alors très délicate. Cependant, l'algorithme de Durbin permet d'approcher la solution optimale avec de bons résultats.

Conclusion, l'obtention des M valeurs b_i se fait en résolvant deux systèmes AR, il faut:

- Modéliser le signal $X(n)$ sous la forme d'un processus AR d'ordre $N \gg M$ et trouver le vecteur des paramètres, grâce à l'algorithme de Levinson par exemple,

- Modéliser le signal constitué par les paramètres précédents sous une forme AR d'ordre M et trouver le nouveau vecteur des paramètres.

1.2.4.3. Le modèle auto régressif à moyenne ajustée - modèle ARMA- :

ARMA Combinaison des modèles AR et MA où le signal $X(n)$ est supposé pouvoir s'écrire en fonction de N valeurs passées et de M échantillons d'un bruit blanc décorrélé.

$$X(n) = \sum_{i=1}^N a_i X(n-i) + \sum_{i=0}^M b_i e(n-i) \quad (1.38)$$

D'où :

$$P_{arma} = \frac{\left| \sum_{i=0}^M b_i e^{-j2\pi f_i} \right|^2}{\left| 1 - \sum_{i=1}^N a_i e^{-j2\pi f_i} \right|^2} \sigma^2 \quad (1.39)$$

$$\begin{aligned} H(Z)_{arma} &= H(z)_{ar} H(Z)_{ma} \\ &= \frac{\sum_{i=0}^M b_i Z^{-i}}{1 - \sum_{i=1}^N a_i Z^{-i}} \end{aligned} \quad (1.40)$$

1.3. Définition de quelques estimateurs statistiques :

La **valeur moyenne** d'une variable $X(n)$ est définie comme suit :

$$\bar{X} = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=0}^{N-1} X(n) \quad (1.41)$$

On peut également calculer la puissance moyenne de la variable :

$$\bar{X}^2 = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=0}^{N-1} (X(n))^2 \quad (1.42)$$

On notera que la valeur moyenne \bar{X} représente la composante continue du signal autour de laquelle prennent place les fluctuations. La puissance de ces fluctuations est mesurée par la variance de la variable $X(n)$:

$$\begin{aligned} \sigma_X^2 &= \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=0}^{N-1} (X(n) - \bar{X})^2 \\ \sigma_X^2 &= \bar{X}^2 - 2\bar{\mu}_X \bar{\mu}_X + \mu_X^2 = \mu_{X^2} - 2\bar{\mu}_X^2 + \bar{X}^2 \end{aligned}$$

On en déduit donc que la variance vaut :

$$\sigma_x^2 = \sigma_{x^2} - \sigma_x^2 \quad (1.43)$$

Il est intéressant d'interpréter cette équation en disant que la puissance des variations (σ_{x^2}) est égale à la puissance totale (σ_{x^2}) moins la puissance continue (σ_x^2).

On définit également l'écart type comme la racine carrée de la variance :

$$\sigma_x = \sqrt{\sigma_{x^2} - \sigma_x^2} \quad (1.44)$$

Sa valeur est égale à la valeur efficace des variations du signal autour de la valeur moyenne.

Dans le cas où le nombre de valeurs N de la variable $X(n)$ n'est pas infiniment grand, on parle de son espérance mathématique $E\{f(X(n))\}$ plutôt que de valeur moyenne $\bar{f(X(n))}$.

Ce qui donne :

$$E\{X(n)\} = \frac{1}{N} \sum_{n=0}^{N-1} X(n) \approx \bar{X} \quad (1.45)$$

$$E\{X(n)^2\} = \frac{1}{N} \sum_{n=0}^{N-1} X(n)^2 \approx \sigma_{x^2} \quad (1.46)$$

Il est intéressant de relever que si l'on considère une notation vectorielle, la puissance s'écrit simplement sous la forme d'un produit scalaire :

$$P_x = \frac{1}{N} \sum_{n=0}^{N-1} X(n)^2 = \frac{1}{N} \underline{X} \underline{X}^T \quad (1.47)$$

De même, l'indépendance (ou la correspondance) de deux signaux ou vecteurs peut se mesurer avec le produit scalaire :

$$a = \underline{X}^T \underline{Y} = |\underline{X}| |\underline{Y}| \cos(\alpha) \quad (1.48)$$

1.4. Régression linéaire :

La régression linéaire consiste en la recherche de la droite passant au mieux parmi un ensemble de points mesurés (**figure.1. 2**). Le critère conduisant à cet optimum est la minimisation des distances quadratiques entre les points mesurés et la droite optimum.

On notera que la régression linéaire s'applique aux systèmes statiques alors que l'approche de Wiener (que l'on verra dans la section suivante) sert à optimiser des systèmes évoluant au cours du temps.

1.4.1. Mesure, modèle et écart :

Comme on souhaite faire passer une droite parmi un ensemble de points, on se donne un modèle dont l'équation est :

$$y_m = ax + b \quad (1.49)$$

L'écart de $y(n)$ par rapport au modèle s'écrit donc :

$$e(n) = y(n) - y_m(n) \quad e(n) = y(n) - (ax(n) + b)$$

Si l'on décrit la mesure $y(n)$ par rapport au modèle $y_m(n)$, on a évidemment : $y(n) = y_m(n) + e(n)$

On associe généralement deux grandeurs à l'écart $e(n)$:

- sa valeur moyenne μ_e qui doit tendre vers 0 si le modèle n'est pas biaisé.
- sa puissance σ_e^2 .

Qui doit diminuer avec l'amélioration du modèle. On notera que pour le calcul d'une régression linéaire, on fait l'hypothèse qu'il n'y a pas de bruit sur la valeur de la variable indépendante $x(n)$.

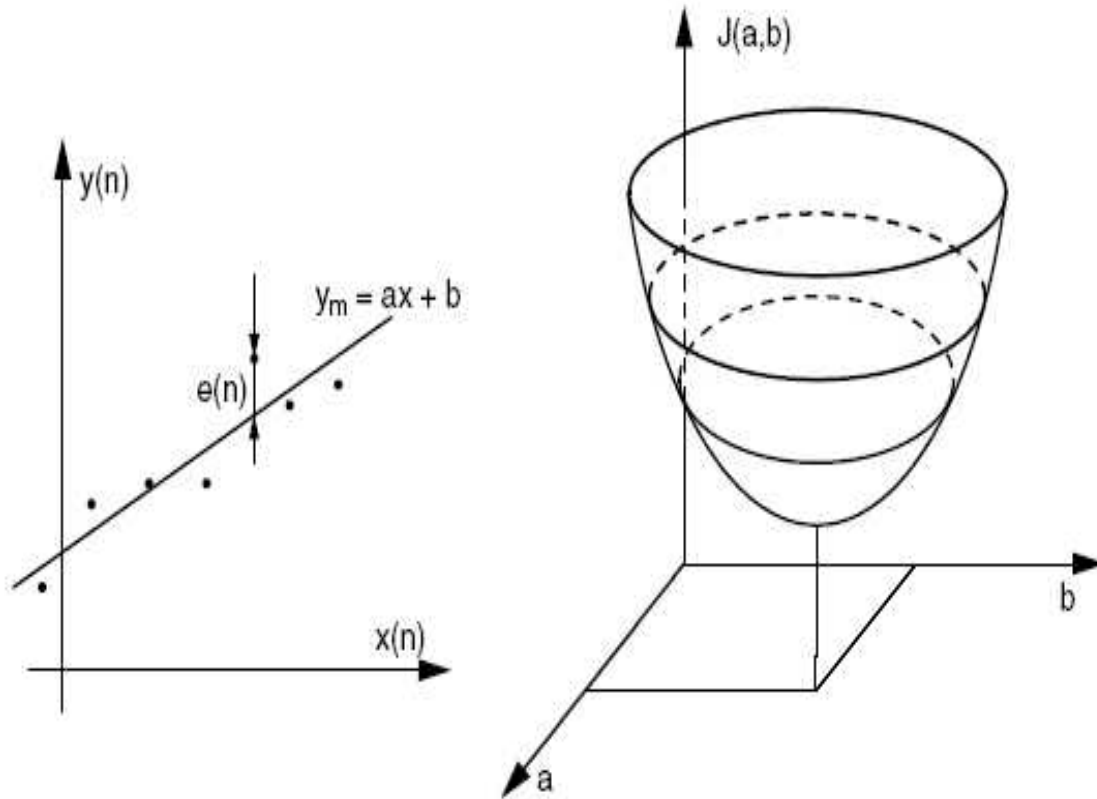


Figure.1.2- Régression linéaire.

1.4.2. Minimisation de l'écart quadratique :

L'obtention de la droite passant au mieux parmi les points mesurés nécessite la recherche du minimum d'une fonction dépendant des paramètres recherchés a et b . Pour cela, on définit un critère d'optimisation qui mesure la puissance ou la variance de l'écart :

$$J(a,b) = \frac{1}{N} \sum_{n=0}^{N-1} e^2(n) = \frac{1}{N} \sum_{n=0}^{N-1} (y(n) - (ax(n) + b))^2$$

Lorsque l'écart quadratique est minimum, on a :

$$\frac{\partial J(a,b)}{\partial a} = 0 \qquad \frac{\partial J(a,b)}{\partial b} = 0$$

Avec :

$$\frac{\partial J(a,b)}{\partial a} = \frac{2}{N} \left(-\sum_{n=0}^{N-1} (y(n)x(n)) + a \sum_{n=0}^{N-1} x(n)^2 + \sum_{n=0}^{N-1} bx(n) \right)$$

$$\frac{\partial J(a,b)}{\partial b} = \frac{2}{N} \left(-\sum_{n=0}^{N-1} y(n) + a \sum_{n=0}^{N-1} x(n) + \sum_{n=0}^{N-1} b \right)$$

On en tire 2 équations dont les inconnues sont a et b :

$$\frac{1}{N} a \sum_{n=0}^{N-1} x(n)^2 + \frac{1}{N} \sum_{n=0}^{N-1} bx(n) = \frac{1}{N} \sum_{n=0}^{N-1} y(n)x(n) \tag{1.50}$$

$$\frac{1}{N} a \sum_{n=0}^{N-1} x(n) + \frac{1}{N} \sum_{n=0}^{N-1} b = \frac{1}{N} \sum_{n=0}^{N-1} y(n) \tag{1.51}$$

1.4.3. Équations de la régression linéaire :

Se souvenant de la définition d'une valeur moyenne, on voit que les équations (1.50) et (1.51) s'écrivent plus simplement sous la forme :

$$a \tilde{x} + b = \tilde{y}$$

$$a \tilde{x}^2 + b \tilde{x} = \tilde{xy}$$

Sous forme matricielle, cela donne :

$$\begin{pmatrix} \tilde{x} & 1 \\ \tilde{x}^2 & \tilde{x} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \tilde{y} \\ \tilde{xy} \end{pmatrix} \text{ Dont la solution est : } \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \tilde{x} & 1 \\ \tilde{x}^2 & \tilde{x} \end{pmatrix}^{-1} \begin{pmatrix} \tilde{y} \\ \tilde{xy} \end{pmatrix}$$

L'inversion de la matrice et le calcul explicite de a et b donnent alors :

$$a = \frac{\tilde{x} \tilde{xy} - \tilde{y} \tilde{x}^2}{\tilde{x}^2 - \tilde{x}^2} \text{ et } b = \frac{\tilde{x} \tilde{xy} - \tilde{y} \tilde{x}^2}{\tilde{x}^2 - \tilde{x}^2}$$

Dans le cas particulier où la droite passe par l'origine, les valeurs moyennes μ_x et μ_y sont nulles et on a :

$$a = \frac{\tilde{xy}}{\tilde{x}^2} = \frac{\sum x(n)y(n)}{\sum x^2(n)} = \frac{X^T Y}{X^T X} \quad b=0$$

1.5. Conclusion :

Il existe un grand nombre de méthodes d'analyse spectrale et le choix est souvent dicté par l'application considérée. En général, les méthodes non paramétriques trouvent leur intérêt en présence de signaux longs et stationnaires tandis que les méthodes paramétriques sont plutôt utilisées pour les signaux brefs et non stationnaires. En fin, les méthodes de modélisation AR peuvent être rendues adaptatives, ce qui permet une analyse temps réel de la densité spectrale de puissance du signal observé.

Dans de nombreuses applications, les signaux temporels sont entachées d'une interférence ou d'un bruit non désirés. Il faut alors trouver une solution permettant de supprimer ou tout au moins réduire ces composantes perturbatrices. Dans le cas où le spectre du signal désiré et celui du signal perturbateur se superposent, il n'est pas possible de recourir au filtrage classique.

Le filtre de Wiener apporte une solution à ce problème lorsque le processus est stationnaire.

Chapitre N°02

Filtrage adaptatif

2.1. Introduction :

2.1.1. Objectifs et déroulement du travail :

De façon générale les filtres adaptatifs sont des systèmes appliqués sur des données bruitées pour obtenir une information utile à un certain instant (t), ces systèmes étant mis en oeuvre dans trois configurations :

Le filtrage c'est à dire l'extraction de cette information au temps t à partir des données bruitées mesurées jusqu'au temps t inclus.

Le lissage qui utilisera aussi les données postérieures au temps t .

La prédiction qui ne se sert que des données jusqu'au temps ($t - 1$) pour déduire l'information qui nous intéresse au temps t .

Dans la première partie de ce chapitre nous exposerons l'approche statistique du problème (filtrage de Wiener) qui suppose la disponibilité de certaines grandeurs statistiques (moyenne et autocorrélation) du signal utile et du bruit. L'approche consiste alors à minimiser la moyenne statistique du carré de l'erreur (EQM ou MSE en anglais) entre l'information désirée et la sortie du filtre.

Ce filtrage de Wiener [1] est inadéquat pour les situations dans lesquelles le signal ou le bruit sont non stationnaires [2]. Dans de telles situations le filtre optimal doit être variable dans le temps. La solution à ce problème est fournie par le filtrage de Kalman.

Le filtrage adaptatif pourra aussi être utilisé dans ce cas. Par rapport au filtrage classique le filtrage adaptatif comporte une mise à jour récursive des paramètres (coefficients) du filtre. L'algorithme part de conditions initiales prédéterminées et modifie de façon récursive les coefficients du filtre pour s'adapter au processus. Si celui-ci est stationnaire l'algorithme doit converger vers la solution optimale de Wiener, sinon il présentera une capacité à suivre des variations des grandeurs statistiques du processus si celles-ci sont suffisamment lentes.

Pour présenter ces techniques adaptatives nous présenterons tout d'abord l'algorithme du gradient qui fournit un algorithme récursif de calcul des coefficients du filtre [3]. Nous donnerons ensuite une version dans laquelle les grandeurs statistiques impliquées sont remplacées par des valeurs instantanées, on obtient alors l'algorithme très fréquemment utilisé du gradient stochastique (LMS). Nous reformulerons le problème en termes de moindres carrés et présenterons un des algorithmes récursifs basé sur cette approche : l'algorithme des moindres carrés récursifs (RLS) [4].

2.1.2. Classification des filtres:

Les filtres adaptatifs peuvent être classés en fonction des choix qui sont faits sur les points suivants:

- Le critère d'optimisation,
- L'algorithme de mise à jour des coefficients,
- La structure du filtre programmable,
- Le type de signal traité, mono ou multidimensionnel.

2.1.3. Choix de l'algorithme :

Le choix de l'algorithme se fera en fonction des critères suivants :

- La rapidité de convergence qui sera le nombre d'itérations nécessaires pour converger « assez près » de la solution optimale de Wiener dans le cas stationnaire.
- La mesure de cette « proximité » entre cette solution optimale et la solution obtenue.
- La capacité de poursuite (tracking) des variations (non stationnarités) du processus. On examinera quels sont les algorithmes vraiment adaptatifs.
- La robustesse au bruit
- La complexité (en nombre de MIPS)
- La structure (se prêtant plus ou moins à l'implémentation en VLSI)
- Les propriétés numériques (stabilité – précision) dans le cas d'une précision limitée sur les données et les coefficients (problèmes d'implémentation en virgule fixe).

Nous ne nous intéresserons dans le cadre de cette thèse qu'aux trois premiers critères de choix.

Remarques :

1. La théorie des filtres incluant celle des filtres de Wiener et Kalman a été développée pour des signaux à temps continu et à temps discret mais nous considérerons par la suite uniquement le cas discret où les signaux sont échantillonnés et les filtres sont numériques. Nous nous limiterons de plus aux filtres à réponse impulsionnelle finie (FIR)

2. Le début de ce chapitre est écrit pour des signaux complexes. Les formules obtenues étant un peu lourdes, pour cela la suite du chapitre est écrite pour des signaux réels. Vous pouvez donc, lors de la lecture, ignorer les conjuguaisons $()^*$ et translater les conjuguées hermitiennes $()^+$ ou $()^H$ en transposées simples $()^T$.

2.2. Filtrage linéaire optimal Filtrage de Weiner :

Considérons la figure suivante :

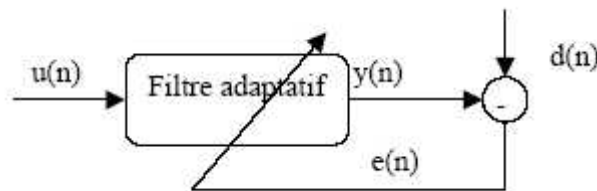


Figure.2.1- filtrage optimal.

Le problème du filtrage optimal est de trouver le « meilleur » filtre c'est à dire celui permettant d'obtenir en sortie une réponse $y(n)$ la plus « proche » possible d'une réponse désirée $d(n)$ lorsque l'entrée est une certaine séquence $u(n)$, On note :

$$e(n) = y(n) - d(n) \quad (2.1)$$

L'erreur entre la réponse désirée $d(n)$ et la sortie $y(n)$. On note également $w(n)$ la réponse pulsionnelle du filtre. On noter :

$$y(n) = \hat{d}(n / \underline{X}_n)$$

Où \underline{X}_n est un vecteur d'état contenant toute l'information utilisée pour prédire $d(n)$:

$$\underline{X}_n = [u(0), \dots, u(n), e(0), \dots, e(n-1), \dots, w(0), \dots, w(M-1), y(0), \dots, y(n-1)] \quad (2.2)$$

Le problème consiste donc à rechercher le filtre assurant l'erreur la plus faible $e(n)$, au sens d'une certaine fonction de coût :

$$w = \arg \min J(e(n)) \quad (2.3)$$

De nombreux choix sont possibles en ce qui concerne la fonction de coût, l'erreur quadratique moyenne est la plus utilisée, car elle conduit à des développements mathématiques complets et simples, fournit la solution en fonction des caractéristiques au second ordre des variables aléatoires, caractéristiques qui sont les plus simples à estimer, et enfin fournit une solution unique. C'est sur l'estimation linéaire en moyenne quadratique que repose le filtrage de Wiener.

2.2.1. Relations d'orthogonalité - Equation de Wiener-Hopf :

La sortie du filtre s'écrit :

$$y(n) = \sum_{k=0}^{M-1} w^* u(n-k) \quad (2.4)$$

Et l'erreur : $e(n) = y(n) - d(n)$

Le filtre de Wiener est celui qui minimise l'erreur quadratique moyenne (**EQM** ou **MSE** anglais) :

$$J = E\{(e(n))^2\} \quad (2.5)$$

En introduisant les vecteurs :

$$\underline{w}^T = [w_0, \dots, w_{M-1}] \text{ et } \underline{u}^T(n) = [u(n), u(n-1), \dots, u(n-M+1)]$$

On a alors :

$$e(n) = d(n) - \underline{w}^H \underline{u}(n) = d(n) - \underline{u}^T(n) \underline{w}^* \quad (2.6)$$

D'où :

$$\begin{aligned} J &= E\{(d(n) - \underline{w}^H \underline{u}(n))(d^*(n) - \underline{w}^T \underline{u}^*(n))\} \\ &= E\{d(n)^2\} - \underline{w}^H E\{\underline{u}(n) d^*(n)\} - \underline{w}^T E\{\underline{u}^*(n) d(n)\} + \underline{w}^H E\{\underline{u}(n) \underline{u}^*(n)\} \underline{w} \end{aligned}$$

Soit :

$$J = \dagger_d^2 - \underline{w}^H \underline{R}_{ud} - \underline{w}^T \underline{R}_{ud}^* + \underline{w}^H \underline{R}_{uu} \underline{w} \quad (2.7)$$

Avec \underline{R}_{uu} qui est la matrice d'autocorrélation de l'entrée $u(n)$. Cette matrice est définie positive, de Toeplitz et à symétrie hermitienne : $\underline{R}_{uu} = \underline{R}_{uu}^H$, on aboutie au même résultats par le théorème projection (**ANNEXE 2**).

Et $\underline{R}_{ud} = E\{\underline{u}(n) d^*(n)\}$ le vecteur d'intercorrélation entre la sortie désirée $d(n)$ et l'entrée $u(n)$.

Le vecteur optimum \underline{w} est celui qui annule le gradient du critère :

$$\frac{dJ}{d\underline{w}} = \underline{0} \tag{2.8}$$

En écrivant J sous la forme $J = E \{ e_n \underline{e}_n^* \}$ on a :

$$\frac{dJ}{d\underline{w}} = E \left\{ e_n \frac{de_n^*}{d\underline{w}} \right\} + E \left\{ e_n^* \frac{de_n}{d\underline{w}} \right\} \tag{2.9}$$

Or le gradient vectoriel complexe (ANNEXE 3). nous donne :

$$\frac{de_n^*}{d\underline{w}} = \underline{0} \quad \text{et} \quad \frac{de_n}{d\underline{w}} = 2\underline{u}(n)$$

Par conséquent, à l'optimum, on a :

$$E \{ e_n^* \underline{u}(n) \} = \underline{0} \tag{2.10}$$

C'est le principe d'orthogonalité signifiant que toutes les entrées $u(n)$ sont décorrélées de e_n^* .

En développant cette dernière équation on obtient :

$$E \left\{ \underline{u}(n) \left(d^*(n) - \underline{u}^H(n) \underline{w} \right) \right\}$$

Alors on a :

$$\underline{R}_{uu} \underline{w} = \underline{R}_{ud} \tag{2.11}$$

Cette relation est appelée Formule de Wiener ou équation de Wiener Hopf. La solution est le filtre optimal :

$$\underline{w} = \left(\underline{R}_{uu} \right)^{-1} \underline{R}_{ud} \tag{2.12}$$

Pour ce vecteur optimal on obtient à partir de (2.7) et (2.11) l'Erreur Quadratique Moyenne minimale (Annexe 3):

$$J_{\min} = \hat{d}^2 - \underline{w}^H \underline{R}_{ud} \underline{w} = \hat{d}^2 - \hat{d}^2 \tag{2.13}$$

On note :

$$\hat{d} = \underline{w}^H \underline{u}(n) \text{ Et } \hat{d}^2$$

Signal filtré optimal et la variance de ce signal.

Cette relation montre que pour le filtre optimal, l'EQM est la différence entre la variance du signal désiré et celle de l'estimée de ce signal produite par le filtre.

2.2.2. Application :

Le filtrage de Wiener adaptatif est un outil puissant en traitement du signal, communications numériques et contrôle automatique. Les applications sont diverses mais présentent toujours les caractéristiques exposées plus haut : on dispose d'une entrée u ainsi que de la réponse désirée d et l'erreur e entre la sortie y et d sert à contrôler (adapter) les valeurs des coefficients du filtre w . Ce qui différencie essentiellement les applications provient de la façon de définir la réponse désirée d .

On peut distinguer quatre grandes classes d'applications :

- L'identification de systèmes
- La prédiction
- La modélisation inverse
- L'annulation d'interférences

Nous donnons ci après les schémas correspondant à ces quatre classes :

2.2.2.1. Identification de Systèmes :

$d(n)$ Est la sortie du système que l'on souhaite identifier

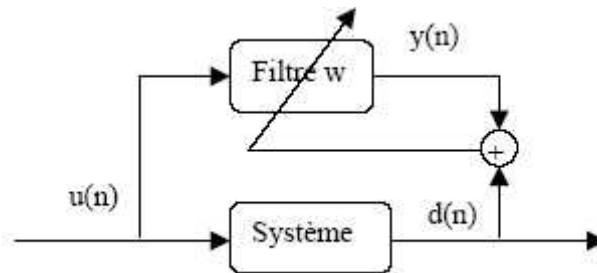


Figure.2.2-Identification de Systèmes.

2.2.2.2. Prédiction :

$d(n)$ Est le signal à l'instant n et $y(n)$ le signal prédit à partir du signal aux instants précédents.

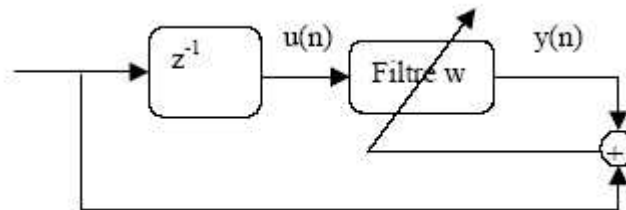


Figure.2.3-Prédiction.

2.2.2.3. Modélisation inverse (égalisation, déconvolution) :

$d(n)$ Est l'entrée (retardée) du système que l'on cherche à « inverser ».

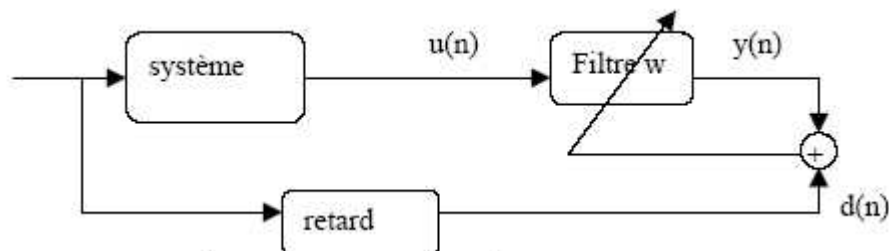


Figure.2.4-Modélisation inverse.

2.2.2.4. Annulation d'interférences (annulation d'écho, de bruit) :

$d(n)$ Est un signal primaire qui comporte les interférences à annuler. $u(n)$ Est le signal de référence dénué (ou presque) d'information et obtenu par un capteur proche de celui qui fournit $d(n)$.

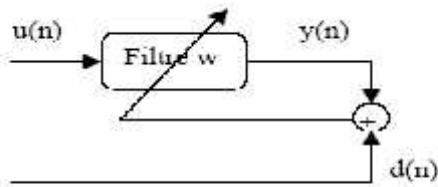


Figure.2.5-Annulation d'interférences.

2.3. Algorithme du gradient :

L'équation de Wiener Hopf (2.11) qui permet de calculer le filtre de Wiener optimal conduit à résoudre un système de M équations à M inconnues : $\underline{R}_{uu} \underline{w} = \underline{R}_{ud}$

Une méthode d'inversion directe est en $O(M^3)$, ce qui est très coûteux. Il peut être préférable de résoudre ce système par une méthode itérative, notamment en se souvenant que la fonction de coût est quadratique, ce qui entraîne que le minimum est unique.

La méthode du gradient consiste à :

- Choisir un vecteur initial de coefficients $\underline{w}(0)$.
- Ayant un vecteur candidat $\underline{w}(n)$ à l'étape n, obtenir un meilleur candidat en incrémentant $\underline{w}(n)$ dans la direction opposée au gradient du coût J.

$$\underline{w}(n+1) = \underline{w}(n) - \frac{1}{2} \tilde{\mu}_n \underline{\nabla} J \Big|_{\underline{w}=\underline{w}(n)} \tag{2.14}$$

La séquence $\{\tilde{\mu}_n\}$ est une série de coefficients appelés pas d'adaptation.

On distingue l'algorithme à pas d'adaptation constant $\tilde{\mu}_n = \tilde{\mu}$ et les algorithmes à pas décroissant.

Le gradient est égal à :

$$\underline{\nabla} J \Big|_{\underline{w}=\underline{w}(n)} = 2 \underline{R}_{uu} \underline{w} - 2 \underline{R}_{ud} \tag{2.15}$$

La forme explicite de l'algorithme est donc :

$$\underline{w}(n+1) = \underline{w}(n) - \tilde{\mu}_n \left(\underline{R}_{uu} \underline{w} - \underline{R}_{ud} \right) \tag{2.16}$$

On note :

$$v(n) = d(n) - \hat{d}(n)$$

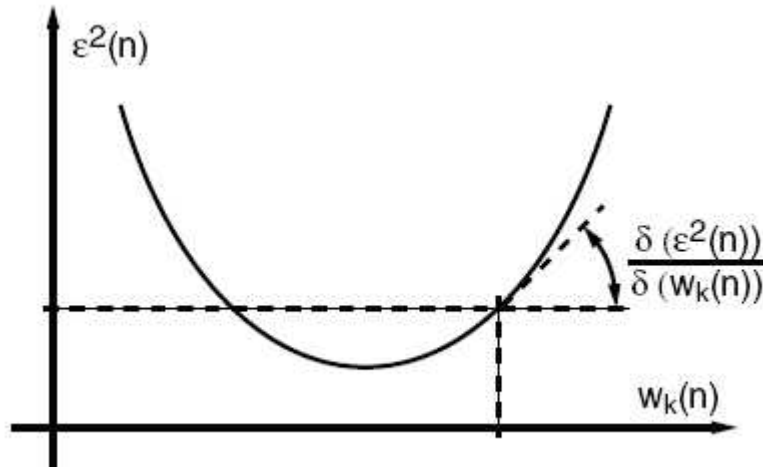


Figure 2.6-Derivee de l'erreur quadratique.

Notons que cette forme est déterministe et nécessite la connaissance des grandeurs statistiques et Le paramètre μ_n contrôle l'importance de la correction apportée au vecteur de coefficients \underline{w} lors du n itération. Le paragraphe suivant porte sur l'étude de la convergence de l'algorithme c'est à dire sur son aptitude à être stable et à tendre vers la solution optimale. Les deux facteurs influençant à priori cette convergence sont le pas d'adaptation μ_n et la matrice d'autocorrélation.

2.3.1. Convergence de l'algorithme du gradient :

Nous commençons en étudiant le comportement du vecteur d'erreur entre le vecteur optimal \underline{w} et le vecteur déterminé \underline{w} à la n itération de l'algorithme.

On note :

$$\underline{v}(n) = \underline{w}(n) - \underline{w}$$

A partir des équations (2.13) et (2.16), on obtient :

$$\underline{w}(n+1) = \underline{w}(n) - \mu_n \left(\underline{R}_{\underline{w}} \left[\underline{w} - \underline{w}(n) \right] \right) \tag{2.17}$$

D'où :

$$\underline{v}(n+1) = \left(I - \mu_n \underline{R}_{\underline{w}} \right) \underline{v}(n) \tag{2.18}$$

Introduisons maintenant la décomposition propre de la matrice d'autocorrélation :

$$\underline{R}_{\underline{w}} = \underline{Q} \underline{P} \underline{Q}^H,$$

Où \underline{Q} est unitaire.

Soit en multipliant les membres de l'équation (2.18) par \underline{Q} et en notant $\underline{r}(n) = \underline{Q}^H \underline{v}(n)$:

$$\underline{r}(n+1) = \left(I - \mu_n \underline{P} \right) \underline{r}(n) \tag{2.19}$$

De cette façon, on fait apparaître les modes propres de l'algorithme.

L'algorithme converge si $\underline{r}(n) \rightarrow 0$, ce qui entraîne bien que $\underline{v}(n) \rightarrow 0$, soit $\underline{w}(n) \rightarrow \underline{w}$.

Le problème est donc de donner les conditions qui assurent cette convergence, de quantifier la rapidité de convergence, et de choisir le pas d'adaptation.

2.3.1.1. Conditions de convergence pour un pas μ constant :

L'équation (2.19) nous donne par récurrence :

$$\underline{r}(n) = (I - \mu \underline{P})^n \underline{r}(0) \tag{2.20}$$

Chacune des composantes s'exprime alors sous la forme :

$$r_k(n) = (I - \mu \lambda_k)^n r_k(0) \tag{2.21}$$

Où λ_k est la k^{ième} valeur propre.

Les différentes composantes convergent à 0 pourvu que nous en déduisons alors la condition de convergence de l'algorithme du gradient :

$$|1 - \mu \lambda_k| < 1, \forall k \in \{0, 1, \dots, M-1\} \tag{2.22}$$

On notant λ_{\max} la valeur propre maximale :

$$0 < \mu < \frac{2}{\lambda_{\max}} \tag{2.23}$$

2.3.1.2. Rapidité de convergence :

Nous nous intéressons maintenant à la vitesse de convergence et choisissons de suivre l'évolution du critère J à minimiser (nous aurions pu continuer de nous intéresser à la convergence des coefficients vers l'optimum):

$$\underline{R} \underline{w} = \underline{R}_{ud}$$

D'après l'équation (2.7) et avec, on peut écrire :

$$\begin{aligned} J &= \frac{1}{d} \|\underline{w} - \underline{R}^{-1} \underline{R} \underline{w} - \underline{R}^{-1} \underline{R}_{ud} + \underline{R}^{-1} \underline{R}_{ud}\|_d^2 \\ &= \frac{1}{d} \|\underline{w} - \underline{R}^{-1} \underline{R} \underline{w} + (\underline{w} - \underline{w})\|_d^2 \end{aligned}$$

D'après l'équation (1.12), on a donc :

$$J = J_{\min} + (\underline{w} - \underline{w})^H \underline{R}^{-1} (\underline{w} - \underline{w}) \tag{2.24}$$

En utilisant la décomposition propre de la matrice d'autocorrélation et en utilisant de nouveau :

$$\underline{r}(n) = \underline{Q} (\underline{w}(n) - \underline{w}(n))$$

On obtient :

$$J = J_{\min} + \underline{r}^H \underline{P} \underline{r}$$

En utilisant maintenant :

$$\underline{r}(n) = (I - \mu \underline{P})^n \underline{r}(0)$$

Nous pouvons écrire :

$$J = J_{\min} + r^H(0) \left(I - \mu P \right)^n \underline{P} \left(I - \mu P \right)^n \underline{P}^T r(0)$$

Toutes les matrices étant diagonales.

$$J(n) = J_{\min} + \sum_{k=0}^{M-1} \lambda_k (1 - \mu \lambda_k)^{2n} |r_k(0)|^2 \tag{2.25}$$

Lorsque l'algorithme est convergent c'est à dire lorsque : $0 < \mu < \frac{2}{\lambda_{\max}}$ quelque soient les conditions initiales $w(0)$.

La courbe obtenue en traçant $J(n)$ en fonction du nombre d'itérations n est appelée courbe d'apprentissage. D'après l'équation (2.25) cette courbe consiste en une somme d'exponentielles décroissantes, chacune d'elle correspondant à un mode propre de l'algorithme. La vitesse de convergence du mode k est liée à $(1 - \mu \lambda_k)^{2n}$.

Le mode le plus lent est lié à la valeur propre la plus petite et le mode le plus rapide est lié à la valeur propre la plus grande.

A une valeur propre, on associe une constante de temps, telle que :

$$|1 - \mu \lambda_k|^{2\tau_k} = \exp\left(\frac{-1}{\tau_k}\right)$$

La constante de temps de l'erreur quadratique est alors

$$\frac{-1}{2 \log(1 - \mu \lambda_{\max})} \leq \tau \leq \frac{-1}{2 \log(1 - \mu \lambda_{\min})} \tag{2.26}$$

Ceci montre que la convergence est d'autant plus lente que le pas est faible.

Remarque : pour $\mu \ll 1$ on peut approcher la constante de temps par :

$$\tau_k \approx \frac{1}{2 \mu \lambda_k}$$

2.3.1.3. Pas optimal :

La rapidité de convergence est gouvernée, pour chacun des modes, par $u_k = |1 - \mu \lambda_k|$

La solution optimale consiste à minimiser le plus grand des u_k : il s'agit donc d'un problème de type minimax :

$$\mu_{opt} = \arg \min_{\mu} \max_k (|1 - \mu \lambda_k|) \tag{2.27}$$

Pour : $\mu > \mu_{opt}$ $1 - \mu \lambda_k < 1 - \mu_{opt} \lambda_k$ $\max (|1 - \mu \lambda_k|) = \mu \lambda_{\max} - 1$

Pour : $\mu < \mu_{opt}$ $1 - \mu \lambda_k > 1 - \mu_{opt} \lambda_k$ $\max (|1 - \mu \lambda_k|) = 1 - \mu \lambda_{\min}$

Donc pour $\mu = \mu_{opt}$ on a $1 - \mu \lambda_k = 1 - \mu_{opt} \lambda_k$

On a par conséquent :

$$\mu_{opt} = \frac{2}{\lambda_{\min} + \lambda_{\max}} \tag{2.28}$$

On peut noter que ce pas appartient bien au domaine des pas qui assurent la convergence, c'est-à-dire : $\tilde{\mu} \in \left[0, \frac{2}{\lambda_{\max}} \right]$

Cependant, calculer les valeurs propres de la matrice de corrélation est aussi compliqué que d'inverser la matrice.

Le choix du pas fait donc perdre l'avantage de prendre un algorithme « économique ».

Une mesure de compromis consiste à prendre :

$$\tilde{\mu} = \frac{2}{\sum_{k=1}^M \lambda_k} = \frac{2}{\text{trace}(\underline{\underline{R}}_{uu})} \tag{2.29}$$

La trace de $\underline{\underline{R}}_{uu}$ est simple à calculer (somme des éléments sur la diagonale principale). La matrice étant de Toeplitz, on peut aussi voir que :

$$\text{trace} \underline{\underline{R}}_{uu} = M r_{uu}(0)$$

Et $r_{uu}(0)$ est la puissance de $u(n)$:

$$r_{uu}(0) = E \left\{ |u(n)|^2 \right\}$$

On constate facilement que $\tilde{\mu} < \frac{2}{\lambda_{\max}}$ et, avec un peu de chance, n'est pas très loin de $\tilde{\mu}_{opt}$.

2.3.1.4. Pas optimal et conditionnement de la matrice d'autocorrélation :

D'après :

$$\underline{\underline{r}}(n) = (\underline{\underline{I}} - \tilde{\mu} \underline{\underline{P}})^n \underline{\underline{r}}(0)$$

La convergence de $\underline{\underline{r}}(n)$ est gouvernée par la valeur propre la plus grande de :

$$\underline{\underline{A}} = \underline{\underline{I}} - \tilde{\mu} \underline{\underline{P}}$$

On peut remarquer que pour n assez grand on a :

$$\underline{\underline{r}}(n) \approx \zeta (\underline{\underline{A}})^n \underline{\underline{r}}(0)$$

Où ζ est le rayon spectral (la plus grande valeur propre) de $\underline{\underline{A}}$. Pour $\tilde{\mu} = \tilde{\mu}_{opt}$ on a :

$$\zeta(\underline{\underline{A}}) = 1 - \tilde{\mu} \lambda_{\min} = \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} = \frac{C(\underline{\underline{R}}) - 1}{C(\underline{\underline{R}}) + 1} \tag{2.30}$$

Où :

$$C(\underline{\underline{R}}) = \frac{\lambda_{\max}}{\lambda_{\min}}$$

Est le conditionnement de $\underline{\underline{R}}$.

Ce résultat montre que la vitesse de convergence est directement liée au conditionnement de $\underline{\underline{R}}$,

Si $C(\underline{\underline{R}})$ est proche de 1, $\zeta(\underline{\underline{A}})$ est proche de 0, et la vitesse de convergence est très élevée.

Au contraire, si $C(\underline{\underline{R}}) \rightarrow \infty$ alors $\zeta(\underline{\underline{A}}) \rightarrow 1$, et l'algorithme est très lent à converger.

En d'autres termes, l'algorithme converge d'autant plus lentement que la matrice de covariance est mal conditionnée.

2.3.2. Autre présentation de la méthode du gradient :

Cette autre présentation s'appuie sur le résultat suivant :

$$\forall \mu < \frac{1}{\lambda_{\max}(\underline{R})} \quad \underline{R}^{-1} = \sum_{k=0}^{\infty} (\underline{I} - \mu \underline{R})^k$$

On peut montrer cette égalité de la même façon que pour établir la somme d'une suite géométrique-convergente.

Dans ce cas :

$$\underline{w} = \underline{R}^{-1} \underline{R}_{du} = \sum_{k=0}^{\infty} (\underline{I} - \mu \underline{R})^k \underline{R}_{du}$$

En posant alors :

$$\underline{w}(n) = \sum_{k=0}^n (\underline{I} - \mu \underline{R})^k \underline{R}_{du} \tag{2.31}$$

On a :

$$\begin{aligned} \underline{w}(n+1) &= \sum_{k=0}^{n+1} (\underline{I} - \mu \underline{R})^k \underline{R}_{du} \\ &= (\underline{I} - \mu \underline{R}) \left[\sum_{k=0}^n (\underline{I} - \mu \underline{R})^k \underline{R}_{du} + \mu (\underline{I} - \mu \underline{R})^0 \underline{R}_{du} \right] \\ &= (\underline{I} - \mu \underline{R}) \underline{w}(n) + \mu \underline{R}_{du} \end{aligned}$$

Soit :

$$\underline{w}(n+1) = \underline{w}(n) + \mu (\underline{R}_{du} - \underline{R}_{du} \underline{w}(n)) \tag{2.32}$$

On retrouve donc l'algorithme du gradient, qui, vu comme ceci, consiste à inverser la matrice \underline{R}_{uu} à l'aide du développement en série de \underline{R}_{uu}^{-1} .

2.4. Algorithme du gradient stochastique :

L'algorithme du gradient stochastique, ou plus exactement la famille des algorithmes de gradient stochastique, consiste à remplacer le gradient :

$$\nabla J|_{\underline{w}=\underline{w}(n)} = 2 \underline{R}_{uu} \underline{w}(n) - 2 \underline{R}_{du}$$

Quantité déterministe, exacte, par une approximation du gradient calculée à partir des données disponibles. Les données étant considérées comme aléatoires, le gradient estimé devient lui aussi une quantité aléatoire et l'algorithme devient stochastique. Parmi cette famille d'algorithmes, le plus connu est l'algorithme LMS (Least Mean Square).

Le problème majeur dans le gradient déterministe est que \underline{R}_{uu} et \underline{R}_{du} sont évidemment inconnus, on approchera donc ces grandeurs déterministes par des estimées $\underline{R}_{uu}(n)$ et $\underline{R}_{du}(n)$ à l'instant n.

Dans le cas du **LMS**, on choisit les estimées les plus simples possibles, à savoir :

$$\underline{R}_{uu}(n) = \underline{u}(n)\underline{u}^H(n) \text{ Et } \underline{R}_{du}(n) = d(n)\underline{u}(n)$$

Ce sont simplement les estimées instantanées des corrélations.

La relation permettant la mise à jour du filtre \underline{w} est ainsi maintenant :

$$\underline{w}(n+1) = \underline{w}(n) + \mu \underline{u}(n) (d^*(n) - \underline{u}^H(n)\underline{w}(n)) \quad (2.33)$$

Il est immédiat de montrer que cette approche est celle qui consiste à prendre l'algorithme du gradient sur le critère instantané : minimiser $|e(n)|^2$ (au lieu du critère statistique : minimiser $E\{|e(n)|^2\}$).

L'algorithme peut en fait se décomposer en :

- Calcul de la sortie du filtre : $y(n) = \underline{w}^H(n)\underline{u}(n)$
- Calcul de l'erreur d'estimation : $e(n) = d(n) - y(n)$
- Mise à jour : $\underline{w}(n+1) = \underline{w}(n) + \mu e^*(n)\underline{u}(n)$

Au temps n , on calcule donc la sortie du filtre et l'erreur d'estimation, en utilisant le filtre courant $\underline{w}(n)$.

Le filtre est ensuite mis à jour en ajoutant au filtre courant, le terme correctif :

$$\mu e^*(n)\underline{u}(n)$$

On remarquera que $\underline{w}(n)$ est maintenant un processus aléatoire, ceci signifie que pour des jeux de données différents, les trajectoires obtenues seront différentes.

On peut s'intéresser à la convergence de la trajectoire moyenne $E\{\underline{w}(n)\}$ et étudier:

$$E\{\underline{w}(n+1)\} = E\{\underline{w}(n)\} + \mu (\underline{R}_{du} - \underline{R}_{uu} E\{\underline{w}(n)\})$$

Si on suppose que $\underline{w}(n)$ est indépendant de $\underline{u}(n)$ (ce qui est très faux) on a :

$$E\{\underline{u}(n)\underline{u}^H(n)\underline{w}(n)\} = \underline{R}_{uu} E\{\underline{w}(n)\}$$

et la trajectoire moyenne suit l'algorithme du gradient déterministe.

Dans les autres cas (de façon réaliste), on n'a pas :

$$E\{\underline{u}(n)\underline{u}^H(n)\underline{w}(n)\} = \underline{R}_{uu} E\{\underline{w}(n)\}$$

Car $\underline{w}(n)$ a toutes les chances d'être corrélé à $\underline{u}(n)$.

La trajectoire moyenne n'est donc pas nécessairement celle du gradient déterministe. On peut cependant espérer qu'elle en soit proche, et que le caractère aléatoire se traduise par des fluctuations autour de cette valeur moyenne.

Comme les estimées instantanées sont très variables, on pourrait penser que l'algorithme LMS a des performances très pauvres. Cependant, comme l'algorithme est par nature récursif, il moyenne ces estimées au cours des itérations, ce qui permet d'obtenir des performances acceptables.

2.4.1. Convergence et stabilité :

On étudie la convergence en s'appuyant sur l'hypothèse d'indépendance entre $\underline{u}(n)$ et $\underline{w}(n)$. Rappelons que cette hypothèse est atrocement fautive. Les résultats corrects (très techniques) sont établis dans le livre d'O. Macchi [5].

Dans ce cas (indépendance entre $\underline{u}(n)$ et $\underline{w}(n)$) si on s'intéresse à la convergence de la trajectoire moyenne $E\{\underline{w}(n)\}$ on a simplement :

$$E\{\underline{w}(n+1)\} = E\{\underline{w}(n)\} + \mu E\{\underline{u}(n)(d^*(n) - \underline{u}^H(n)\underline{w}(n))\}$$

Ceci étant simplement l'algorithme du gradient déterministe, on aura convergence si :

$$\mu < \frac{2}{\lambda_{\max}}$$

(Pour un pas fixe).

Une étude plus précise (en moyenne quadratique), mais qui repose également sur des hypothèses contestables conduit à la condition plus contraignante :

$$\mu < \frac{2}{\text{Trace}(R_{uu})}$$

Où l'on pourra estimer la trace comme $\frac{1}{M} \text{Trace}(M)$ si M est la dimension de la matrice. En pratique, on prendra Γ de l'ordre de 2 ou 3.

2.4.1.1. Comportement de l'erreur :

L'erreur définie par $e(n) = d(n) - \underline{u}^T(n)\underline{w}$ peut être exprimée en fonction de l'erreur optimale obtenue à partir du filtre optimal :

$$\hat{e}(n) = d(n) - \underline{u}^T(n)\underline{y}$$

D'où :

$$e(n) = \hat{e}(n) - \underline{u}^T(n)(\underline{w} - \underline{y}) = \hat{e}(n) - \underline{u}^T(n)\underline{v}(n)$$

Où $\underline{v}(n)$ est l'écart ou filtre optimal, Cet écart vérifie l'équation récursive :

$$\underline{v}(n+1) = (\underline{I} - \mu \underline{u}(n)\underline{u}^T(n))\underline{v}(n) + \mu \hat{e}(n)\underline{u}(n)$$

La solution s'écrit sous la forme :

$$\underline{v}(n+1) = \underline{Q}_n \underline{v}(0) + \underline{F}_{n+1} \tag{2.34}$$

Avec :

$$\underline{Q}_n = \prod_{j=0}^{j=n} ((\underline{I} - \mu \underline{u}(j)\underline{u}^T(j)))$$

Et \underline{F}_n qui suit :

$$\underline{F}(n+1) = (\underline{I} - \mu \underline{u}(n)\underline{u}^T(n))\underline{F}(n) + \mu \hat{e}(n)\underline{u}(n)$$

Cette relation fait apparaître deux termes :

- $Q_{\underline{y}(0)}$ représente la contribution des conditions initiales et correspond à une réponse transitoire.
- E_{n+1} résulte de l'excitation permanente et représente une fluctuation autour de la solution optimale.

On note alors :

$$\underline{y}(n+1) = \underline{y}(n+1)^{tr} + \underline{y}(n+1)^{fl} \tag{2.35}$$

L'erreur d'estimation devient alors :

$$e(n) = \hat{e}(n) - \underline{u}^T(n) \underline{y}(n)^{tr} - \underline{u}^T(n) \underline{y}(n)^{fl} \tag{2.36}$$

On montre que le transitoire tend vers 0 lorsque n tend vers l'infini presque sûrement.

L'erreur moyenne quadratique, lorsque n tend vers l'infini est alors :

$$\begin{aligned} J &= E \left\{ \hat{e}(n)^2 \right\} + E \left\{ \left| \underline{u}^T(n) \underline{y}(n)^{fl} \right|^2 \right\} \\ &= J_{\min} + E \left\{ e^{fl}(n)^2 \right\} \end{aligned} \tag{2.37}$$

L'erreur quadratique liée à la fluctuation est bornée :

$$E \left\{ e^{fl}(n)^2 \right\} \leq \sim J_{\min} \frac{Tr(\underline{R}_{\underline{uu}})}{2 - \sim \underline{\alpha} Tr(\underline{R}_{\underline{uu}})} \quad r \geq 1 \tag{2.38}$$

Notons bien que cette fluctuation provient du fait que la trajectoire est aléatoire, et que c'est la moyenne de $\underline{w}(n)$ qui converge vers \underline{w} .

En effet, si à l'étape n, n étant aussi grand que l'on veut, $\underline{w}(n) = \underline{w}$ alors, à l'étape (n+1) :

$$\underline{w}(n+1) = \underline{w} + \sim \left(\underline{u}(n) \underline{d}(n) - \underline{u}^T(n) \underline{w} \right) \tag{2.39}$$

Qui est différent de \underline{w} , sauf cas très particulier du fait que l'excitation est aléatoire.

De ces résultats, on déduit que :

- plus \sim est grand, mais vérifiant la condition de stabilité, plus la convergence est rapide, mais plus la variance résiduelle est importante, ce qui se traduit par des fluctuations importantes autour de la trajectoire moyenne.
- Pour de faibles valeurs de \sim , la convergence est lente, mais l'erreur résiduelle a une faible variance, c'est-à-dire que l'on a une trajectoire « presque » déterministe.
- La convergence de l'algorithme requiert donc un compromis entre **vitesse et fidélité**.

2.4.1.2. Convergence de l'algorithme à pas variable :

Afin d'accélérer la convergence, il est intéressant de faire varier le pas d'adaptation \sim :

- \sim doit être grand au départ, lorsque l'on est loin de l'optimum
- \sim doit être faible lorsque l'on se retrouve au voisinage de l'optimum.

On peut alors utiliser une séquence $\{ \sim_n \}$ de pas variable.

Dans ce cas, l'algorithme converge en moyenne quadratique, c'est à dire que :

$$E \left\{ \left| \underline{w}(n) - \underline{w} \right|^2 \right\} \xrightarrow{n \rightarrow \infty} 0$$

Si les conditions suffisantes de Robbin et Monroe sont satisfaites :

- $\sim_n > 0$ quelque soit n
- $\sum \sim_n$ diverge
- $\sum \sim_n^2$ converge

2.4.2. Poursuite – Adaptativité :

L'algorithme présenté jusqu'ici a été présenté sous l'angle d'une implantation récursive d'un filtre de Wiener, pour lequel on a implicitement supposé les signaux stationnaires. On s'est alors intéressé à la convergence vers la solution \underline{w} .

A partir de l'implantation récursive, il est également possible de voir l'algorithme comme un algorithme adaptatif, c'est-à-dire capable de suivre des modifications lentes, en permanence, du filtre optimal.

Celui-ci est alors dépendant du temps :

$$\underline{w} \longrightarrow \underline{w}(n)$$

On peut tout de suite noter que si l'on désire que le filtre soit adaptatif, on ne peut pas le laisser converger en prenant une séquence $\{\sim_n\}$, avec :

$$\sim_n \longrightarrow 0 \text{ Lorsque } n \longrightarrow \infty$$

L'algorithme doit conserver une « capacité de réaction » permettant la poursuite des non stationnarités.

On considère donc que $\underline{w}(n)$ évolue maintenant suivant une marche aléatoire :

$$\underline{w}(n) = \underline{w}(n-1) + \underline{v}(n) \tag{2.40}$$

Où $\underline{v}(n)$ est un bruit vectoriel, de matrice de covariance $\underline{\Gamma}$. On définit alors un nouvel écart au filtre optimal :

$$\underline{v}(n) = \underline{w}(n) - \underline{w}(n)$$

On a toujours :

$$\begin{aligned} e(n) &= d(n) - \underline{u}^T(n) \underline{w}(n) - \underline{u}^T(n) (\underline{w}(n) - \underline{w}(n)) \\ &= \hat{e}(n) - \underline{u}^T(n) \underline{v}(n) \end{aligned}$$

En résolvant récursivement, on a :

$$\underline{v}(n+1) = \underline{Q}_{\underline{v}} \underline{v}(0) + \underline{F}_{n+1} + \underline{E}_{n+1}$$

Où \underline{E}_{n+1} est la suite aléatoire « supplémentaire », correspondant à l'ajout de $\underline{v}(n)$ dans le modèle et qui obéit à l'équation récursive :

$$\underline{E}(n+1) = \left(\underline{I} - \sim \underline{u}(n) \underline{u}^T(n) \right) \underline{E}(n) - \underline{v}(n+1)$$

Il s'en suit que :

$$e(n) = \hat{e}(n) + e^{tr}(n) + e^{fl}(n) + e^r(n)$$

Et une fois disparu le transitoire :

$$J = J_{\min} + E\{e^{fl}(n)^2\} + E\{e^r(n)^2\} \tag{2.41}$$

Avec :

$$E\{e^r(n)^2\} = \frac{Tr(\underline{\Gamma})}{2 - \mu \cdot Tr(\underline{R}_{uu})} \quad r \geq 1$$

On rappelle que :

$$E\{e^{fl}(n)^2\} \leq J_{\min} \frac{Tr(\underline{\Gamma})}{2 - \mu \cdot Tr(\underline{R}_{uu})} \quad r \geq 1$$

La minimisation de l'erreur quadratique conduit alors à choisir un pas optimal qui effectue un compromis entre ces deux erreurs.

2.4.3. Algorithme du signe :

Parmi les variantes simplifiées de l'algorithme LMS, l'algorithme du signe est intéressant, On remplace simplement l'erreur d'estimation $e(n)$ par son signe, ce qui conduit à la récurrence :

$$\underline{w}(n+1) = \underline{w}(n) + \mu(n) \cdot \text{signe}(e(n)) \tag{2.42}$$

On évite ainsi d'effectuer la multiplication dans la relation d'actualisation des coefficients, ce qui conduit ainsi à une complexité de M multiplications par itération.

Il est à noter qu'en dépit de cette simplification extraordinaire, la convergence reste convenable dans beaucoup d'applications, et en particulier en égalisation adaptative.

2.4.4. Algorithme LMS normalisé :

Dans la récurrence de l'algorithme LMS standard (avec des coefficients de filtre réels)

$$\underline{w}(n+1) = \underline{w}(n) + \mu(n) \cdot (\underline{d}(n) - \underline{u}^H(n) \cdot \underline{w}(n)) \tag{2.43}$$

La dynamique des fluctuations dépend de la dynamique du signal d'entrée $\underline{u}(n)$. On dit qu'on a une amplification du bruit par le gradient.

Reprenons l'équation du gradient déterministe :

$$\underline{w}(n+1) = \underline{w}(n) - \frac{1}{2} \cdot \nabla J \Big|_{\underline{w}=\underline{w}(n)}$$

Avec :

$$\nabla J \Big|_{\underline{w}=\underline{w}(n)} = 2\underline{R}_{uu} \cdot \underline{w} - 2\underline{R}_{ud} = \underline{\nabla}_n$$

A l'instant $n+1$, on a :

$$J(n+1) = E\{e(n+1)^2\}$$

Avec :

$$\begin{aligned} e(n+1) &= d(n+1) - \underline{w}^T \underline{u}(n+1) \\ &= d(n+1) - \underline{w}^T \underline{u}(n+1) + \frac{1}{2} \tilde{\mu}_n \nabla_n^T u(n+1) \end{aligned}$$

Si on cherche le pas optimal $\tilde{\mu}_n$ qui minimise $J(n+1)$, on doit avoir :

$$\frac{dJ(n+1)}{d\tilde{\mu}_n} = 2E \left\{ e(n+1) \frac{de(n+1)}{d\tilde{\mu}_n} \right\} = 0$$

Soit :

$$E \left\{ \nabla_n^T [d(n+1) \underline{u}(n+1) - \underline{w}_n^T \underline{u}(n+1)] \underline{u}^T(n+1) \nabla_n + \frac{1}{2} \mu_n \underline{u}(n+1) \underline{u}^T(n+1) \nabla_n \right\} = 0$$

Dans l'algorithme du gradient déterministe ∇_n est certain, la relation précédente s'écrit donc :

$$\nabla_n^T \underline{R}_{du} - \underline{w}_n^T \underline{R}_{uu} \nabla_n + \frac{1}{2} \mu_n \nabla_n^T \underline{R}_{uu} \nabla_n = 0$$

Soit finalement :

$$\tilde{\mu}_n = \frac{2 \left(\underline{w}_n^T \underline{R}_{uu} \nabla_n - \nabla_n^T \underline{R}_{du} \right)}{\underline{w}_n^T \underline{R}_{uu} \nabla_n}$$

L'estimée instantanée de \underline{R}_{uu} est :

$$\underline{R}_{uu}(n) = \underline{u}(n) \underline{u}^T(n)$$

Celle de \underline{R}_{du} est :

$$\underline{R}_{du} = d(n) \underline{u}_n$$

Celle de ∇_n est :

$$\nabla_n = 2 \left(d(n) \underline{u}_n - \underline{u}_n \underline{u}_n^T \underline{w}_n \right)$$

En remplaçant \underline{R}_{uu} , \underline{R}_{du} et ∇_n par leurs estimées respectives il vient :

$$\tilde{\mu}_n = \frac{1}{\underline{u}_n^T \underline{u}_n} \tag{2.44}$$

Ce qui revient à prendre un pas inversement proportionnel à la puissance « instantanée » du signal d'entrée.

2.5. Formulation du filtrage de Wiener dans le cadre des moindres carrés :

Nous avons résolu précédemment le problème du filtrage de Wiener en nous appuyant sur la fonction de coût déterministe :

$$J = E \{ e(n)^2 \}$$

L'inconvénient de cette approche est que les grandeurs statistiques sont difficiles à obtenir. D'autre part comme nous l'avons déjà signalé, la solution optimale obtenue n'est valable que pour un signal stationnaire. L'approche du gradient stochastique nous a permis, à partir de ce critère statistique de nous affranchir de ces inconvénients mais nous pouvons choisir de reformuler le problème dans le cadre des moindres carrés, c'est-à-dire rechercher le filtre qui minimise la somme des carrés des erreurs, jusqu'à l'instant considéré n .

Nous donnerons dans le paragraphe « Algorithme des moindres carrés récursifs » une approche récursive de ce problème (autorisant une évolution pour un fonctionnement adaptatif).

Dans un premier temps nous allons considérer le cas dans lequel on connaît le signal d'entrée u_n sur N points et trouver le filtre optimal pour ce bloc de données, pour simplifier les équations nous considérerons maintenant le cas d'un filtre à coefficients réels.

2.5.1. Critère des moindres carrés pour un bloc de données de N points (u_0, u_1, \dots, u_{N-1}) :

$$J_{MC} = \sum_i (e(i)^2) \quad (2.45)$$

Avec :

$$y(i) = \sum_{l=0}^{M-1} w_l u(i-l)$$

Pour préciser les bornes de la sommation sur i , il faut prendre en compte le fait que pour $i < M - 1$ et $i > N - 1$, $y(i)$ dépend de valeurs d'entrée inconnues.

Différentes hypothèses sont possibles concernant les valeurs inconnues de l'entrée :

- Soit considérer $u(i) = 0$ pour $i < 0$
- Soit considérer $u(i) = 0$ pour $i > N - 1$
- Soit ignorer ces valeurs de l'entrée

On note la relation précédente :

$$\underline{y} = \underline{A}^T \underline{w}$$

La matrice \underline{A} est appelée matrice de données (ANNEXE 4).

Suivant les hypothèses, on pourra extraire plusieurs matrices :

- Si on refuse d'affecter les valeurs inconnues, on extrait une matrice de Toeplitz de dimension $(M)(N - M + 1)$, qui ne contient que les valeurs connues de l'entrée. En d'autres termes, on ne pourra calculer l'erreur que pour $i = M - 1$ à $i = N - 1$. Cette forme est appelée « forme covariance ».
- Si on affecte la valeur 0 aux échantillons de l'entrée pour $i < 0$, on parle de préfenêtrage, et on extrait la forme préfenêtrée, de dimension $M \times N$. On peut calculer $e(i)$ pour $i = 0$ à $N - 1$

- Si on affecte 0 aux valeurs inconnues pour $i > N-1$, on a une forme post-fenêtrée, et on peut calculer $e(i)$ pour $i = M-1$ à $N+M-2$.
- Enfin, en utilisant à la fois le pré et post fenêtrage, on obtient la forme «autocorrélation » de dimension $M \times (N+M-2)$, et $e(i)$ est calculable pour $i = 0$ à $N+M-2$.

Les termes de covariance et autocorrélation sont consacrés, en particulier en traitement de la parole. Cependant, il n’y a là aucun (ou peu) de rapport avec les définitions des fonctions et matrice de covariance et d’autocorrélation.

Notons de plus que le fait de compléter la séquence par des zéros n’est pas la seule solution. En particulier, il est possible de périodiser la séquence, ce qui conduit à des matrices circulantes. Suivant les hypothèses faites sur le fenêtrage, on ne minimisera donc pas nécessairement l’erreur sur tout l’horizon :

$$J = \sum_{i=M-1}^{N-1} e(i)^2$$

Pour la forme covariance, par exemple.

On note \underline{d} le vecteur de la sortie désirée, \underline{y} le vecteur de sortie, et \underline{A} la matrice de donnée.

Le vecteur d’erreur \underline{e} vaut :

$$\underline{e} = \underline{y} - \underline{d}$$

Le critère J s’écrit :

$$J = \underline{e}^T \underline{e} = (\underline{y} - \underline{d})^T (\underline{y} - \underline{d})$$

Il ne reste plus qu’à exprimer le gradient de J par rapport à \underline{w} :

$$\nabla J|_{\underline{w}} = 2\underline{A}(\underline{A}^T \underline{w} - \underline{d})$$

Et on en déduit que :

$$\underline{A} \underline{A}^T \underline{w} = \underline{A} \underline{d}$$

En développant $\underline{A} \underline{A}^T$ et $\underline{A} \underline{d}$, il est facile de voir que les termes génériques valent :

$$(\underline{A} \underline{A}^T)_{ij} = \sum_l u(l-i)u(l-i) = \hat{r}_{uu}(i-j)$$

$$\underline{A} \underline{d} = \sum_l u(l-i)d(l) = \hat{r}_{du}(i)$$

On retrouve ainsi les équations normales (équation de Wiener Hopf) en utilisant des corrélations empiriques (ou temporelles).

Notons que suivant les hypothèses effectuées sur le fenêtrage, ces estimées ne seront pas identiques.

En particulier, la matrice de corrélation empirique est de Toeplitz avec la méthode d’autocorrélation, alors qu’elle ne l’est pas forcément dans les autres hypothèses.

On peut encore noter que :

$$\underline{R}_{uu} = \underline{A} \underline{A}^T$$

Est par construction définie non négative (c’est un « carré » matriciel) et symétrique (à symétrie hermitienne).

2.5.1.1. Théorème de projection :

A l'optimum, le gradient s'annule :

$$\underline{A}(\underline{A}^T \underline{w} - \underline{d}) = \underline{0} \Leftrightarrow \underline{A} \underline{e} = \underline{0}$$

Par conséquent, en multipliant par \underline{w}^T on obtient :

$$\underline{y}^T \underline{e} = 0$$

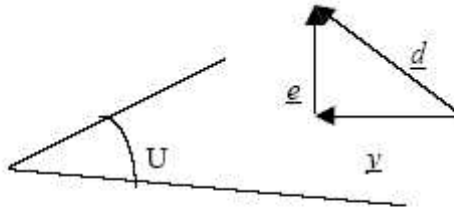


Figure.2.7- Théorème de projection.

2.5.2. Solution de l'équation normale :

A partir de :

$$\underline{A} \underline{A}^T \underline{w} = \underline{A} \underline{d}$$

On obtient :

$$\underline{w}_{MC} = (\underline{A} \underline{A}^T)^{-1} \underline{A} \underline{d}$$

Si $\underline{A} \underline{A}^T$ est inversible, donc le filtre optimal pour le critère des moindres carrés J_{MC} pour le bloc de données de N points $(u_0, u_1, \dots, u_{N-1})$.

Les inconvénients de cette approche sont les suivants :

- La solution optimale s'obtient en inversant la matrice ce qui représente une grosse charge de calcul.
- Le fonctionnement en bloc ne permet pas à priori de prendre en compte une nouvelle donnée u_N pour ajuster le filtre sans tout recalculer.
- L'algorithme peut être vu comme adaptatif si on considère des blocs successifs de N points mais ne peut pas suivre des non stationnarités à évolution rapide.

Ces différents inconvénients justifient la recherche d'une approche récursive pour ce critère des moindres carrés.

2.6. Algorithme des moindres carrés récursifs :

L'approche des moindres carrés consiste à minimiser la somme des carrés des erreurs :

$$J_{MC} = \sum_{i=1}^{i_2} (e(i)^2).$$

On obtient alors la solution des moindres carrés :

$$\underline{w}_{MC} = \arg \min_{\underline{w}} \left(\underline{A}^T \underline{w} - \underline{d} \right)^T \left(\underline{A}^T \underline{w} - \underline{d} \right) \quad \text{Où } \underline{A} \text{ est la matrice de données.}$$

La solution exacte s'écrit :

$$\underline{w}_{MC} = \left(\underline{A} \underline{A}^T \right)^{-1} \underline{A} \underline{d}$$

Nous avons vu que la matrice $\left(\underline{A} \underline{A}^T \right)$ (correspond à une estimation de la matrice de corrélation \underline{R}_{uu} , et le produit $\underline{A} \underline{d}$ est l'estimée de l'intercorrélation \underline{R}_{ud} .

L'approche des moindres carrés est par essence une approche bloc, c'est-à-dire qu'on traite un bloc de données pour construire \underline{A} , puis en déduire \underline{w}_{MC} .

L'inversion de $\left(\underline{A} \underline{A}^T \right)$ est coûteuse (de l'ordre de M^3).

Par ailleurs, la nature « bloc » de l'algorithme empêche à priori de prendre en compte simplement une nouvelle donnée, en profitant du résultat déjà obtenu :

Si on a résolu $\underline{w}_{MC} = \arg \min_{\underline{w}} \sum_{i=1}^n (e(i)^2)$ Peut-on relier $\underline{w}_{MC}^{(n+1)} = \arg \min_{\underline{w}} \sum_{i=1}^{n+1} (e(i)^2)$ à $\underline{w}_{MC}^{(n)}$?

La réponse est bien entendu affirmative. L'algorithme correspondant est l'algorithme des moindres carrés récursifs, qui permet de mettre à jour la solution au cours du temps. Notons dès à présent qu'il ne s'agit pas, tel quel, d'un algorithme adaptatif : dans la solution est contenue toute l'histoire du signal jusqu'à l'instant n, avec une hypothèse implicite de stationnarité. Pour obtenir un algorithme adaptatif on pourra introduire un « facteur d'oubli » sur les données les plus anciennes (voir paragraphe correspondant).

2.6.1. Construction de l'algorithme des MCR :

On pose :

$$\underline{w}_{MC}^{(n)} = \left(\underline{R}_{uu}^{(n)} \right)^{-1} \underline{R}_{ud}^{(n)}$$

Où $\underline{R}_{uu}^{(n)}$ et $\underline{R}_{ud}^{(n)}$ sont la matrice $\left(\underline{A} \underline{A}^T \right)$ et le vecteur $\underline{A} \underline{d}$ construits à partir des données jusqu'à l'instant n .

Dorénavant nous considérerons de plus que $i_1 = 0$ (hypothèse préférée)

La matrice $\underline{R}_{uu}^{(n)}$ possède alors la structure suivante :

$$\underline{R}_{uu}^{(n)} = \sum_{i=0}^n \underline{u}(i) \underline{u}^T(i)$$

Dans ces conditions, à l'instant $(n+1)$:

$$\underline{\underline{R}}^{(n+1)} = \underline{\underline{R}}^{(n)} + \underline{u}(n+1)\underline{u}^T(n+1)$$

Et on dispose d'une équation récursive pour estimer la matrice de corrélation.

Le problème est maintenant que pour obtenir une formulation récursive pour le vecteur optimal, il faut avoir une formulation récursive sur l'inverse de la matrice de corrélation. Pour cela on utilise, le **lemme d'inversion matriciel**.

2.6.1.1. Lemmes d'inversion matriciels :

Si \underline{A} \underline{B} \underline{C} \underline{D} sont des matrices de dimensions convenables :

$$\left(\underline{A} + \underline{B}\underline{C}\underline{D}\right)^{-1} = \underline{A}^{-1} - \underline{A}^{-1}\underline{B}\left(\underline{C}^{-1} + \underline{D}\underline{A}^{-1}\underline{B}\right)^{-1}\underline{D}\underline{A}^{-1} \tag{2.46}$$

Ce lemme se démontre par vérification directe (**ANNEXE 5**).

Dans notre cas, il s'agit d'inverser :

$$\underline{\underline{R}}^{(n+1)} = \underline{\underline{R}}^{(n)} + \underline{u}(n+1)\underline{u}^T(n+1)$$

On prendra donc :

$$\underline{A} = \underline{\underline{R}}^{(n)}, \underline{B} = \underline{u}(n+1), \underline{C} = 1, \underline{D} = \underline{u}^T(n+1)$$

En appliquant le lemme d'inversion matricielle, il vient alors :

$$\left(\underline{\underline{R}}^{(n+1)}\right)^{-1} = \left(\underline{\underline{R}}^{(n)}\right)^{-1} - \frac{\left(\underline{\underline{R}}^{(n)}\right)^{-1}\underline{u}(n+1)\underline{u}^T(n+1)\left(\underline{\underline{R}}^{(n)}\right)^{-1}}{1 + \underline{u}^T(n+1)\left(\underline{\underline{R}}^{(n)}\right)^{-1}\underline{u}(n+1)}$$

Pour alléger l'écriture, on pose :

$$\underline{\underline{K}}_n = \left(\underline{\underline{R}}^{(n)}\right)^{-1}$$

On a alors :

$$\underline{\underline{K}}_{n+1} = \underline{\underline{K}}_n - \frac{\underline{\underline{K}}_n \underline{u}(n+1)\underline{u}^T(n+1)\underline{\underline{K}}_n}{1 + \underline{u}^T(n+1)\underline{\underline{K}}_n \underline{u}(n+1)} \tag{2.47}$$

On notera que cette formule permet de calculer l'inverse de la matrice de corrélation, sans calcul explicite d'inverse.

Le vecteur d'intercorrélation $\underline{R}_{ud}^{(n)}$ s'écrit quant à lui, sous forme récursive, selon :

$$\underline{R}_{ud}^{(n+1)} = \underline{R}_{ud}^{(n)} + d(n+1)\underline{u}(n+1)$$

Le filtre optimal à l'itération $(n+1)$ s'écrit donc :

$$\underline{w}_{MC}^{(n+1)} = \left(\underline{\underline{R}}^{(n+1)}\right)^{-1} \underline{R}_{ud}^{(n+1)} = \underline{\underline{K}}_{n+1} \left(\underline{R}_{ud}^{(n)} + d(n+1)\underline{u}(n+1)\right)$$

Pour simplifier plus loin nous explicitons le dernier terme de cette expression : $\underline{\underline{K}}_{n+1} \underline{u}(n+1)$

$$\begin{aligned} \underline{K}_{n+1} \underline{u}(n+1) &= \left(\underline{K}_{n+1} - \frac{\underline{K}_{n+1} \underline{u}(n+1) \underline{u}^T(n+1) \underline{K}_{n+1}}{1 + \underline{u}^T(n+1) \underline{K}_{n+1} \underline{u}(n+1)} \right) \underline{u}(n+1) \\ &= \underline{K}_{n+1} \underline{u}(n+1) \left(1 - \frac{\underline{u}^T(n+1) \underline{K}_{n+1} \underline{u}(n+1)}{1 + \underline{u}^T(n+1) \underline{K}_{n+1} \underline{u}(n+1)} \right) \\ &= \frac{\underline{K}_{n+1} \underline{u}(n+1)}{1 + \underline{u}^T(n+1) \underline{K}_{n+1} \underline{u}(n+1)} \end{aligned}$$

Nous avons donc :

$$\begin{aligned} \underline{w}^{(n+1)} &= \underline{K}_{n+1} \left(\underline{R}_{ind}^{(n)} + d(n+1) \right) \underline{u}(n+1) \\ &= \left(\underline{K}_{n+1} - \frac{\underline{K}_{n+1} \underline{u}(n+1) \underline{u}^T(n+1) \underline{K}_{n+1}}{1 + \underline{u}^T(n+1) \underline{K}_{n+1} \underline{u}(n+1)} \right) \left(\underline{R}_{ind}^{(n)} + d(n+1) \right) \underline{u}(n+1) \\ &= \underline{w}^{(n)} - \frac{\underline{K}_{n+1} \underline{u}(n+1) \underline{u}^T(n+1) \underline{K}_{n+1} \left(\underline{R}_{ind}^{(n)} + d(n+1) \right)}{1 + \underline{u}^T(n+1) \underline{K}_{n+1} \underline{u}(n+1)} \\ &= \underline{w}^{(n)} - \frac{\underline{K}_{n+1} \underline{u}(n+1)}{1 + \underline{u}^T(n+1) \underline{K}_{n+1} \underline{u}(n+1)} \left(\underline{u}^T(n+1) \underline{K}_{n+1} \underline{R}_{ind}^{(n)} - d(n+1) \right) \end{aligned}$$

Soit finalement :

$$\underline{w}^{(n+1)} = \underline{w}^{(n)} - \underline{K}_{n+1} \underline{u}(n+1) \left(d(n+1) - \underline{u}^T(n+1) \underline{w}^{(n)} \right) \tag{2.48}$$

Cette formule, définissant le nouveau filtre optimal à partir du filtre est aussi appelée, par analogie, formule de Kalman.

Les équations (2.47) et (2.48) constituent l’**algorithme des moindres carrés récursifs**.

Dans l’algorithme développé ci-dessus, on n’a pas utilisé toutes les caractéristiques de la matrice de corrélation, et en particulier le fait que la matrice de corrélation (dans le cas de l’hypothèse autocorrélation) est de Toeplitz.

On peut alors obtenir une formule de renouvellement plus simple.

Par ailleurs, on peut effectuer à la fois une récurrence sur le temps, mais aussi sur l’ordre (Ordre Récursive Adaptive Least Squares).

On peut également trouver une récurrence sur le gain $\underline{K}_{n+1} \underline{u}(n+1)$ plutôt que sur l’inverse de la matrice de corrélation. Les algorithmes obtenus sont les algorithmes des moindres carrés récursifs rapides, qui peuvent poser des problèmes d’instabilité numérique.

On notera que le terme d’erreur s’écrit :

$$e(n+1) = d(n+1) - \underline{u}^T(n+1) \underline{w}^{(n)}$$

Alors qu’on avait dans le cas des algorithmes du gradient :

$$e(n+1) = d(n+1) - \underline{u}^T(n+1) \underline{w}(n+1)$$

La première erreur est l'erreur à priori, c'est-à-dire l'erreur effectuée avant d'avoir mis à jour le filtre avec la nouvelle donnée $u(n+1)$.

Cette erreur est différente de l'erreur à posteriori, dans laquelle le filtre a été mis à jour.

2.6.2. Problèmes d'initialisation :

La mise en oeuvre pratique des algorithmes des MCR requiert le choix des valeurs initiales \underline{K}_0 et $\underline{w}^{(0)}$

La matrice de corrélation, à l'étape n s'exprimant comme une somme de dyades :

$$\underline{R}_{uu}^{(n)} = \sum_{i=0}^n u(i) u^T(i)$$

Cette matrice est singulière tant que $n < M$, ce qui empêche de démarrer la récurrence.

Deux solutions sont alors possibles :

- évaluer la matrice de corrélation et son inverse pour un $n > M$
- modifier légèrement la formule d'estimation de en introduisant un « talon » :

$$\underline{R}_{uu}^{(n)} = \sum_{i=0}^n u(i) u^T(i) + \sigma \underline{I}$$

Où σ est un scalaire positif très faible.

Il s'en suit que :

$$\underline{R}_{uu}^{(n)} = \sigma \underline{I} \quad \text{et} \quad \underline{K}_n = \frac{1}{\sigma} \underline{I}$$

Ce choix consiste en fait à modifier le critère initial en :

$$J' = \sigma \|\underline{w}^{(n)}\|^2 + \sum_{i=0}^n e(i)^2$$

L'intervention de la constante σ introduit une régularisation de l'algorithme, en pénalisant la norme du vecteur optimal (ce qui est équivalent à effectuer la minimisation de $J_{MC}(n)$, sous contrainte).

Il reste à choisir $\underline{w}^{(0)}$, que l'on choisit usuellement comme le vecteur nul.

2.6.3. Version adaptative – pondération exponentielle :

L’algorithme précédent n’est pas adaptatif. Pour le rendre adaptatif, il faut « oublier » le passé du signal lors de l’évolution de l’algorithme.

Pour cela, une approche classique consiste à donner un poids plus important aux valeurs les plus récentes : on modifie le critère en introduisant une pondération exponentielle :

$$J' = \sum_{i=0}^n \lambda^{n-i} e(i)^2$$

On montre alors aisément que seule l’équation de renouvellement de l’inverse de la corrélation est modifiée :

$$K_{n+1} = \frac{1}{\lambda} K_n - \frac{1}{\lambda} \frac{K_n \underline{u}(n+1) \underline{u}^T(n+1) K_n}{1 + \frac{1}{\lambda} \underline{u}^T(n+1) K_n \underline{u}(n+1)} \tag{2.49}$$

Cet algorithme ne converge plus (la matrice de corrélation reste aléatoire lorsque $n \rightarrow \infty$, mais ceci est normal puisque l’on se place dans une perspective adaptative, en permettant à l’algorithme de réagir à des non stationnarités.

2.6.4. Equation récursive sur le critère somme quadratique :

A l’instant n , pour le vecteur optimal, on a :

$$\begin{aligned} J(n) &= \sum_{i=0}^n \lambda^{n-i} e(i)^2 \\ &= \sum_{i=0}^n \lambda^{n-i} \left(d(i) - \underline{w}^{T(n)} \underline{u}(i) \right)^2 \\ &= \sum_{i=0}^n \lambda^{n-i} \left(d(i)^2 - 2 \underline{R}_{ud}^{T(n)} \underline{w}^{(n)} \right) \end{aligned}$$

Où $\underline{R}_{ud}^{T(n)}$ est l’intercorrélation « exponentielle » :

$$\underline{R}_{ud}^{T(n)} = \sum_{i=0}^n \lambda^{n-i} \underline{u}(i) d(i).$$

En posant :

$$\dagger_d^2(n) = \sum_{i=0}^n \lambda^{n-i} (d(i))^2$$

On a :

$$\dagger_d^2(n) = \lambda \dagger_d^2(n-1) + (d(n))^2$$

Et :

$$\underline{R}_{ud}^T(n) = \lambda \underline{R}_{ud}^T(n-1) + d(n) \underline{u}(n)$$

En utilisant ces deux relations, il vient :

$$J(n) = \lambda J(n-1) + \zeta_n e_n \tag{2.50}$$

Notons que :

$$e(n) = d(n) - \underline{w}^T(n) \underline{u}(n)$$

$$= d(n) - \left(\underline{w}^T(n-1) + \underline{K}_n \underline{u}(n-1) \right) \underline{u}(n)$$

Soit :

$$e(n) = \left(1 - \underline{u}^T(n) \underline{K}_n \underline{u}(n) \right) \xi_n$$

Le rapport entre les erreurs à posteriori et à priori s'appelle facteur de conversion :

$$x(n) = \frac{e(n)}{\xi(n)} = 1 - \underline{u}^T(n) \underline{K}_n \underline{u}(n)$$

2.6.5. Convergence de l'algorithme, en moyenne :

On supposera ici que :

$$d(n) = \underline{w}^T \underline{u}(n) + \hat{e}(n)$$

Où \underline{w} est le vecteur filtre exact et un bruit d'observation blanc centré, décorrélé de $\underline{u}(n)$ (notez que ce modèle est restrictif : rien ne dit, dans le filtrage de Wiener que $d(n)$ est issu d'une filtrée linéaire de $u(n)$).

En écrivant l'algorithme, on a :

$$\underline{w}^n = \underline{w}^{n+1} + \underline{K}_n \underline{u}(n) \left(d(n) - \underline{u}^T(n) \underline{w}^{n-1} \right)$$

Exprimons $\underline{R}_{ud}^{(n)}$ en remplaçant $d(i)$ par son expression :

$$\underline{R}_{ud}^{(n)} = \sum_{i=0}^n \underline{w}^T \underline{u}(i) \underline{u}(i) + \hat{e}(i) \underline{u}(i)$$

$$= \underline{R}_{uu}^{(n)} \underline{w} + \sum_{i=0}^n \hat{e}(i) \underline{u}(i)$$

En multipliant par \underline{K}_n il vient :

$$\underline{w}^{(n)} = \underline{w} + \underline{K}_n \sum_{i=0}^n \hat{e}(i) \underline{u}(i)$$

En utilisant maintenant la composition des espérances :

$$E_{X,Y} (X(Y)) = E_Y (E_{X/Y} (X/Y))$$

Et grâce à la décorrélation (l'indépendance) entre $e(n)$ et $u(n)$, et au fait que soit centré, on obtient :

$$E(\underline{w}(n)) = \underline{w} \tag{2.51}$$

L'algorithme converge donc en moyenne, ce qui est rassurant, et il converge à nombre d'itérations fini : il n'est pas nécessaire d'avoir $n \longrightarrow \infty$ pour obtenir la convergence (cas du LMS).

2.6.6. Erreur quadratique moyenne :

Avec des hypothèses irréalistes, mais qui permettent d'aboutir à un résultat bien vérifié en pratique, on montre que :

$$E\left\{\left|w(n) - \underline{w}\right|^2\right\} \propto \frac{1}{n} \sum_{i=0}^{M-1} \frac{1}{\lambda_i}$$

Ce résultat montre que l'algorithme converge en $1/n$, et que l'erreur quadratique est inversement proportionnelle aux valeurs propres. Une fois de plus, le mauvais conditionnement limite la vitesse de convergence.

2.6.7. Algorithmes sous optimaux :

L'idée des algorithmes sous optimaux est de simplifier l'équation de renouvellement de \underline{K}_n . En particulier, on obtient des algorithmes du type LMS en effectuant la simplification ultime :

$$\underline{K}_{n+1} = \sim_{n+1} \underline{I}$$

2.7. Conclusion :

Les algorithmes adaptatifs utilisés pour minimiser l'erreur quadratique sont le **LMS** (" **Least Mean Square**" ou moindre carré moyen), le **RLS** (" **récurif least square**" ou moindre carré récursif)

L'algorithme **LMS** est un membre important de la famille des algorithmes basés sur le gradient stochastique dont la caractéristique principale est sa simplicité.

L'algorithme **RLS**, basé sur les moindres carrés, possède la caractéristique très intéressante de converger plus rapidement que le **LMS**.

Chapitre N°03

Les antennes intelligentes

3.1. Rangée d'antennes linéaires :

3.1.1. Introduction aux Antennes :

Le terme d'antenne évoque d'abord un élément sensible plongé dans le milieu extérieur dans lequel se propagent des ondes. Une antenne est donc d'abord un capteur destiné à « capturer » une partie du champ extérieur (si possible sans le perturber), et à fournir un signal (généralement électrique) à un dispositif qui va le transformer de manière à satisfaire un utilisateur (démodulation, analyse spectrale, ...).

Pour augmenter la sensibilité du capteur, il est assez naturel de proposer d'accoupler un grand nombre d'éléments sensibles, c'est-à-dire d'augmenter la taille de l'antenne.

Un phénomène typique aux antennes apparaît dès que leur taille atteint ou dépasse la longueur des ondes utilisées : elles deviennent sélectives à la fois « spatialement » et « spectralement ».

Cela veut dire que le niveau du signal obtenu dépend de l'orientation de l'antenne par rapport à la direction des ondes et de la fréquence de ces ondes ; ce phénomène est assez compréhensible puisque l'antenne va combiner, suivant son orientation, des signaux en phase ou en opposition de phase.

L'utilisation de structures de grande taille (vis-à-vis de la longueur d'onde) pour augmenter la sensibilité va donc de paire avec la nécessité d'orienter l'antenne, à moins que le contexte ne justifie des antennes fixes non pointables.

La nécessité des réseaux de capteurs est apparue lorsque la solution du pointage mécanique de l'antenne s'est révélée impraticable, soit en raison de la taille de l'antenne (inertie), soit en raison du type d'application.

- Dans le domaine SONAR, le problème se pose à cause de la vitesse très faible de propagation des ondes; les réseaux de capteurs (pointés électroniquement) se sont imposés pratiquement dès l'origine de l'acoustique sous-marine.
- Dans le domaine RADAR, le problème se pose lorsque l'antenne devient si grande que son pointage mécanique soulève des difficultés insurmontables ; c'est par exemple le cas en basse fréquence ou pour les radars de veille stratégique : dans ces domaines, des « réseaux phases » (autre dénomination du pointage électronique) ont été développés.

3.1.2. Caractéristiques d'une antenne :

3.1.2.1. Principe de réciprocité :

Le principe de réciprocité stipule qu'une antenne peut être utilisée indifféremment en émission ou en réception.

3.1.2.2. Antenne isotrope :

On appelle antenne isotrope une antenne théorique rayonnant uniformément dans toutes les directions de l'espace. Une telle antenne n'a donc pas de direction de propagation privilégiée; on dit qu'elle n'est pas directive.

A une distance r la densité de puissance vaut :

$$p_{iso}(r, \theta, \phi) = \frac{P_E}{4\pi r^2} \quad (3.1)$$

Si l'antenne isotrope est intéressante pour le calcul théorique, il est important de noter qu'elle est impossible à réaliser dans la pratique...

3.1.2.3. Directivité :

La directivité (parfois appelée gain directionnel, ce qui porte à confusion...) d'une antenne est définie ainsi :

$$D(q,j) = \frac{p(r,q,j)}{P_{iso}} \quad (3.2)$$

On admet que p_{iso} est la densité de puissance émise par une antenne isotrope qui aurait la même puissance émise P_E .

La directivité indique dans quelles directions la densité de puissance est meilleure ou moins bonne que celle de l'antenne isotrope. La directivité ne dépend pas de r , car les deux densités décroissent en $1/r^2$.

3.1.2.4. Gain en puissance et rendement :

Le rendement d'une antenne est défini ainsi :

$$h = \frac{P_E}{P_F} \quad (3.3)$$

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

Les différents gains en puissance d'une antenne sont les suivants :

$$\text{Gain directif: } G(q,j) = h D(q,j) \quad (3.4)$$

$$\text{Gain: } G_0 = \max(G(q,j)) \quad (3.5)$$

Le gain d'une antenne correspond au meilleur gain directif dans une (ou plusieurs) direction(s). Les plus grands gains sont obtenus avec des antennes paraboliques (20 à 60 dB).

Attention :

Une antenne est un élément strictement passif qui n'amplifie pas le signal ! Son gain, par définition, représente la concentration de puissance dans une direction privilégiée par rapport à une antenne isotrope sans pertes.

On déduit de (3.2), (3.3) et (3.4) la densité de puissance d'une antenne par rapport à la puissance fournie P_F :

$$p(r,q,j) = G(q,j) \frac{P_F}{4\pi r^2} \quad (3.6)$$

De fait, le gain en puissance d'une antenne est un paramètre plus intéressant que sa directivité; en effet, comme le rendement y est inclus, on peut calculer la densité de puissance en introduisant directement la valeur de la puissance fournie P_F dans (3.6)

3.1.3. Énumération des différents types d'antennes :

On trouve à bord d'avion plusieurs types d'antennes qui sont bien placées pour qu'elles fassent leurs travaux en bonnes conditions et on site parmi eux :

- Antenne cadre.
- Doublet replié.
- Assiette
- Sabre très court.
- Antenne Localizer.
- Antenne GLIDE PATH.
- Antenne horizontale $/2$.
- deux assiettes.
- Antenne omnidirectionnelle.
- Ensemble d'antennes.
- Antenne (radar météorologie).

3.1.4. Émission et réception d'antenne :

3.1.4.1. Émission :

3.1.4.1.1. Définition :

L'antenne est utilisée exclusivement pour émettre.

3.1.4.1.2. Représentation :

Elle est donnée *figure.3.1*.

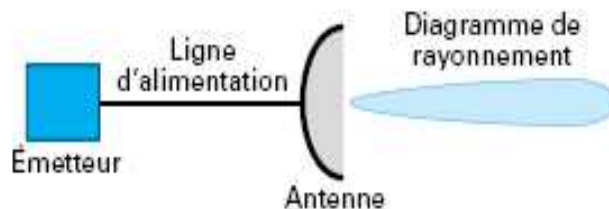


Figure.3.1 – Représentation d'une antenne d'émission seule.

Exemples d'application :

Les antennes d'émission seule sont largement utilisées dans de nombreuses applications :

- Emetteurs de télévision et de radio, terrestres ou satellisés ;
- Balises de navigation aérienne ou maritime ;
- Emetteurs satellisés des systèmes GPS (Global Positioning System) ;
- Communications lointaines (stations terriennes des satellites d'exploration, pour lesquels les antennes d'émission et de réception sont distinctes) ;
- Antennes d'émission des radars multistatiques ;
- Armes hyperfréquence ; etc.

3.1.4.1.3. Caractéristiques principales :

En général, dans les antennes d'émission seule, on se préoccupe d'optimiser le rendement, car la production d'énergie est toujours coûteuse.

- Les facteurs importants sont alors :
- Les pertes de la ligne d'alimentation ;
- L'adaptation de cette ligne à l'antenne ;
- Le rendement de la loi d'illumination (aussi uniforme que possible, s'il s'agit d'une ouverture).

3.1.4.1.4 Difficultés :

Ce sont les suivantes :

- Tenue en puissance des éléments constitutifs de l'antenne et notamment de la ligne d'alimentation
- Zone de sécurité pour les êtres vivants ;
- Brouillage des systèmes électroniques voisins (notamment par les fuites de l'antenne, rayonnement arrière ou autres), ce qui peut entraîner un filtrage drastique des puissances émises ;
recherche d'une loi d'illumination aussi uniforme que possible, qui conduit à des réflecteurs dits « Conformés », lorsque l'antenne utilise des systèmes focalisants.

3.1.4.2. Réception :

3.1.4.2. 1. Définition :

L'antenne est utilisée exclusivement pour recevoir.

3.1.4.2. 2. Représentation :

Elle est donnée *figure.3.2*.

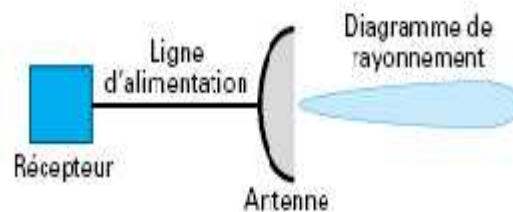


Figure .3.2 – Représentation d'une antenne de réception seule.

3.1.4.2. 3. Exemples d'application :

Les antennes de réception seule sont largement utilisées dans de nombreuses applications :

- Réception de télévision ou de radio (émetteurs au sol ou satellisés) ;
- Réception de signaux émis par les balises de navigation aérienne ou maritime ;
- Réception des signaux émis par les satellites GPS ;
- Réception des signaux horaires ;
- Communications lointaines (stations terriennes des satellites d'exploration, pour lesquels les antennes d'émission et de réception sont distinctes) ;
- Antennes de réception des radars multistatiques ;
- Antennes de radioastronomie ; etc.

3.1.4.2. 4. Caractéristiques principales :

En général, les contraintes des antennes de réception seule sont faibles, car le bilan de liaison est confortable. Un dipôle très petit devant la longueur d'onde et donc à faible gain est souvent suffisant.

On notera cependant les quelques cas particuliers suivants :

Antennes de réception de télévision par satellites : la recherche d'un diamètre aussi petit que possible, pour des raisons d'esthétique et de coût, conduit à placer l'amplificateur juste derrière la source pour supprimer le bruit qui serait apporté par les pertes de la ligne d'alimentation.

Antennes de radar : pour éviter le brouillage et réduire les échos de sol, on est conduit à rechercher des diagrammes dont les niveaux de lobes latéraux proches et lointains sont très faibles, ce qui exige une bonne conception de l'antenne et une réalisation soignée.

Antennes de radioastronomie : ces antennes sont généralement gigantesques pour assurer le bilan de liaison ; la chasse aux degrés de température de bruit devient fondamentale (refroidissement du système de réception, réduction des lobes latéraux visant le sol, masquage du sol par un réseau de conducteurs...).

3.1.4.2. 5. Difficultés :

Il s'agit notamment des problèmes suivants :

- Le pointage, pour les antennes à grand gain : il doit être précis, sous peine de réception nulle ;
- Le filtrage pour les antennes à diagramme omnidirectionnel ; celles-ci sont sensibles à une grande quantité d'émetteurs ;
- La polarisation : elle doit être, bien sûr, celle de l'émetteur ou presque ;
- La nécessité d'une liaison sans masque avec l'émetteur, notamment pour les fréquences élevées.

3.1.4.3. Émission et réception :

3.1.4.3.1. Définition :

L'antenne est utilisée pour émettre et recevoir.

3.1.4.3.2. Représentation :

Elle est donnée *figure.3.3*.

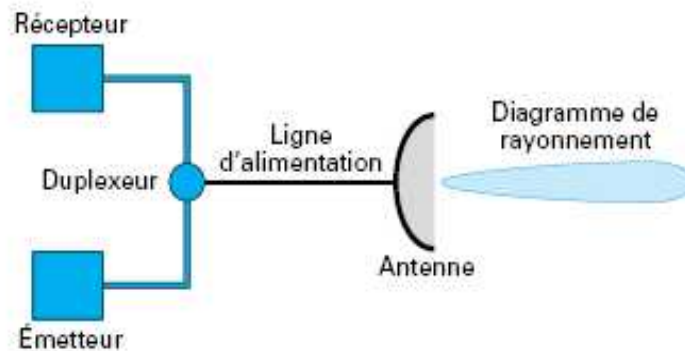


Figure .3.3 – Représentation d'une antenne d'émission et de réception.

Exemples d'application :

Ces antennes sont utilisées notamment pour :

- Les radars : la plupart des radars utilisent la même antenne pour émettre des impulsions, puis recevoir les échos de l'environnement ;
- Les radiotéléphones (un émetteur central fixe et plusieurs petits émetteurs-récepteurs mobiles) ;
- Les « talkies-walkies » (deux, ou plus, petits émetteurs-récepteurs mobiles) ;
- Les liaisons lourdes par satellites (stations au sol fixes communiquant par relais satellisé) ;
- Les liaisons légères par satellites (un mobile au sol, communiquant avec une station fixe au sol, grâce à un relais satellisé) ;
- Les liaisons sol-sol ou air-sol par radiotéléphonie.

3.1.4.3.3. Caractéristiques principales :

Le diagramme de l'antenne est un compromis entre les qualités souhaitables à l'émission (gain) et celles souhaitables à la réception (température de bruit, niveaux des latéraux).

3.1.4.3.4. Difficultés :

La coexistence au même endroit de composants à puissance élevée (émission) et de composants très sensibles (réception) est toujours une difficulté à résoudre.

Le découplage émetteur-récepteur peut atteindre des valeurs considérables (par exemple 150 dB !).

3.1.5. Réseaux linéaire :

Les réseaux linéaires sont utilisés principalement en radar :

- Soit directement pour certaines applications telles que radars secondaires, radars de trafic maritime...
- Soit, plus généralement, comme composant d'une structure plus complexe.

On peut, par exemple, les empiler pour former une antenne dalle, c'est-à-dire un réseau plan.

On peut aussi les utiliser au foyer d'un réflecteur cylindro-parabolique, obtenant ainsi une antenne mixte.

Dans ce cas, l'onde émise par le réseau est cylindrique tant que la distance est beaucoup plus petite que $2d^2/\lambda$, d étant la distance au réseau. Cette onde est transformée en onde plane par le réflecteur cylindrique.

On peut classer les réseaux linéaires en deux types :

- Les réseaux dispersifs, qui ont une direction de rayonnement maximal qui varie avec la longueur d'onde ;
- Les réseaux non dispersifs, c'est-à-dire ceux dont la direction de rayonnement maximal ne dépend pas de la longueur d'onde.



Figure.3.4 – Réseau linéaire de 3 dipôles

3.1.5.1. Réseaux linéaires dispersifs :

3.1.5.1.1. Principe :

Ces réseaux sont aussi appelés réseaux à alimentation série, parce que les différentes sources rayonnantes sont alimentées successivement à partir d’une seule ligne d’alimentation.

On demande généralement à ces réseaux d’avoir les trois propriétés suivantes :

- Absence de lobe de réseau ;
- TOS faible, comme toute antenne qui se respecte... ;
- Direction du rayonnement maximal voisin de la normale au réseau.

La structure générale représentée par la **figure.3.5** permet de satisfaire ces trois exigences.

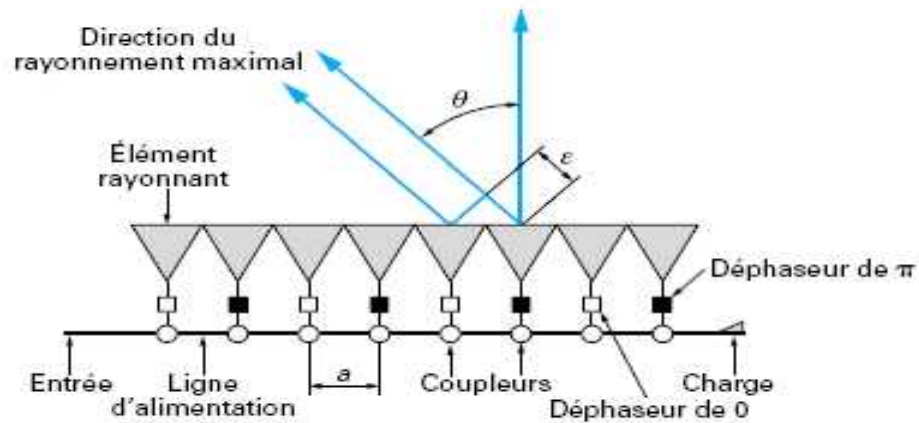


Figure.3.5 – Réseau linéaire dispersif : description.

3.1.5.1.2. Balayage angulaire standard :

Le pas a entre deux sources est très voisin d’une demi-longueur d’onde dans la ligne d’alimentation

$$\begin{cases} a = \frac{g}{2} - x \\ x = g \end{cases} \tag{3.7}$$

Avec g longueur d’onde dans la ligne d’alimentation.

Le déphasage apporté par la ligne entre deux sources est donc voisin de 180° .

La direction de rayonnement maximal est la direction pour laquelle toutes les sources rayonnent en phase, c’est donc, compte tenu du déphaseur de f existant en série avec chaque source, la direction donnée par :

$$2p \frac{a}{g} + 2p \frac{a}{g} \sin q = p$$

Soit encore :

$$\sin q = \frac{a}{g} - \frac{a}{g}$$
(3.8)

Les sources rayonnent en phase puisque le déphasage total est de 2π

Si le pas est exactement égal à une demi-longueur d’onde dans le guide, alors le maximum de rayonnement correspond à la normale au guide, ce qui est souvent le but recherché ;

Toutes les réflexions dues aux éléments rayonnants et aux coupleurs se retrouvent, alors, en phase à l’entrée du guide, provoquant généralement un TOS très grand. Aussi, généralement, on ne cherche pas à rayonner suivant la normale avec ce type d’antenne, mais juste d’un côté ou de l’autre.

Le pas entre deux sources étant voisin de $\lambda/2$, si la bande de fréquence n’est pas trop grande, il n’y a pas de lobes de réseau.

Lorsque la fréquence varie, la direction du maximum varie également. On peut ainsi obtenir un balayage électronique très simplement en faisant varier la fréquence dans une bande donnée.

Le balayage angulaire (en radians) obtenu avec une bande de fréquence est donné par la formule approximative suivante :

$$Dq = -\frac{g}{f} \frac{Df}{f}$$
(3.9)

Avec Df : Excursion de fréquence

f : Fréquence centrale.

3.1.5.1.3. Augmentation du balayage angulaire :

Dans certaines applications, on peut chercher à obtenir un grand balayage angulaire avec la même largeur de bande de fréquence. Alors, il faut utiliser une ligne à méandres telle que illustrée sur la **Figure.3.6**.

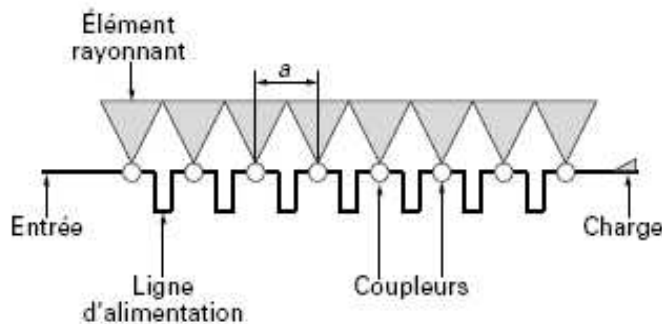


Figure .3.6– Réseau linéaire dispersif avec ligne à méandres.

Le balayage angulaire est alors :

$$Dq = \frac{l}{a} \frac{g}{f} \frac{Df}{f} \quad (3.10)$$

Avec l longueur de ligne entre deux éléments rayonnants.

3.1.6. Antenne à balayage électronique :

3.1.6. 1. Définition :

Une antenne à balayage électronique est généralement une antenne réseau dont les éléments rayonnants sont équipés de déphaseurs électroniques, permettant de réaliser sur l'ouverture, très rapidement, des lois de phase linéaires et donc de pointer le faisceau dans des directions différentes.

3.1.6. 2. Représentation :

Elle est donnée *figure.3.7*.

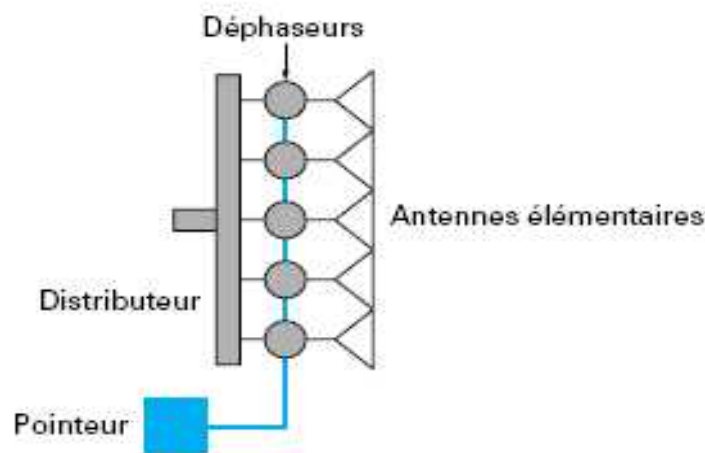


Figure .3.7 – Représentation d'une antenne à balayage électronique.

3.1.6. 3. Exemples d'application :

- Les antennes de radars multifonctions, c'est-à-dire des radars devant assurer la veille et la poursuite de cibles, sont des antennes à balayage électronique.
- Certaines antennes de satellites de télécommunication et, notamment, celles qui équipent les avions commerciaux sont à balayage électronique.
- Dans ce type d'antenne, les déphaseurs sont quantifiés, que cette quantification est source de problèmes (perte de gain, élévation du niveau des lobes latéraux...).

3.1.7. Etude théorique de rangée d'antennes linéaire :

3.1.7. 1. Introduction :

3.1.7. 1. 1. Généralités :

Les rangées d'antennes sont utilisées pour diriger la puissance rayonnée vers un secteur angulaire désiré.

Le nombre, l'arrangement géométrique, et les amplitudes et les phases relatives de la rangée les éléments dépendent du modèle angulaire qui doit être réalisé.

Une fois une rangée a été conçue pour se focaliser vers une direction particulière, il devient facile de l'orienter vers une autre direction en changeant les phases de parent des éléments de rangée, le processus a appelé la direction ou le balayage (steering or scanning).

La **figure.3.8** montre quelques exemples de rangées des antennes linéaires identiques.

Un élément linéaire d'antenne placé à la direction z , a un modèle omnidirectionnel en ce qui concerne l'angle azimutal j . Par la réplique l'élément d'antenne placé à la direction x ou y , la symétrie azimutale est cassé.

Par le choix approprié des coefficients d'alimentation de rangée a_n , n'importe quel modèle désiré de gain $G(j)$ peut être synthétisé.

Si l'élément d'antenne est replié à direction z , puis l'omnidirectionnalité en ce qui concerne j est maintenu avec d'éléments de rangée, tout angle polaire prescrit le modèle $G(j)$ peut être conçu.

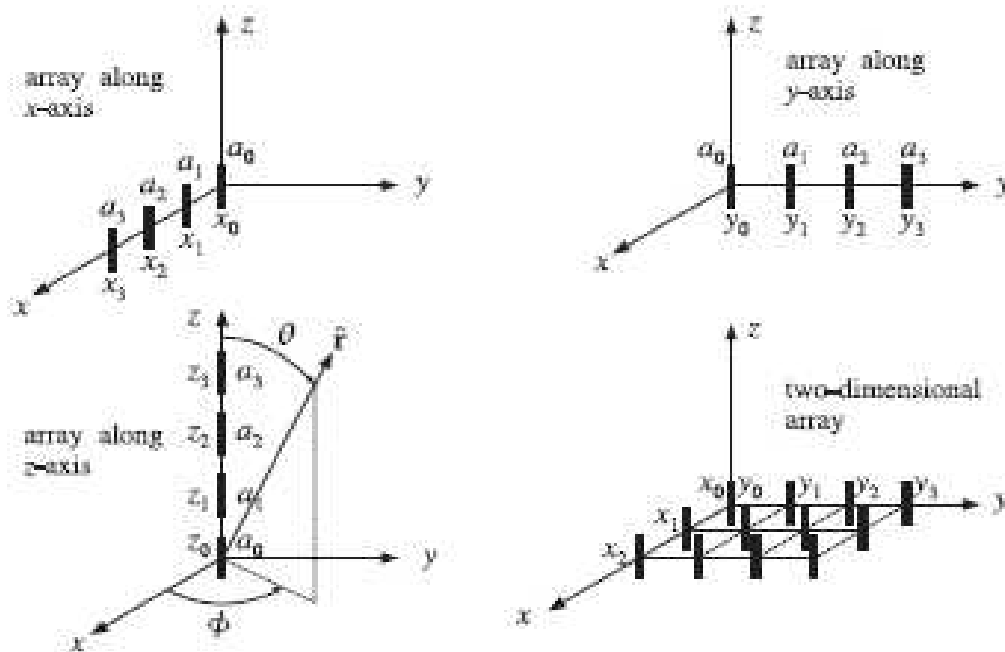


Figure .3.8– Configurations typiques de rangée.

3.1.7. 1. 2. Déphasage De translation :

La propriété la plus fondamentale d'une rangée est que les déplacements relatifs des éléments d'antenne en ce qui concerne l'un l'autre présenter déphasage relatif dans les vecteurs de rayonnement, ce qui peut alors s'ajouter de manière constructive dans quelques directions ou nuisiblement dans d'autres. C'est une conséquence directe de la propriété de translation de déphasage [16] de Fourier transforme : a la traduction dans l'espace ou le temps devient déphasage dans le domaine de Fourier.

La **figure.3.9** montre du côté gauche une antenne traduite par le vecteur d, et sur redresser, plusieurs antennes traduites à différents endroits et alimentées avec différentes amplitudes relatives.

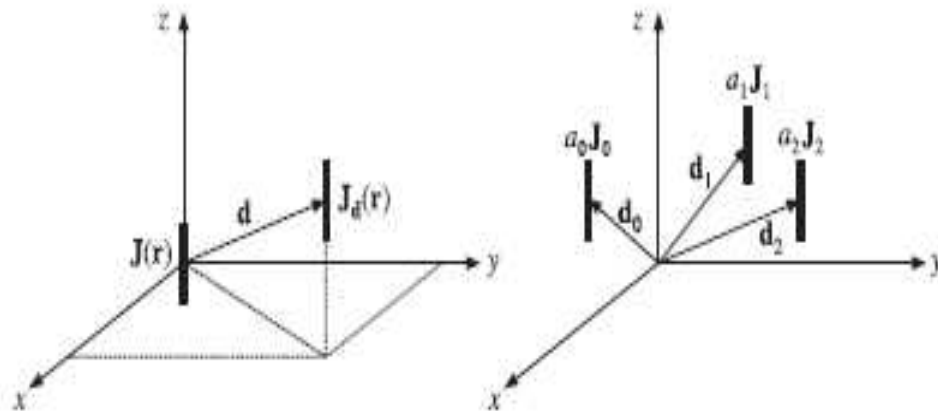


Figure .3.9– Antennes traduites.

La densité de courant de l'antenne traduite sera :

$$J_d(r) = J(r - d)$$

Par définition, le vecteur de rayonnement est la transformée de Fourier tridimensionnelle de la densité de courant.

Ainsi, le vecteur de rayonnement du courant traduit sera :

$$\begin{aligned} F_d &= \int \delta e^{jkr} J_d(r) d^3r = \int \delta e^{jkr} J_d(r - d) d^3r = \int \delta e^{jk(r'+d)} J_d(r') d^3r' \\ &= \int \delta e^{jkr'} J_d(r') d^3r' = e^{jkd} F \end{aligned} \tag{3.11}$$

Où nous avons changé des variables $r' = r - d$

Ainsi :

$$F_d(k) = e^{jkd} F(k) \text{ (Déphasage De translation)} \tag{3.12}$$

3.1.7. 2. Rangées unidimensionnelles :

3.1.7. 2.1. Facteur de rangée :

Après, nous considérons des rangées unidimensionnelles uniforme espacées. Une rangée placée sur l'axe x (voir la **figure.3.10**) avec placés aux endroits x_n , $n = 1, 2, \dots$

Le facteur de rangée est :

$$A(q, j) = \sum_n a_n \exp(jk d_n) = \sum_n a_n \exp(jk_x x_n) = \sum_n a_n \exp(jk x_n \sin\theta \cos\phi) \quad (3.13)$$

Avec

$$d_n = x_n \hat{u}_x$$

a_n : Les poids des éléments

Pour des rangées équidistantes, les endroits d'élément sont $x_n = nd$, où d est la distance entre les éléments.

Dans ce cas-ci, le facteur de rangée devient :

$$A(q, j) = \sum_n a_n \exp(jk n d \sin\theta \cos\phi) \quad (3.14)$$

Puisque la dépendance angulaire $k_x d = k d \sin\theta \cos\phi$, nous sommes menée à définir la variable :

$$y = k_x d = k d \sin\theta \cos\phi$$

Puis, le facteur de rangée peut être considéré en fonction de y :

$$A(y) = \sum_n a_n \exp(jy n) \quad (3.15)$$

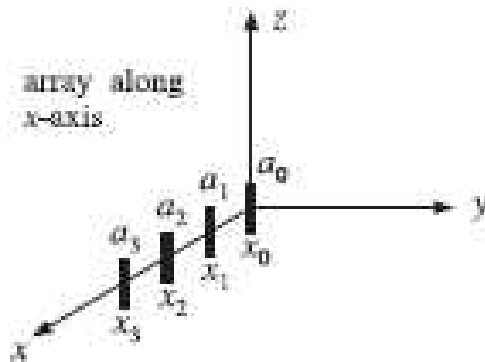


Figure.3.10– Rangée unidimensionnelle.

La variable y est une version normale du nombre d'onde k_x et mesurée en unités des radians par échantillon (de l'espace). Ce peut s'appeler un nombre d'onde numérique normal, la réponse fréquentielle de filtre digital du facteur de rangée est :

$$A(w) = \sum_n a_n \exp(-j w n) \quad (3.16)$$

Le nombre d'onde est défini pareillement pour des rangées placée sur les directions y et z, nous avons les définitions :

$$\begin{aligned} y = k_x d &= k d \sin q \cos j && \text{direction } x \\ y = k_y d &= k d \sin q \sin j && \text{direction } y \\ y = k_z d &= k d \cos q && \text{direction } z \end{aligned} \quad (3.17)$$

Les facteurs de rangées placée sur les direction y et z (les rangées d'axe montrées dans *figure16*) seront :

$$\begin{aligned} A(q, j) &= \sum_n a_n \exp(j k_y y_n) = \sum_n a_n \exp(j k y_n \sin q \sin j) \\ A(q, j) &= \sum_n a_n \exp(j k_z z_n) = \sum_n a_n \exp(j k z_n \cos j) \end{aligned} \quad (3.18)$$

3.1.7. 2.2. La transformée en Z :

La transformé en z peut être considéré comme une généralisation de la transformation de Fourier à laquelle elle peut s'identifier dans un cas particulier.

La transformée en z constitue l'outil pour l'étude des systèmes discrets. Elle joue un rôle équivalent à celui de la transformée de Laplace.

Par exemple, la transformée en z permet de représenter un signal possédant un infinité d'échantillons par un ensemble fini de nombre.

Application :

En définissant la variable :

$$z = \exp(j \omega)$$

La transformée en Z de $A(y)$ est :

$$A(z) = \sum_n a_n z^n \quad (\text{Facteur de rangée dans spatial } z - \text{domaine}) \quad (3.19)$$

La différence dans le signe entre l'espace domaine et les définitions de temps domaine est également évident ici, d'où l'expansion est dans les puissances z^n au lieu de z^{-n} .

Le facteur de rangée $A(y)$ peut s'appeler la transformée de Fourier de l' discret-espace (DSFT) de vecteur de poids a_n , juste comme la transformée de Fourier de temps discret (DTFT) du cas de temps domaine.

Le DSFT inverse correspondant est obtenu près :

$$a_n = \int_{-p}^p A(y) \exp(-j y n) dy \quad (\text{DSFT inverse}) \tag{3.20}$$

Cet inverse transformation des formes la base de la plupart des méthodes de conception pour les coefficients de rangée.

3.1.7. 3. Rangées unidimensionnelles uniformes :

La rangée unidimensionnelle la plus simple est la rangée uniforme ayant les poids égaux.

Pour choix N d'éléments isotropes aux endroits $x_n = nd, n=0, 1, 2, \dots, N-1$

Nous définissons :

$$a = [a_0, a_1, \dots, a_{N-1}] = \frac{1}{N} [1, 1, \dots, 1]$$

De sorte que la somme des poids soit unité.

Le polynôme de rangée et la rangée correspondants le facteur sont :

$$A(z) = \frac{1}{N} (1 + z + z^2 + \dots + z^{N-1}) = \frac{1}{N} \frac{z^N - 1}{z - 1}$$

$$A(y) = \frac{1}{N} (1 + \exp(jy) + \exp(2 jy) + \dots + \exp((N-1) jy)) = \frac{1}{N} \frac{\exp((N-1) jy) - 1}{\exp(jy) - 1} \tag{3.21}$$

Où $z = \exp(jy)$ et $y = kd \cos j$ pour une rangée placée sur l'axe x et direction de regard sur le plan xy. Nous pouvons également écrire $A(y)$ sous la forme :

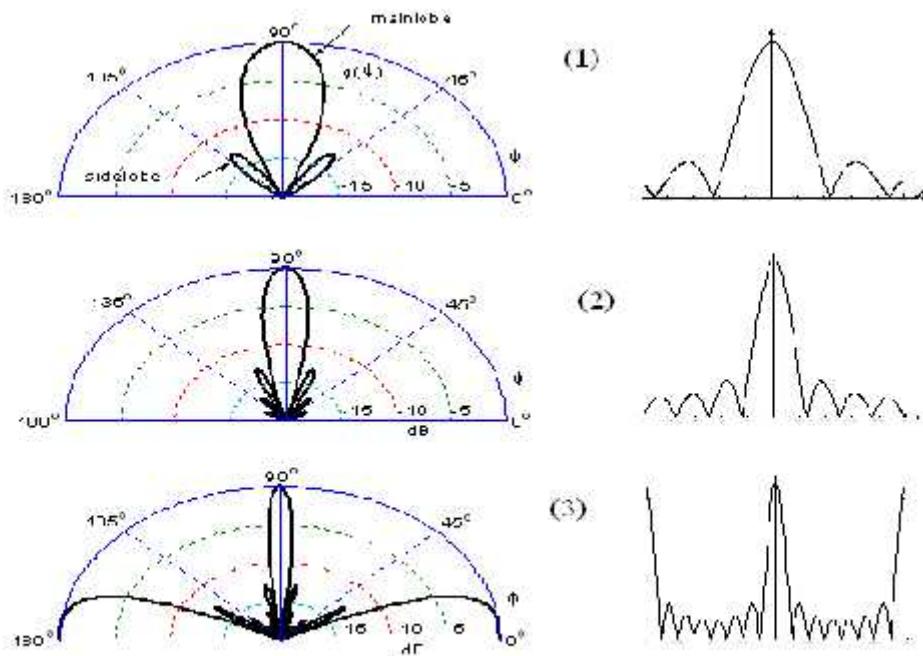
$$A(y) = \frac{\sin\left(\frac{N y}{2}\right)}{N \sin\left(\frac{y}{2}\right)} \exp(j(N-1)y/2) \quad (\text{rangée uniforme}) \tag{3.21}$$

Le gain de puissance de la rangée sera :

$$g(j) = |A(y)|^2 = \left| \frac{\sin\left(\frac{N y}{2}\right)}{N \sin\left(\frac{y}{2}\right)} \right|^2 = \left| \frac{\sin((N kd/2) \cos j)}{N \sin((kd/2) \cos j)} \right|^2 \tag{3.22}$$

La **figure.3.11** représente facteur et gain de puissance de rangée de 8 éléments (N=8), l'espacement entre les éléments

- (1)..... $d = 0.25$, $y = (y/2) \cos j$ $\mathbb{P} - p/2 \leq y \leq p/2$
- (2)..... $d = 0.5$, $y = p \cos j$ $\mathbb{P} - p \leq y \leq p$
- (3)..... $d = p$, $y = 2p \cos j$ $\mathbb{P} - 2p \leq y \leq 2p$



Gain de puissance

Facteur de rangée

Figure.3.11- Facteur de rangée et modèle angulaire de la rangée 8-element uniforme.

3.1.7.4. Direction De Rangée :

Une rangée est typiquement conçue pour avoir le gain directif maximum c à d = 90° , le facteur de rangée maximum correspondant $y = k d \cos j = 0$ donc $|A|_{MAX} = |A(0)|$

Nous souhaitons "électroniquement" tourner, ou orienter le modèle de rangée vers une autre direction j_0 , sans la tourner physiquement. Le nombre d'onde correspondant à la direction désirée sera :

$$y_0 = k d \cos j_0$$

Une telle opération d'orientation peut être réalisée par l'espace de traduction de nombre d'onde , en remplaçant le modèle de lobe $A(y)$ par le modèle traduit $A(y - y_0)$.

Ainsi nous définissons :

$$A'(y) = A(y - y_0)$$

Et le nombre d'onde traduite :

$$y' = y - y_0 = k d \cos(j - j_0)$$

Puis :

$$A'(y) = A(y')$$

Le maximum de $A'(y)$ coïncide avec le maximum de $A(y')$, qui se produit $y' = 0$ c à d $y = y_0$, ou l'angle $\theta = \theta_0$

La figure.3.12 illustre ce procédé de traduction de nombre d'onde et la rotation correspondante du modèle angulaire, pour une rangée 11-element uniforme avec $d = 0.5$, orientation du lobe principale à $\theta_0 = 60^\circ$.

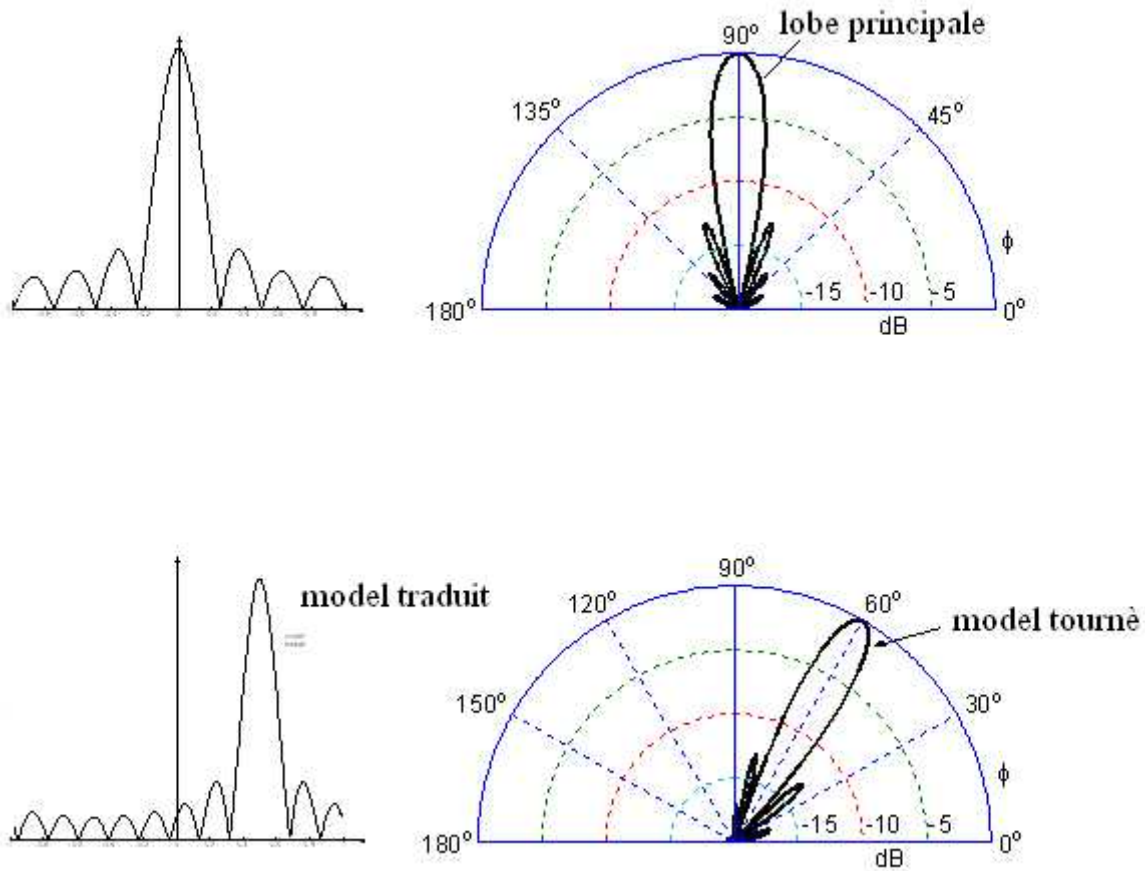


Figure.3.12- Balayage par traduction.

D'après la théorème de traduction du transformation Fourier, les coefficients de poids a'_n du modèle traduit $A(y')$ sera donné par :

$$a'_n = a_n \exp(-j\theta_0 n)$$

En raison des facteurs de phase progressifs $\exp(-j\theta_0 n)$ dans les poids a'_n , orientée la rangée balayée s'appelle parfois une rangée de balayage.

3.1.7.5. La comparaison entre les traitements matriciels des rangées et le traitement de signal numérique :

La comparaison entre les traitements matriciels des rangées et le traitement de signal numérique dans le domaine du temps sont montrés dans le **tableau.3.1**.

discrete-time signal processing	discrete-space array processing
time-domain sampling $t_n = nT$	space-domain sampling $x_n = nd$
sampling time interval T	sampling space interval d
sampling rate $1/T$ [samples/sec]	sampling rate $1/d$ [samples/meter]
frequency	wavenumber k_x
digital frequency $\omega = \omega_c T$	digital wavenumber $\omega = k_x d$
Nyquist interval $-\pi$	Nyquist interval $-\pi$
sampling theorem $\omega_c \leq \pi/T$	sampling theorem $k_x \leq \pi/d$
spectral images	grating lobes or fringes
frequency response $A(\omega)$	array factor $A(\theta)$
z-domain $z = \exp(j\omega T)$	z-domain $z = \exp(jy)$
transfer function $A(z)$	transfer function $A(z)$
DTFT and inverse DTFT	DSFT and inverse DSFT
pure sinusoid $\exp(j\omega_0 n)$	narrow beam $\exp(jy_0 n)$
windowed sinusoid $w(n)\exp(j\omega_0 n)$	windowed narrow beam $w(n)\exp(jy_0 n)$
resolution of multiple sinusoids	resolution of multiple beams
frequency shifting by AM modulation	phased array scanning
filter design by window method	array design by window method
bandpass FIR filter design	angular sector array design
frequency-sampling design	Woodward-Lawson design
DFT Blass	matrix
FFT Butler	matrix

Tableau.3.1. La comparaison entre les traitements matriciels des rangées et le traitement de signal numérique.

3.2. Les antennes intelligentes :

3.2.1. Définition des antennes intelligentes :

Les systèmes à antennes adaptatives ou intelligentes (*smart antennas*) [6] entrent dans un domaine technologique multidisciplinaire dont la croissance a été très importante dans le courant de ces dernières décennies. Ils bénéficient des interactions croissantes entre l'électromagnétisme et le traitement de signal. L'intérêt de ces systèmes est leur capacité à réagir automatiquement, en temps réel, à des modifications du canal de propagation. Ils permettent de réduire les niveaux des lobes secondaires existants dans la direction de l'interférence, tout en maintenant le lobe principal en direction utile [7]. Habituellement, ces systèmes reposent sur des antennes réseau et sur un processeur récepteur adaptatif temps réel qui attribue des poids aux éléments de l'antenne réseau afin d'optimiser le signal de sortie selon des algorithmes de contrôle prédéfinis.

Une antenne réseau adaptative peut donc être définie comme un réseau capable de modifier son diagramme de rayonnement, sa réponse fréquentielle et d'autres paramètres grâce à une boucle à retour de décision interne pendant le fonctionnement de l'antenne.

Dans les systèmes d'antennes traditionnels, chaque émetteur envoie son signal de façon très large principalement à cause du fait que la position du récepteur est inconnue [8]. Ce type de transmission pollue l'environnement électromagnétique en augmentant le niveau d'interférence global par une émission de puissance dans des directions inutiles. A contrario, les systèmes à antennes intelligentes déterminent la localisation du mobile pour se focaliser et émettre l'énergie uniquement dans les directions souhaitées.

Les antennes intelligentes emploient un ensemble d'éléments d'antenne dans une rangée et les signaux reçus de chaque élément sont combinés de telle manière que le modèle de faisceau puisse être orienté par des techniques de traitement de signal numérique à une direction désirée afin de dépister la cible. Les systèmes d'antenne intelligente peuvent être classifiés :

1. Antennes commutées de faisceau
2. Rangées échelonnées dynamiques
3. Rangées d'antenne adaptatives

3.2.2. La Conception d'antenne intelligente :

Les antennes intelligentes ont été conceptualisées la première fois dans les années 60 avec la naissance du signal numérique.

Ils comportent des systèmes d'antenne qui peuvent changer son antenne rayonnez le modèle afin de recevoir l'utilisateur désiré et dans certains cas, rejetez simultanément les signaux des interférés.

Ils fournissent beaucoup de systèmes d'antenne fixés par excédent d'avantages qui peut adapter au désiré les conditions d'assurance mais ne peuvent pas réagir dynamiquement aux changements du trafic dans une cellule.

Les divers concepts futés d'antenne tels que beamforming, leur complexité informatique et la motivation pour la recherche faite dans cette thèse sont élucidés en ce chapitre.

3.2.3. Avantages des antennes intelligentes :

La caractéristique principale des antennes intelligentes est le pouvoir d'élimination des interférences pour les applications radiocommunication ce qui donne un meilleur rapport signal sur bruit, donc une augmentation de la capacité en nombre d'utilisateurs.

3.2.4. Rangée D'Antenne Adaptative :

Une rangée d'antenne adaptative est un choix d'éléments d'antenne qui peut changer le son modèle d'antenne dynamiquement à ajuster sur le bruit, l'interférence et par trajets multiples (*figure.3.13*).

La figure montre la rangée d'antenne adaptative qui couvre un des secteurs de a cellule formant un faisceau dans la direction du signal désiré et le plaçant simultanément annule dans la direction de deux signaux d'intervention.

Des rangées d'antenne adaptatives peuvent être employées améliorer la réception du signal désiré et peut également être employée pour former des faisceaux pour transmission. Dans une rangée d'antenne adaptative, les poids sont ajustés pour maximiser SINR et fournissent la discrimination maximum contre les signaux d' intervention.

Dans absence des interférés et du bruit étant la seule entité peu désirée, antennes adaptatives comportez-vous comme combinateur maximal de rapport en maximisant le rapport de signal bruit (SNR).

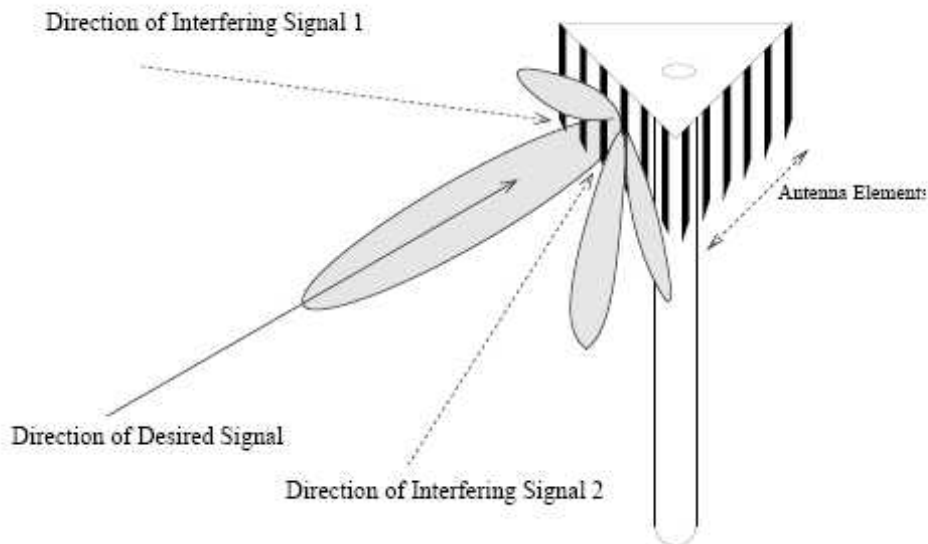


Figure.3.13- Modèle de faisceau d'exemple d'une rangée d'antenne adaptative.

Une variété d'algorithmes de traitement des signaux (algorithmes beamforming) peut être employée pour distinguer les signaux désirés et d'interventions. Dans certains cas, les angles de l'arrivée des signaux sont calculés et leur connaissance est employée à dynamiquement réduisez au minimum l'interférence et maximisez la réception prévue de signal. Section 3.2.7 décrit les divers algorithmes beamforming qui peuvent être employés pour mettre en application fonctions de traitement des signaux des rangées d'antenne adaptatives.

3.2.5. Structure d'une Rangée D'Antenne Adaptative :

La structure d'une antenne intelligente [9.10] est illustrée par la (Figure.3.14). Elle est constituée d'un réseau de M antennes élémentaires dont les sorties sont pondérées par un terme complexe w avant d'être sommées entre elles. L'unité de contrôle des pondérations permet de mettre en forme un diagramme de rayonnement par l'ajustement "intelligent" de l'amplitude et (ou) de la phase avec laquelle se combinent les signaux reçus (ou émis) sur les différents éléments.

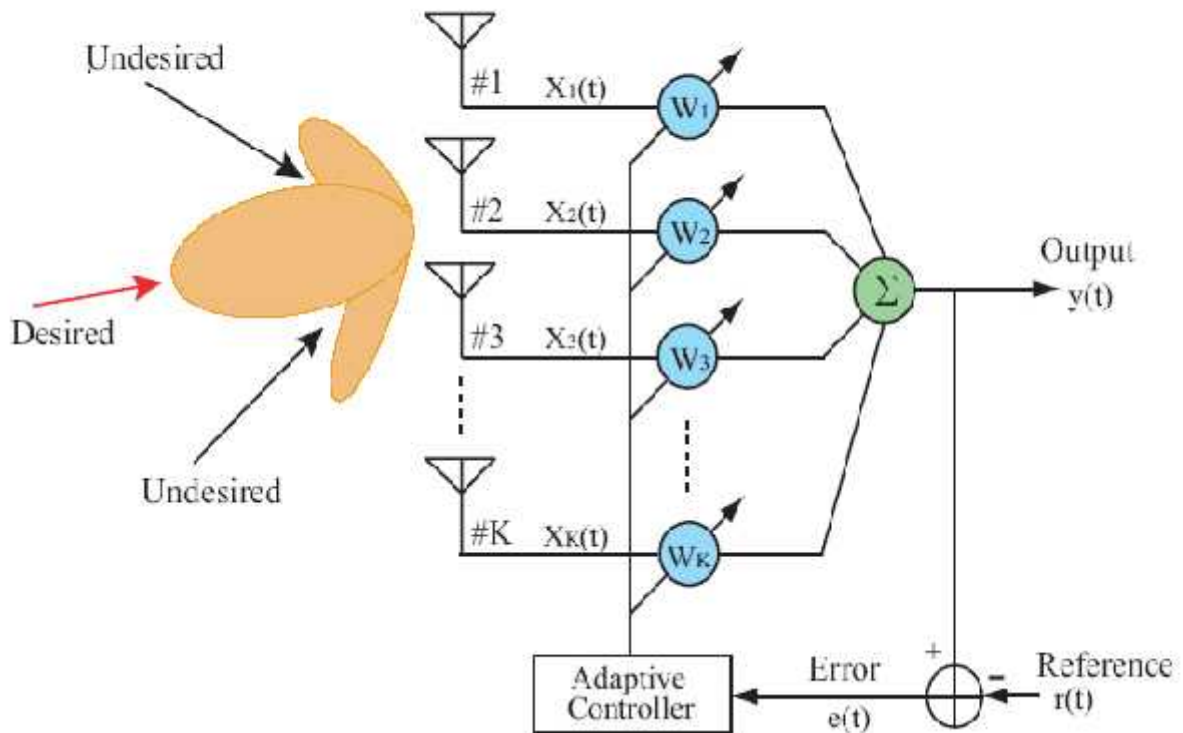


Figure.3.14- Schéma représentatif d'une antenne adaptative.

3.2.6. Les algorithmes adaptatifs :

Pour obtenir les pondérations optimales, diverses méthodes sont possibles. Le choix de l'algorithme qui permet d'atteindre la solution optimale est une étape cruciale car de lui dépendent la vitesse de convergence et la complexité d'intégration matérielle. Les algorithmes utilisés sont :

- LMS: algorithme du gradient stochastique.
- Inversion directe de la matrice de convergence (DMI)
- Algorithme des moindres carrés récursifs (RLS, Récursive least squares algorithm)
- Algorithme à module constant (CMA- Constant modulus algorithm)

3.3. Formation de voies optimales (statistically optimum beamforming) :

3.3.1. Application de filtre adaptif au rangée d'antenne linéaire et annulation des interférences :

Une rangée d'antenne se compose d'un choix de bas éléments d'antenne de gain avec un ensemble de poids qui sont reliés par un réseau de combinaison. Les prétentions suivantes et des notations sont employées pour décrire l'opération d'une rangée d'antenne :

Les signaux à bande étroite sont assumés pour arriver aux éléments de la rangée d'antenne sous forme d'ondes planes (ANNEXE 6). Avec le centre de la rangée comme élément d'antenne de référence, les signaux à chacun des éléments d'antenne sont donnés par le signal détecté au centre décalé dans la phase. Ainsi le vecteur de direction est donné par déphasage respectifs éprouvés à chacun de éléments d'antenne dans la rangée.

Le modèle à bande étroite de la rangée d'antenne est valide parce que le temps qu'elle prend pour le front des ondes passer par la rangée est beaucoup plus petit que l'intervalle de morceau (T_c) [11] et donc le vecteur de direction comporte de déphasage à chacun des éléments d'antenne.

L'espacement des éléments est assez petit qu'il n'y a aucune variation d'amplitude entre les signaux reçus aux différents éléments.

L'espacement est également tel qu'il y a peu si n'importe quel accouplement mutuel entre les éléments. Ceci implique que l'espacement préféré d'antenne entre les éléments est $d = \lambda/2$ (ANNEXE 7), où λ est la longueur d'onde.

En Considérant une onde plane d'un signal à bande étroite, empiétant sur une rangée linéaire de un angle (θ) relativement à l'axe de la rangée.

Laisser l'onde plane modulée être représenté en utilisant l'enveloppe complexe de bande de base $s(t)$.

Laisser la séparation entre deux éléments adjacents d'antenne soit dénotés d , et M les éléments d'antenne est placé comme indiqué dans **figure3.15**.

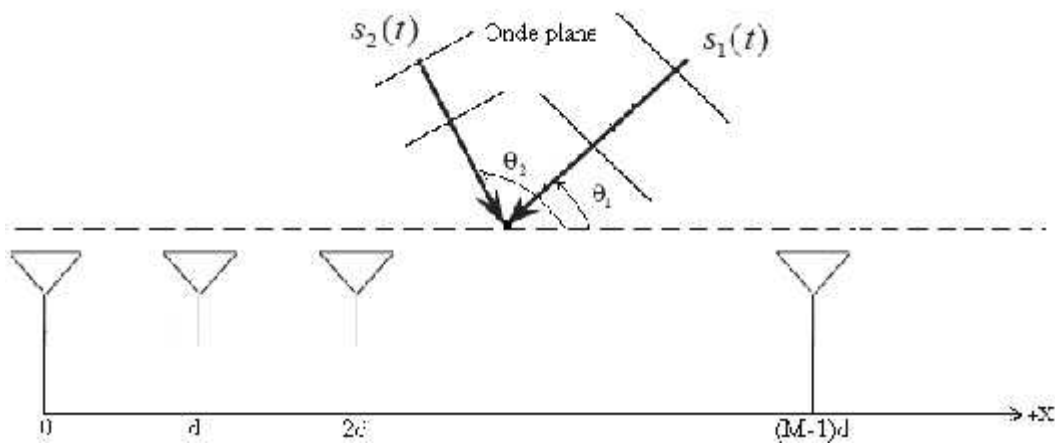


Figure.3.15- Rangée d'antennes linéaire reçoit plusieurs ondes planes.

θ_k L'angle d'arrivée (entre l'axe de la rangée et le normal de l'onde plane)

$k = 0, 1, \dots, p$

p Le nombre total des ondes planes

Maintenant, on considérant une seule onde plane (*Figure.3.16*).

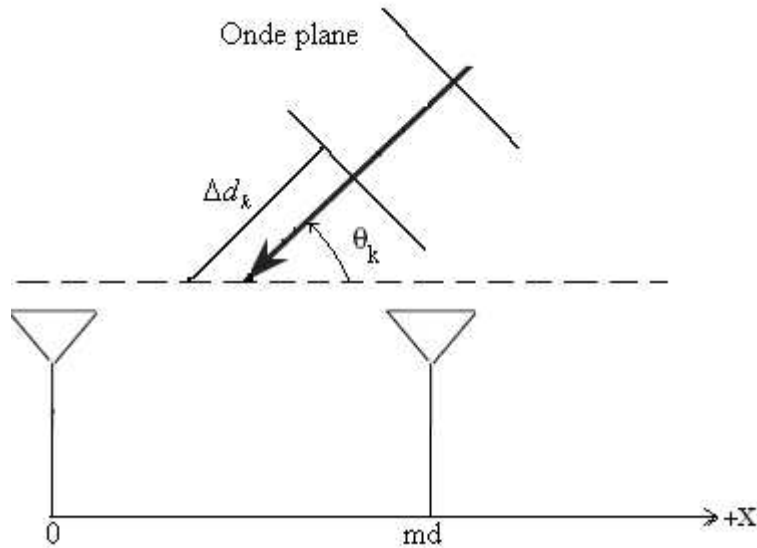


Figure.3.16-Rangée d'antennes linéaire reçoit une seule onde plane.

Notation :

$m = 0, 1, \dots, (M - 1)$

M Nombre d'éléments de la rangée

Le signal transmis par la k^{eme} source est :

$$S_k(t) = S_{I_k}(t) \cos(2\pi f t) - j S_{Q_k}(t) \sin(2\pi f t)$$

Avec :

$S_{I_k}(t)$ La composante réelle

$S_{Q_k}(t)$ La composante imaginaire

$S_{I_k}(f) = 0$ Pour $|f| > \tilde{S}$

$S_{Q_k}(f) = 0$ Pour $|f| > \tilde{S}$

Les composantes $S_{I_k}(t)$ et $S_{Q_k}(t)$ est récupéré grasse (*Figure.3.17*) :

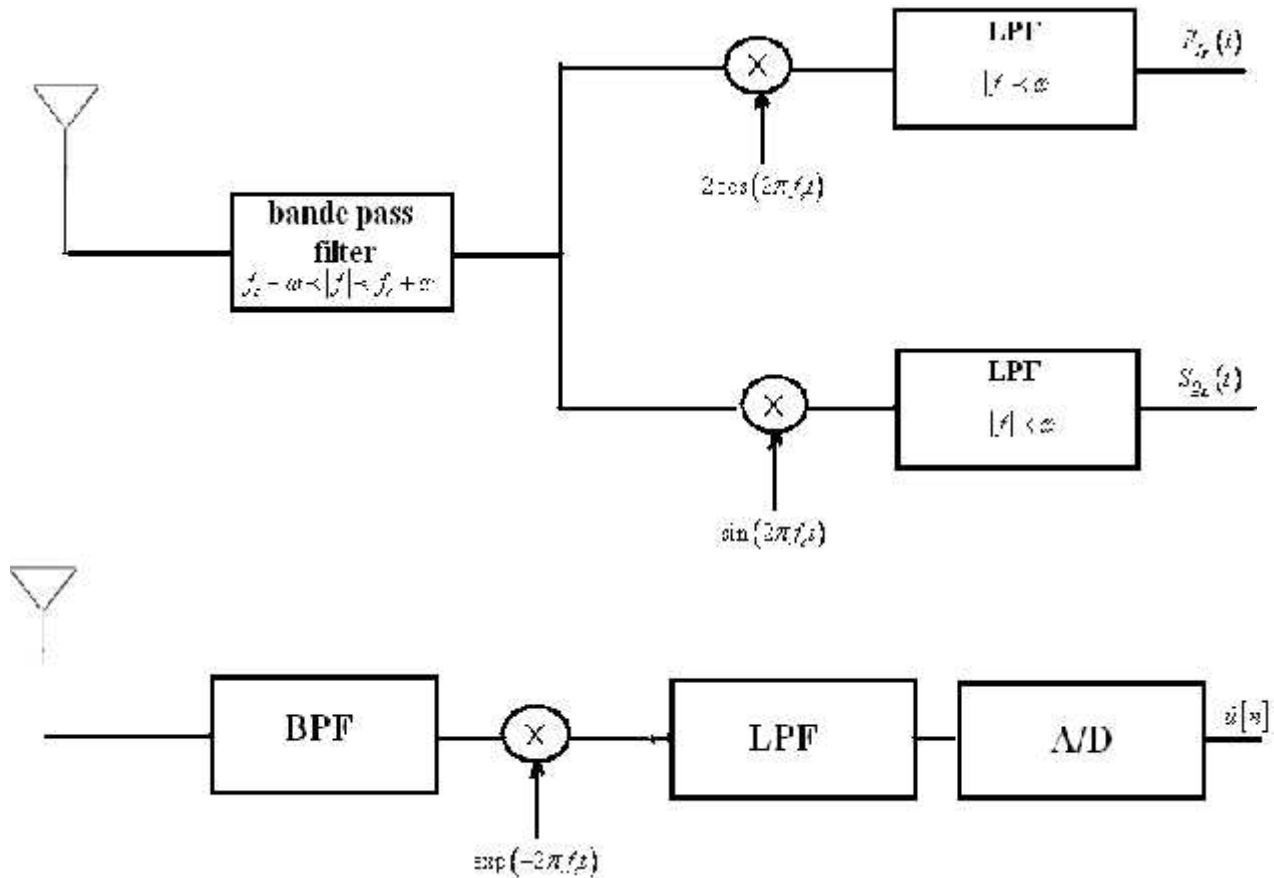


Figure.3.17- schéma synoptique de récupération de l'enveloppe complexe.

Représentation de l'enveloppe complexe :

L'enveloppe complexe définie par :

$$\tilde{S}_k(t) = S_{I_k}(t) + j S_{Q_k}(t)$$

En prend :

$$S_k(t) = \text{Re}\{\tilde{S}_k(t) \exp(2f ft)\}$$

Etude spatiale :

D'après la figure .3.16

Considérant :

La première antenne $x = 0$ et la dernière antenne $x = md$

Le temps de propagation entre eux \dagger_{m_k} nommé aussi le retard

$$\dagger_{m_k} = \frac{md \cos(\theta_k)}{c}$$

Tel que :

$$\Delta d_k = md \cos \theta_k$$

$$\cos \theta_k = \frac{\Delta d_k}{md}$$

Le signal de sortie du m^{eme} élément de la rangée $\tilde{u}(m, t)$ est donné par :

$$\begin{aligned} \tilde{u}(m, t) &= x\left(0, t + \dagger m_k\right) \\ &= \tilde{S}_k\left(t + \dagger m_k\right) \exp\left[j2f f\left(t + \dagger m_k\right)\right] \\ &= \tilde{S}_k\left(t + \dagger m_k\right) \exp\left(j2f f \dagger m_k\right) \exp(j2f ft) \\ &= \tilde{S}_k\left(t + \frac{md \cos(\theta_k)}{c}\right) \exp\left(j \frac{2f fmd \cos(\theta_k)}{c}\right) \exp(j2f ft) \end{aligned}$$

En note :

$$c = \lambda f$$

Et défini :

$$\begin{aligned} u(m, t) &= \tilde{u}(m, t) \exp(-j2f ft) \\ &= \tilde{S}_k\left(t + \frac{md \cos(\theta_k)}{c}\right) \exp\left(j \frac{2f md \cos(\theta_k)}{\lambda}\right) \end{aligned}$$

En peut prendre l'approximation suivante :

$$\tilde{S}_k\left(t + \frac{md \cos(\theta_k)}{c}\right) = \tilde{S}(t) \quad \text{(ANNEXE 8)} \tag{3.23}$$

En définit la fréquence spatiale :

$$\tilde{\theta}_k = \frac{2f}{\lambda} \cos \theta_k$$

Donc :

$$u(m, t) = \tilde{S}_k(t) \exp(j \tilde{\theta}_k m)$$

Pour p sources, nous appliquons théorème de superposition :

$$u(m, t) = \sum_{k=1}^p \tilde{S}_k(t) \exp(j \tilde{\theta}_k m)$$

Avec :

$$m = 0, 1, \dots, M$$

$$k = 0, 1, \dots, p$$

Par définition le vecteur spatial instantané en temps n :

$$\begin{aligned} \underline{u}[n] &= [u[0, n], u[1, n], \dots, u[M-1, n]] \\ &= \sum_{k=1}^p \underline{S}_k[n] \underline{S}_k[\tilde{\theta}_k] + \underline{u}[n] \end{aligned}$$

Ou :

$$\underline{S}_k [\tilde{\tau}_k] = [u[0, n], u[1, n], \dots, u[M-1, n]]$$

La réponse du filtre spatial effectué :

$$y[n] = \sum_{m=0}^{M-1} h^* [m, n] u[m, n]$$

En vecteur :

$$y[n] = \underline{h}_M^H [n] \underline{u}[n]$$

Avec :

$$\underline{u}[n] = [u[0, n], u[1, n], \dots, u[M-1, n]]^T$$

$$\underline{h}_M [n] = [h[0, n], h[1, n], \dots, h[M-1, n]]^T$$

Le signal reçue est bruité ; donc:

$$\begin{aligned} y[n] &= \underline{h}_M^H [n] \underline{u}[n] \\ &= \sum_{k=1}^p S_k [n] (\underline{h}_M^H \underline{S}_k [\tilde{\tau}_k]) + \underline{h}_M^H \underline{u}[n] \end{aligned}$$

En considère le signal reçue correspondant la source $k = 1$, alors :

$$\begin{aligned} y[n] &= S_1 [n] (\underline{h}_M^H \underline{S}_1 [\tilde{\tau}_1]) \\ &+ \sum_{k=2}^p S_k [n] (\underline{h}_M^H \underline{S}_k [\tilde{\tau}_k]) \longrightarrow \text{les interferences} \\ &+ \underline{h}_M^H \underline{u}[n] \longrightarrow \text{le bruit} \end{aligned}$$

3.3.2. Beamforming Adaptif :

Les divers algorithmes Beamforming qui peuvent être classifiés dans les catégories suivantes :

- Beamforming Conventionnel
- Algorithmes temporeles de référence
- Algorithmes Multi Target
- Algorithmes Blind Beamforming.

3.3.2.1. Beamforming Conventionnel :

Le beamformer conventionnel utilise l'information d'AoA pour diriger le faisceau vers l'utilisateur désiré. Les poids de beamformer pour la rangée d'antenne sont indiqués par le conjugué du vecteur de direction du signal de la cible désiré. Les méthodes de sous-espace utilisées comme Multiple Signal Classification (MUSIC) ou Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT).ces méthodes peuvent être utilisées pour déterminer l'AoA du signal d'intérêt.

3.3.2.2. Algorithmes Temporels De Référence:

Les algorithmes temporels de référence se servent du peu d'ordre de formation pour la convergence des poids du Beamforming.

Certains des algorithmes populaires pour les quels peuvent être utilisé Beamforming l'inversion simple de matrice (SMI), moindres à angle droit moyens (LMS) et Algorithmes des moindres carrés récursifs (RLS).

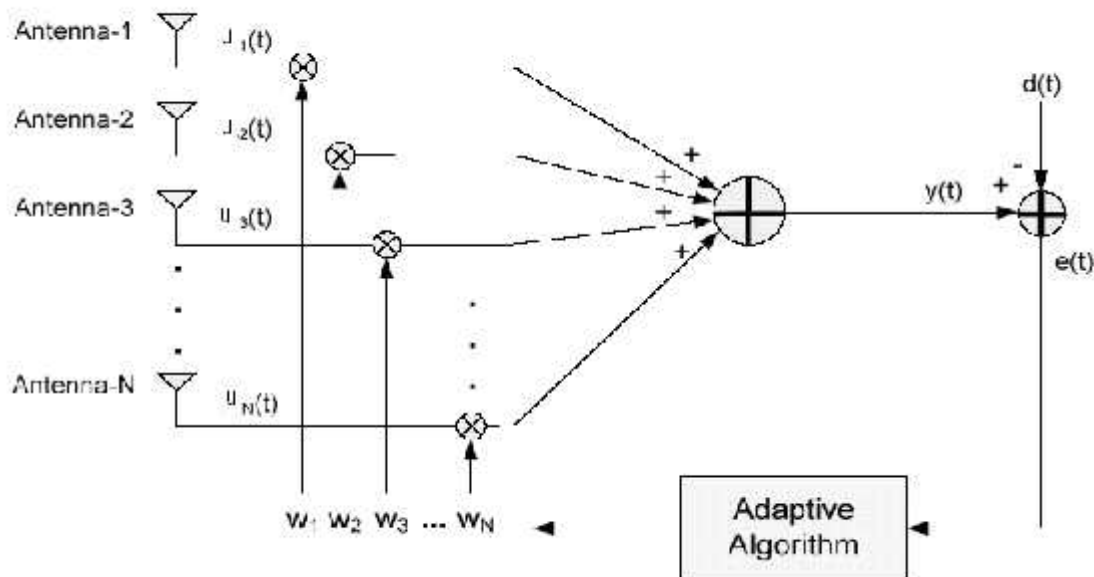


Figure.3.18- Schéma fonctionnel De Beamforming Adaptatif.

3.3.2.2.1. Beamforming LMS :

Un des algorithmes les plus simples pour le traitement adaptatif est basé sur **Least Mean Square (LMS)**. Bien que la complexité de l'algorithme soit très basse, ses résultats sont satisfaisants dans beaucoup de cas.

L'algorithme est très stable et il a besoin de peu de calculs, qui est important pour l'exécution de système.

L'algorithme de **LMS** est basé sur la connaissance du signal d'arrivée.

La connaissance du signal reçu élimine le besoin de Beamforming, mais la référence peut également être un vecteur cela est en partie connu, ou corrélé avec le signal reçu.

L'algorithme de LMS peut être considéré comme l'algorithme adaptatif le plus commun pour l'adaptation Continue, il calcule l'erreur et met à jour périodiquement le vecteur de poids.

Le vecteur d'erreur quadratique moyenne MSE entre la sortie du beamformer et le signal de référence est donné par la dérivation suivante [17].

MSE est défini par :

$$(v(n))^2 = (d^*(n) - w^H u(n))^2$$

Ou :

$u(n)$ Est le signal reçu

$w(n)$ Est le vecteur beamforming de poids.

$d^*(n)$ Représente le conjugué complexe du signal de référence

$w^H u(n)$ Est la réponse d du filtre adapté

D'après la démonstration vue dans le chapitre2 on aura :

$$E\{(v(n))^2\} = J = E\{d(n)^2\} - 2w^H R_{ud} + w^H R_{uu} w$$

Et on aboutit à l'équation de wiener-hopf :

$$w = (R_{uu})^{-1} R_{ud}$$

La forme final du LMS est donné par :

$$w(n+1) = w(n) + \mu_n u(n) E^*(n)$$

Résumé de l'algorithme LMS :**Paramètre :**

N = Nombre d'itération

\sim = Pas du LMS

$$0 < \sim < \frac{2}{\text{puissance totale du signal } u(n)}$$

Conditions initiales :

$$\underline{w}(0) = 0$$

Les données :

$u(n)$ = Vecteur de donnée à l'entrée à l'instant n

$d(n)$ = Sortie désiré a l'instant n

À calculer :

$\underline{w}(n+1)$ = Vecteur de poids estimé à l'instant $n+1$

Calcul :

Pour $n = 1, \dots, N$ calculer

$$v(n) = d^*(n) - \underline{w}^H(n) \underline{u}(n)$$

$$\underline{w}(n+1) = \underline{w}(n) + \sim_n \underline{u}(n) v^*(n)$$

3.3.2.2.2. Beamforming RLS :

L'algorithme de RLS [12] est un algorithme efficace à employer pour des applications Beamforming.

En utilisant l'algorithme de RLS pour la formation des poids et de leur convergence.

$e(n)$ est l'erreur la limite modifiait le vecteur W de poids pour fournir l'adaptation et la convergence.

Le facteur d'oublie nous permet de prendre les valeurs précédentes en considération, ce facteur d'oublie peut être variable sa formule [18] :

$$\lambda(n) = r \lambda(n-1) + 1 - r$$

Choix courant :

$$r \in [0.9, 0.99] \text{ et } \lambda(1) \in [0.9, 0.99]$$

$$\lambda(n) \xrightarrow{n \rightarrow \infty} 1$$

Ou on peut ajuster $\lambda(n)$ a chaque pas d'itération a fin de maintenir la trace de la matrice K_n constante.

On évite ainsi au gain d'adaptation de tendre vers zéro, cette méthode correspond au filtrage des signaux fortement non stationnaire.

$\lambda(n)$ Ce calcule à partir de l'équation suivante [18] :

$$trace[K_n] = \frac{1}{\lambda(n)} trace \left[K_{n-1} - \frac{K_{n-1} u(n) u(n)^T K_{n-1}}{1 + u(n)^T K_{n-1} u(n)} \right]$$

On a :

$u(n)$ Est le signal reçu

$w(n)$ Est le vecteur beamforming de poids.

$g(n)$ Variable s'appelle le vecteur de gain

K_n Est la matrice inverse de corrélation.

$d(n)$ Représente le signal de référence

Résumé de l'algorithme RLS :**Paramètre :**

N = Nombre d'itération
 $0 < \mu \ll 1$ Constante positive très petite
 $\beta \in [0.9, 0.99]$ Facteur d'oublie

Conditions initiales :

$\underline{w}(0) = 0$
 $\underline{K}_0 = \frac{I}{\mu}$ Ou et I matrice identité

Les données :

$u(n)$ = Vecteur de donnée à l'entrée à l'instant n
 $d(n)$ = Sortie désiré a l'instant n

À calculer :

$\underline{w}(n+1)$ = Vecteur de poids estimé à l'instant $n+1$

Calcul :

Pour $n = 1, \dots, N$ calculer

1-L'erreur a priori :

$$e(n) = d(n) - \underline{w}^H(n-1)u(n)$$

2-Le gain d'adaptation :

$$g(n) = \frac{\beta^{-1} \underline{K}_{n-1} u(n)}{1 + \beta^{-1} u^H(n) \underline{K}_{n-1} u(n)}$$

3-Mise a jour du vecteur de poids :

$$\underline{w}(n) = \underline{w}(n-1) + g(n)e^*(n)$$

4-Mise a jour de matrice d'auto corrélation :

$$\underline{K}_n = \beta^{-1} \underline{K}_{n-1} - g(n)u^H(n) \underline{K}_{n-1}$$

5-retour a (1)

Fin de la boucle

3.3.2.2. La complexité informatique de l'algorithme de RLS :

La complexité informatique de l'algorithme de RLS est dans l'ordre de $K M^2$.

Où M est le nombre d'éléments d'antenne dans la rangée et K le nombre de sources.

Elle est représentée dans le **tableau.3.2**

L'algorithme	Multiplications Complexes	Additions complexes
L'algorithme RLS	$4M^2 + 4M + 1$	$3M^2 + M$

tableau.3.2 La complexité informatique de l'algorithme de RLS.

Exemple :

L'algorithme RLS	Multiplications Complexes	Additions complexes
M=2	25	14
M=4	81	52
M=6	169	114
M=8	289	200

3.4. Conclusion :

Dans cette partie nous avons présenté le contexte général de notre étude. Après une introduction du concept des antennes intelligentes : structure, types et avantages et une mise au point sur les différentes méthodes actuelles de traitement d'antennes intelligentes (**RLS, LMS**), nous avons adopté la technique de formation de voies.

Cette technique est indépendante des informations portées par le signal donc indépendante du type d'application, aussi elle est simple à installer et permet de réaliser le formage du lobe directif avec un temps de calcul et un coût relativement faible.

Chapitre N°04

Simulation

2.1. Introduction :

2.1.1. Objectifs et déroulement du travail :

De façon générale les filtres adaptatifs sont des systèmes appliqués sur des données bruitées pour obtenir une information utile à un certain instant (t), ces systèmes étant mis en oeuvre dans trois configurations :

Le filtrage c'est à dire l'extraction de cette information au temps t à partir des données bruitées mesurées jusqu'au temps t inclus.

Le lissage qui utilisera aussi les données postérieures au temps t .

La prédiction qui ne se sert que des données jusqu'au temps ($t - 1$) pour déduire l'information qui nous intéresse au temps t .

Dans la première partie de ce chapitre nous exposerons l'approche statistique du problème (filtrage de Wiener) qui suppose la disponibilité de certaines grandeurs statistiques (moyenne et autocorrélation) du signal utile et du bruit. L'approche consiste alors à minimiser la moyenne statistique du carré de l'erreur (EQM ou MSE en anglais) entre l'information désirée et la sortie du filtre.

Ce filtrage de Wiener [1] est inadéquat pour les situations dans lesquelles le signal ou le bruit sont non stationnaires [2]. Dans de telles situations le filtre optimal doit être variable dans le temps. La solution à ce problème est fournie par le filtrage de Kalman.

Le filtrage adaptatif pourra aussi être utilisé dans ce cas. Par rapport au filtrage classique le filtrage adaptatif comporte une mise à jour récursive des paramètres (coefficients) du filtre. L'algorithme part de conditions initiales prédéterminées et modifie de façon récursive les coefficients du filtre pour s'adapter au processus. Si celui-ci est stationnaire l'algorithme doit converger vers la solution optimale de Wiener, sinon il présentera une capacité à suivre des variations des grandeurs statistiques du processus si celles-ci sont suffisamment lentes.

Pour présenter ces techniques adaptatives nous présenterons tout d'abord l'algorithme du gradient qui fournit un algorithme récursif de calcul des coefficients du filtre [3]. Nous donnerons ensuite une version dans laquelle les grandeurs statistiques impliquées sont remplacées par des valeurs instantanées, on obtient alors l'algorithme très fréquemment utilisé du gradient stochastique (LMS). Nous reformulerons le problème en termes de moindres carrés et présenterons un des algorithmes récursifs basé sur cette approche : l'algorithme des moindres carrés récursifs (RLS) [4].

2.1.2. Classification des filtres:

Les filtres adaptatifs peuvent être classés en fonction des choix qui sont faits sur les points suivants:

- Le critère d'optimisation,
- L'algorithme de mise à jour des coefficients,
- La structure du filtre programmable,
- Le type de signal traité, mono ou multidimensionnel.

4.1. Introduction :

Beamforming adaptatif peut être exécuté dans beaucoup de manières.

Les algorithmes existent pour beaucoup d'applications changeant dans la complexité.

Les algorithmes temporels de référence se servent du peu d'ordre de formation pour la convergence des poids du Beamforming.

Certains des algorithmes peuvent être utilisés pour Beamforming :

Moindres carrés moyennes (LMS) et Algorithmes des moindres carrés récurrents (RLS).

- L'algorithme de LMS peut être considéré comme l'algorithme adaptatif le plus commun pour l'adaptation continue, il calcule l'erreur et met à jour périodiquement le vecteur de poids.
- L'algorithme de RLS est un algorithme efficace à employer pour des applications Beamforming.

En utilisant l'algorithme de RLS pour la formation des poids et de leur convergence.

$e(n)$ est l'erreur la limite modifie le vecteur W de poids pour fournir l'adaptation et la convergence.

4.2. Simulations et commentaires :

Pour nos simulations, l'antenne utilisée est une antenne linéaire de M capteurs équi-répartis, omnidirectionnels et de gain unité ; le milieu de propagation est un milieu non dispersif, Les sources ont été supposées suffisamment loin du réseau pour les considérer comme des ondes planes. La distance inter capteur est égale à une demi longueur d'onde.

Le bruit a été supposé blanc dans la bande passante de l'antenne et sa puissance est normalisée à 1, à phase uniforme ($-\pi$, π), et amplitude de distribution gaussienne.

Le bruit supplémentaire du système à tous les antennes est un bruit blanc avec une moyenne nulle et une variance 0.1.

Les signaux reçus sont des signaux à modulation MSK avec un échantillonnage supérieur (haut prélèvement) d'ordre 4, et une amplitude normalisée égale à 1. Le signal de sortie vrai $y(t)$ converge vers le signal de référence $d(t)$ (désiré). Les interférences sont éliminées en plaçant des nulles dans leurs directions.

On a élaboré la simulation pour :

- Angle d'arrivée constant.
- Angle d'arrivée variable (variation linéaire).

4.3. Organigramme et interface de simulation :

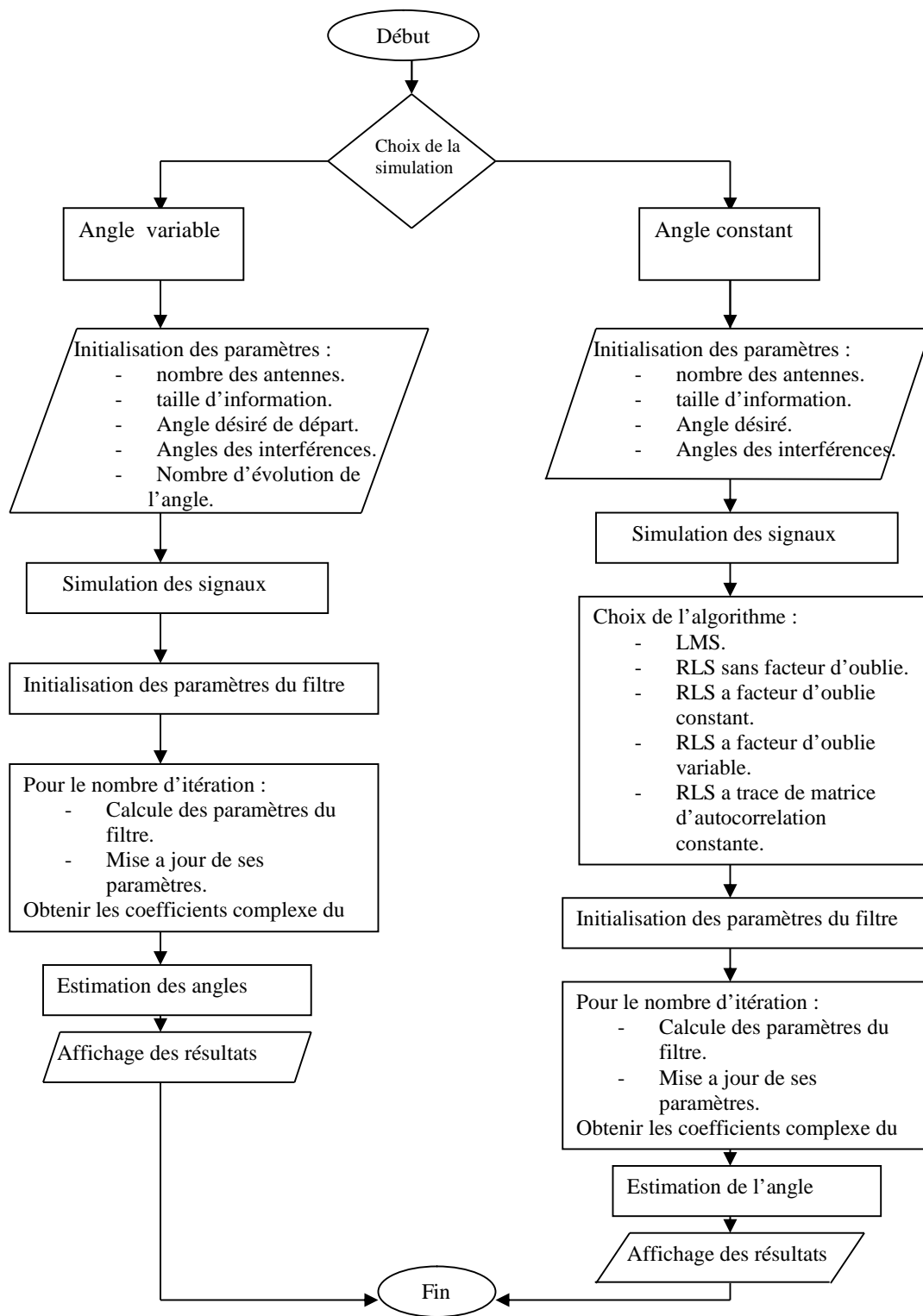


Figure 4.1 - Organigramme de simulation.

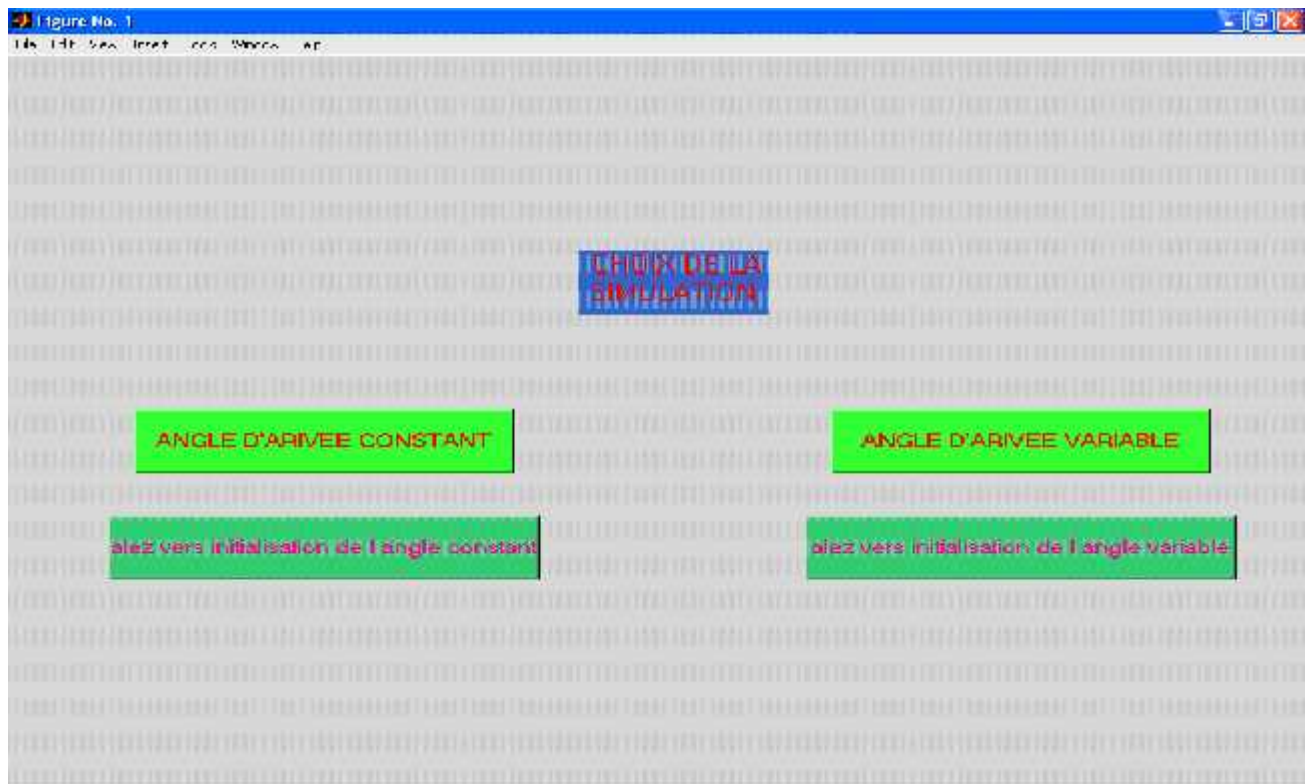


Figure 4.2 - Interface de simulation du programme (sur Matlab).

4.4. Simulation et commentaires pour AOA (DOA) constante :

Dans cette partie on simule Beamforming adaptatif pour un angle d'arrivé constant, On variant a chaque fois l'angle d'arrivé de la cible, les angles d'arrivés des interference, le nombre d'itération, et le nombre d'antenne.

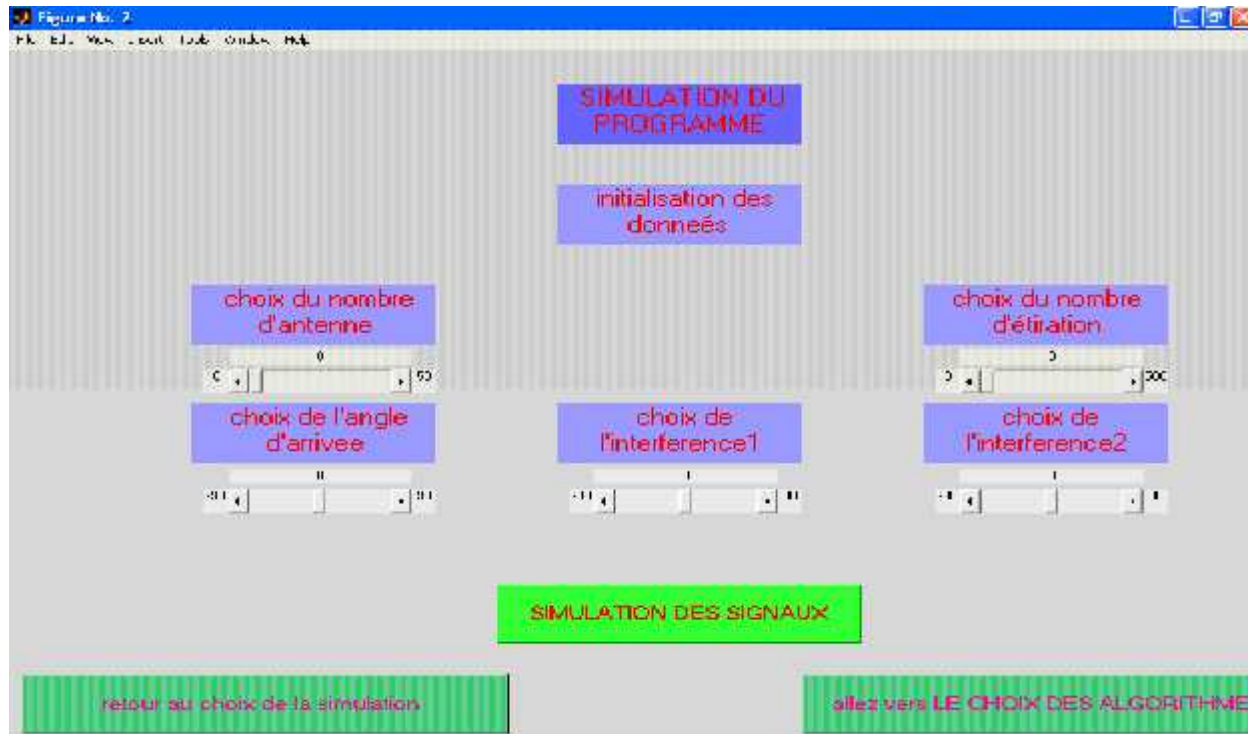


Figure 4.3 - Initialisation des données pour AOA constant.

Et voir l'évolution de l'adaptation par chaque algorithme. on utilisant les algorithmes :

- LMS.
- RLS sans facteur d'oublie.
- RLS a facteur d'oublie constant.
- RLS a facteur d'oublie variable.
- RLS a facteur d'oublie de trace constante.



Figure 4.4 - Choix de l'algorithme.

4.4.1. **Simulation n°01** : Analyse de poursuite des signaux par les différentes méthodes de filtrage adaptatif beamformer.

Initialisation n°01 :

Nombre d'antenne	8
Nombre d'information (itération /4)	100
Angle d'arrivée du signal désiré (°)	18
Angle d'arrivée d'interférence 1 (°)	72
Angle d'arrivée d'interférence 1 (°)	-72

Tableau.4.1-initialisation de la simulation n°01

Résultats n°01 :

Poursuite de la phase :

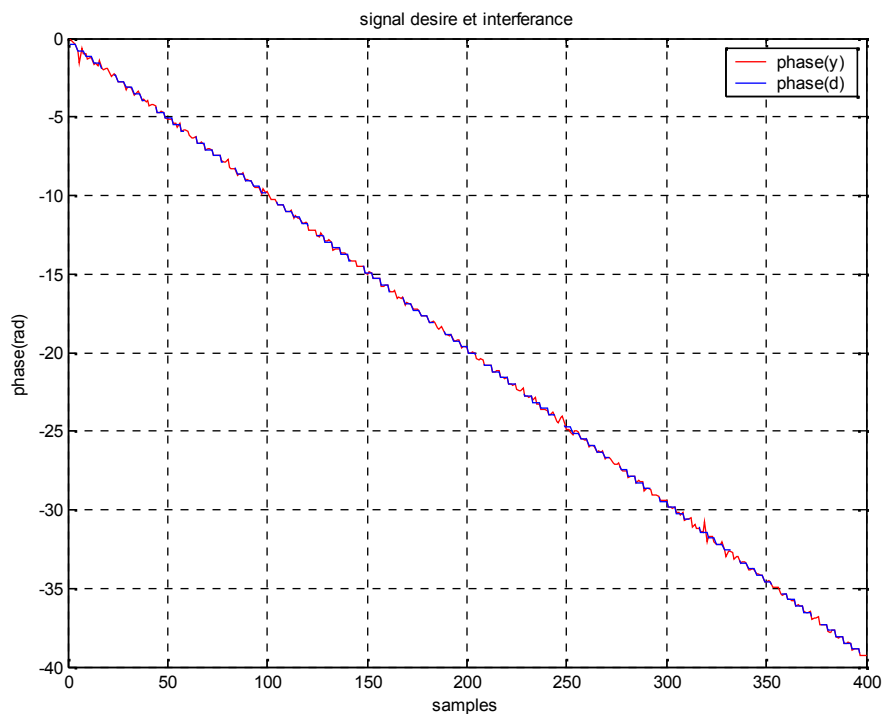


Figure.4.5- poursuite de la phase par LMS

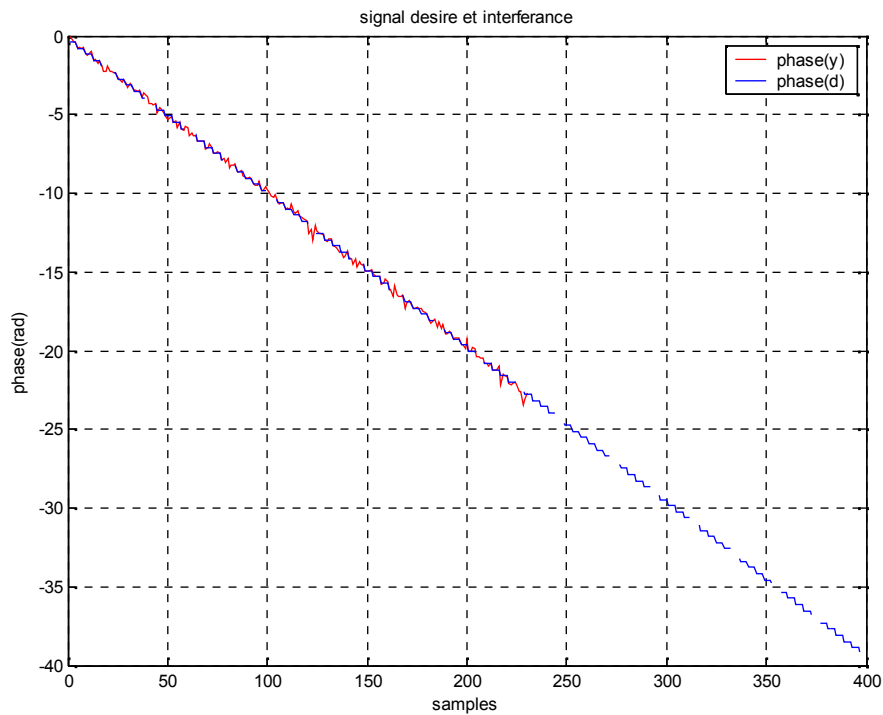


Figure.4.6- Poursuite de la phase par RLS sans facteur d'oublie

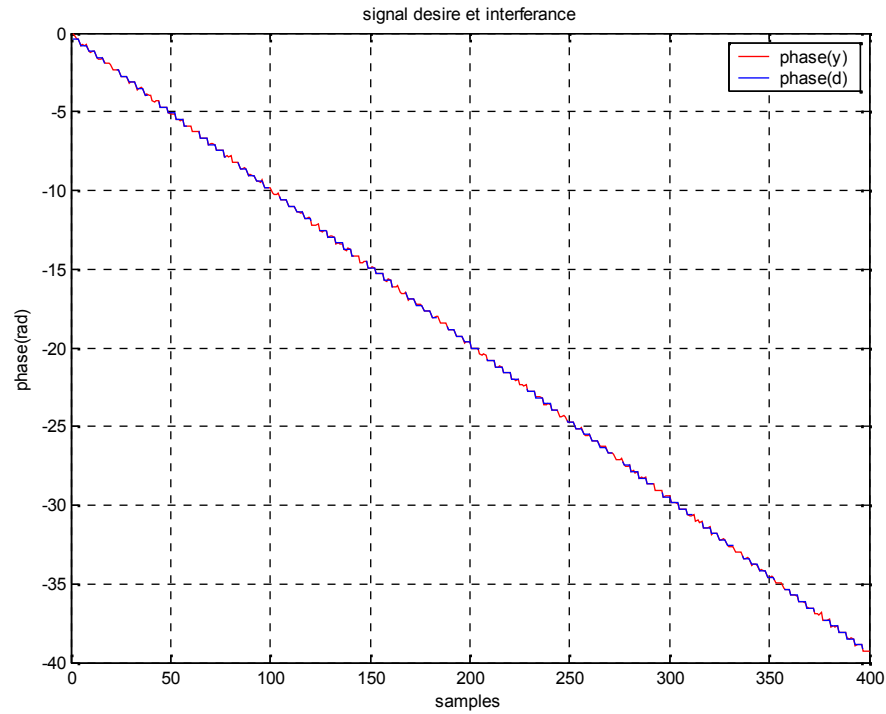


Figure.4.7- Poursuite de la phase par RLS à facteur d'oubli constant

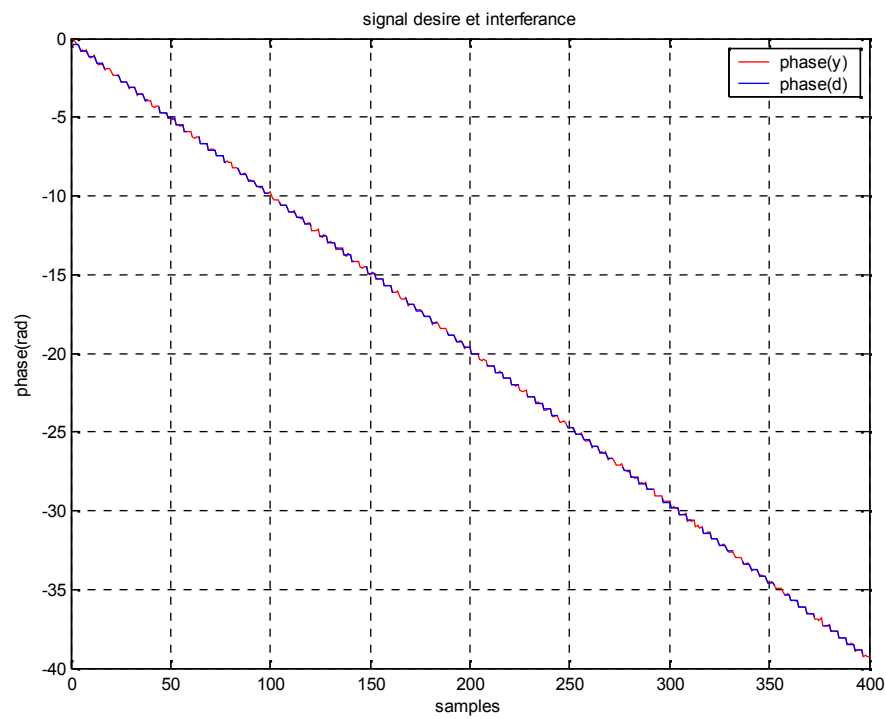


Figure.4.8- Poursuite de la phase par RLS à facteur d'oubli variable

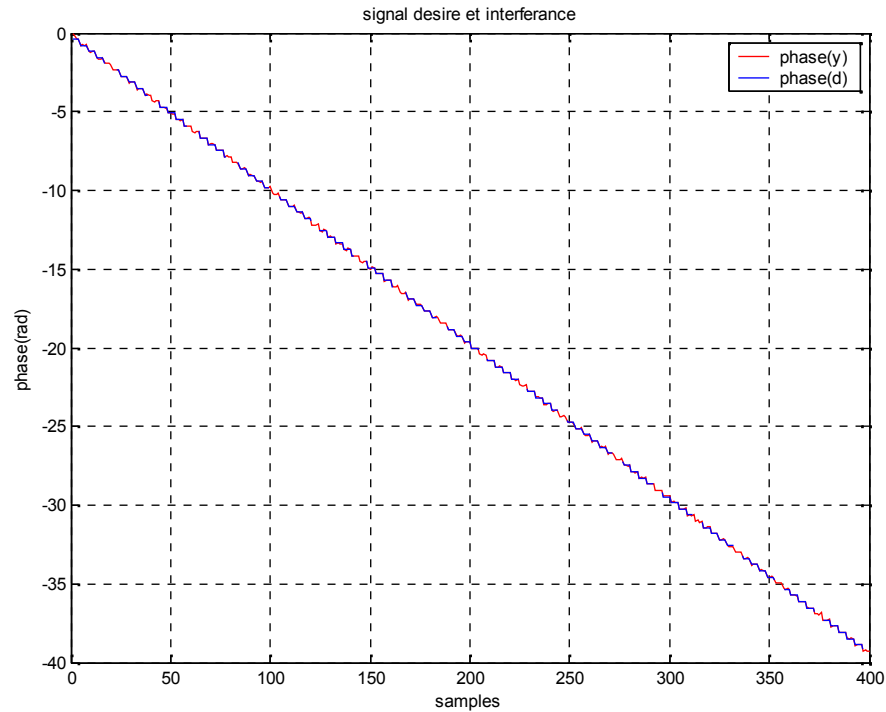


Figure.4.9- Poursuite de la phase par RLS à facteur d'oublie a trace constante.

Commentaires :

Dans la *Figure.4.5*, on voit bien la bonne poursuite de la phase par l'algorithme LMS, pour un nombre d'échantillon très grand, malgré une instabilité entre les échantillons 300 et 350.

Pour la poursuite du RLS sans facteur d'oublie (*Figure.4.6*), on remarque une bonne poursuite avec une maigre stabilité jusqu'à l'échantillon 230, au delà de cet échantillon l'algorithme n'assure plus la poursuite.

Pour les algorithmes du RLS a facteur d'oublie, on voit une très bonne poursuite et stabilité avec une meilleure rapidité de convergence que l'algorithme du LMS.

Poursuite de la magnitude :

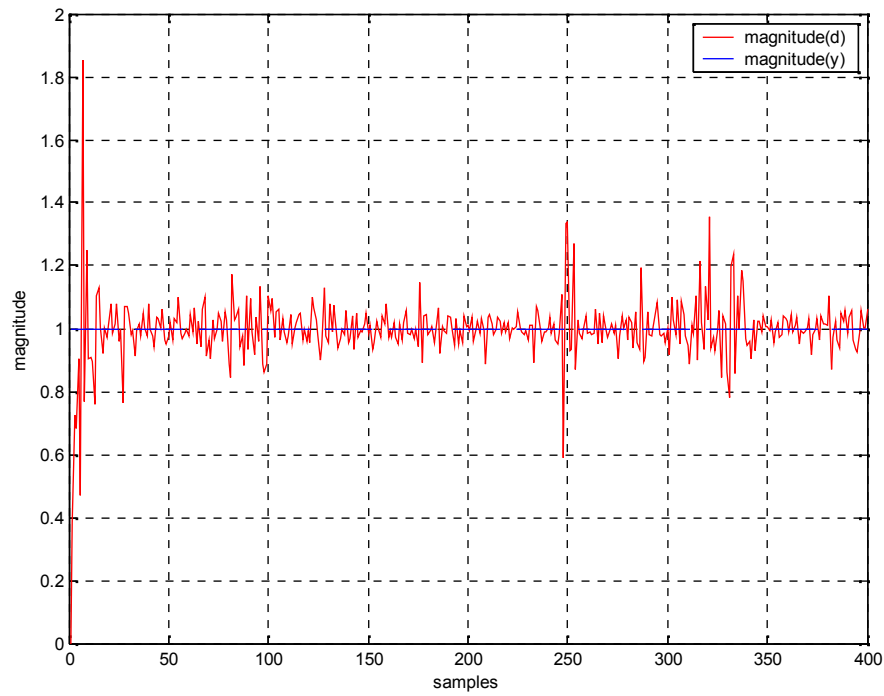


Figure.4.10- poursuite de la magnitude par LMS.

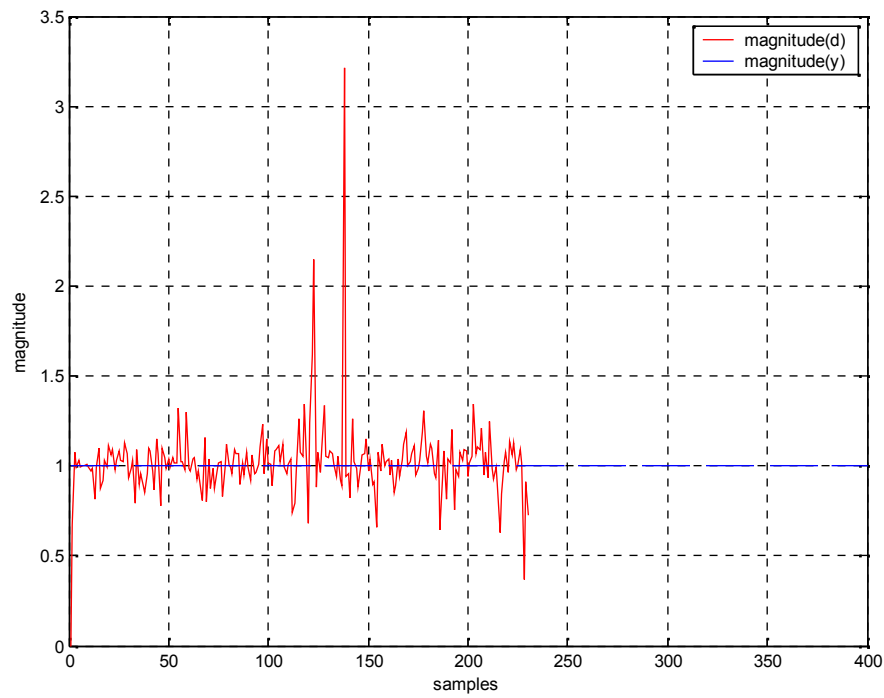


Figure.4.11- Poursuite de la magnitude par RLS sans facteur d'oublie.

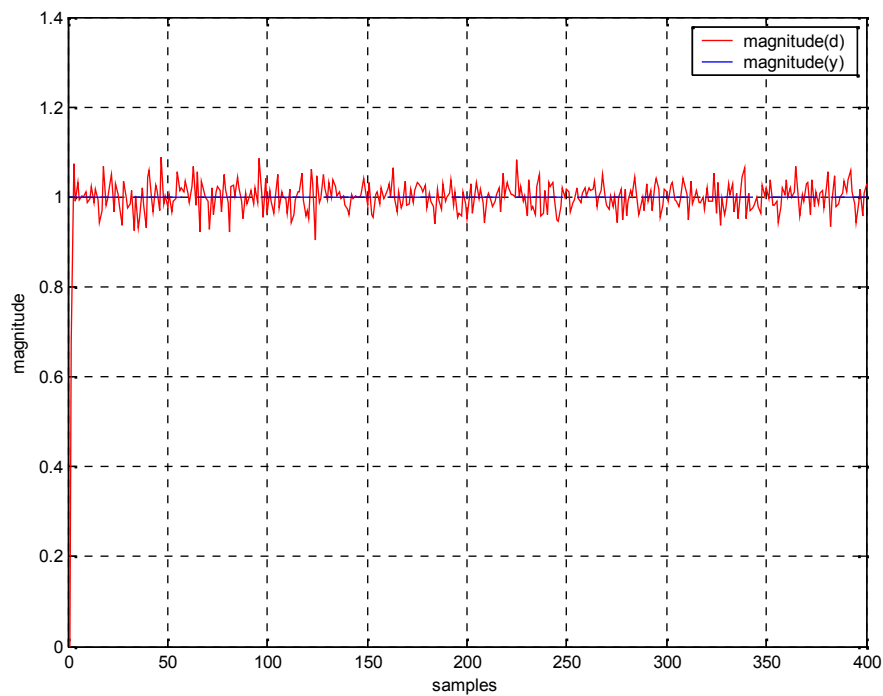


Figure.4.12- Poursuite de la magnitude par RLS à facteur d'oublie constant.

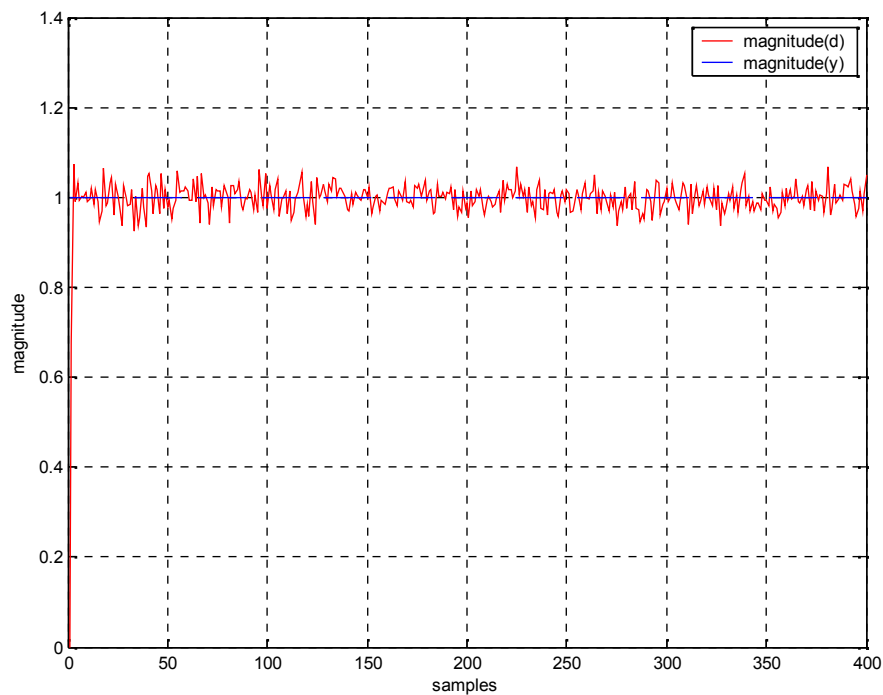


Figure.4.13- Poursuite de la magnitude par RLS à facteur d'oublie variable.

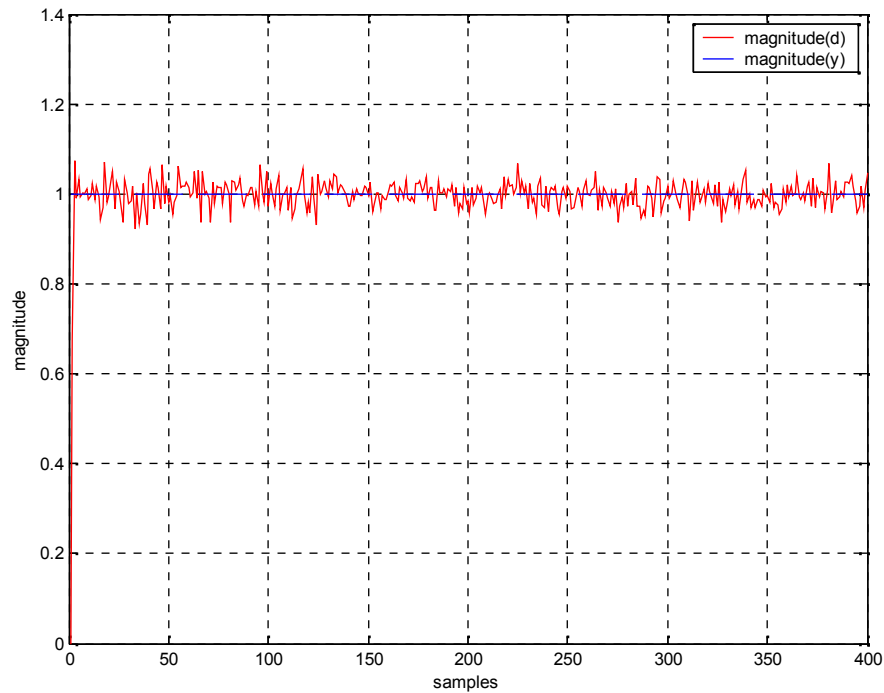


Figure.4.14- Poursuite d la magnitude par RLS à facteur d'oublie a trace constante.

Commentaires :

La poursuite de la magnitude est identique a celle de la phase, mais plus visible car le bruit influe sur l'amplitude plus qu'il influe sur la phase, les algorithmes RLS a facteur d'oublie reste plus stable que les algorithmes LMS et RLS sans facteur d'oublie.

Evolution de l'erreur quadratique :

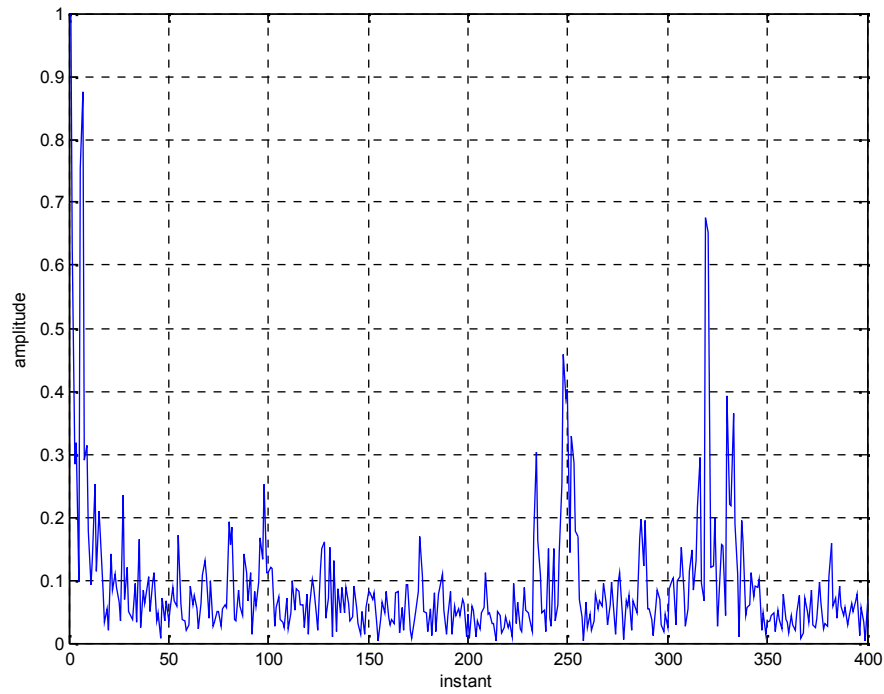


Figure.4.15-Evolution de l'erreur quadratique par LMS.

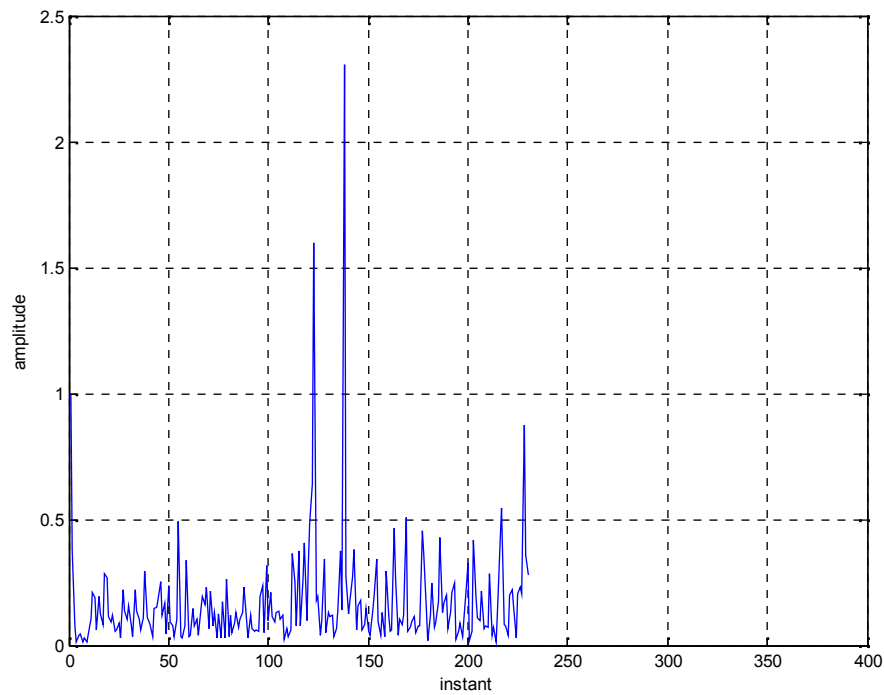


Figure.4.16- Evolution de l'erreur quadratique par RLS sans facteur d'oublie.

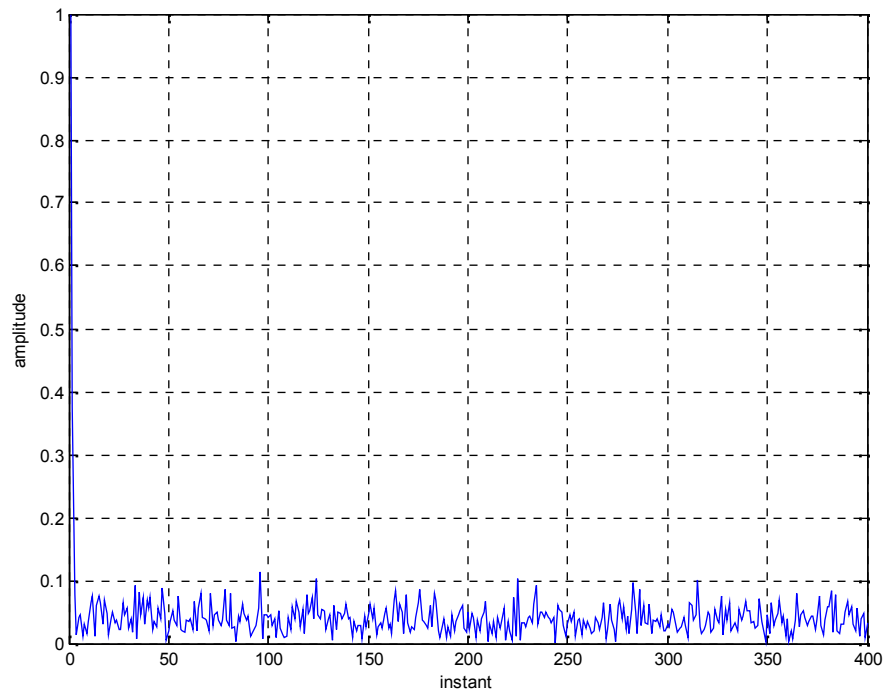


Figure.4.17- Evolution de l'erreur quadratique par RLS à facteur d'oubli constant.

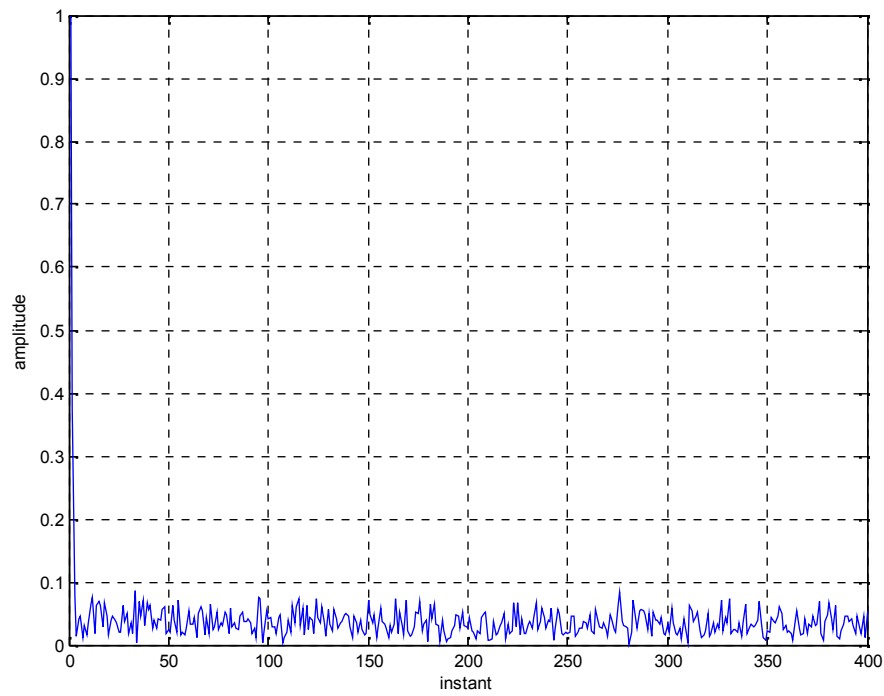


Figure.4.18- Evolution de la l'erreur quadratique par RLS à facteur d'oubli variable.

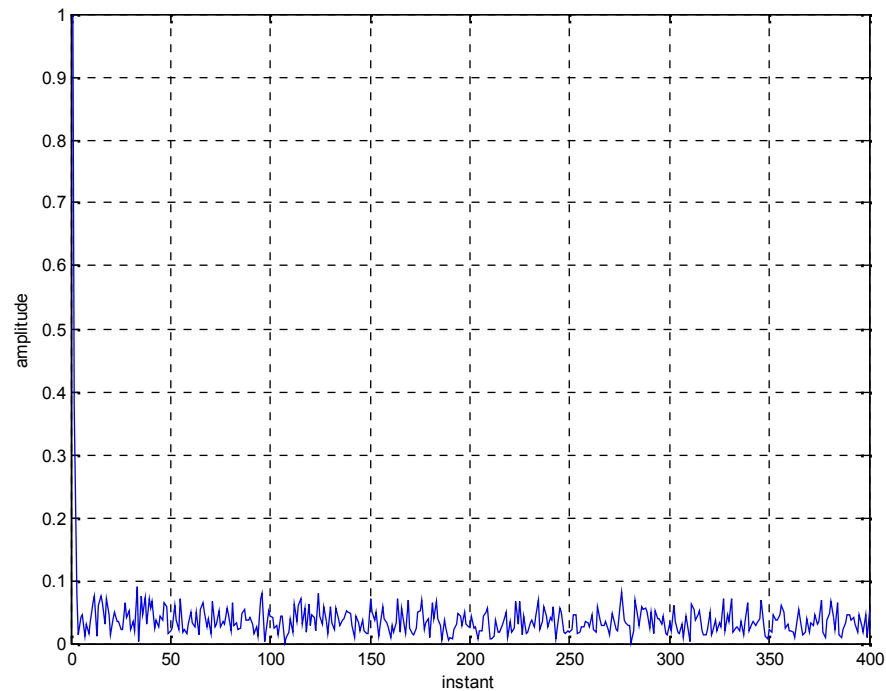


Figure.4.19- Evolution de l'erreur quadratique par RLS à facteur d'oublie a trace constante.

Commentaires :

L'évolution de l'erreur suit le cours de la poursuite de phase et magnitude en terme de convergence et stabilité, pour les algorithmes RLS a facteur d'oublie on a une meilleure minimisation de l'erreur quadratique.

Ce qui signifie une meilleure optimisation et adaptation pour les algorithmes RLS a facteur d'oublie constant variable ou a trace de la matrice de autocorrelation constante.

Tracé de la réponse du filtre :

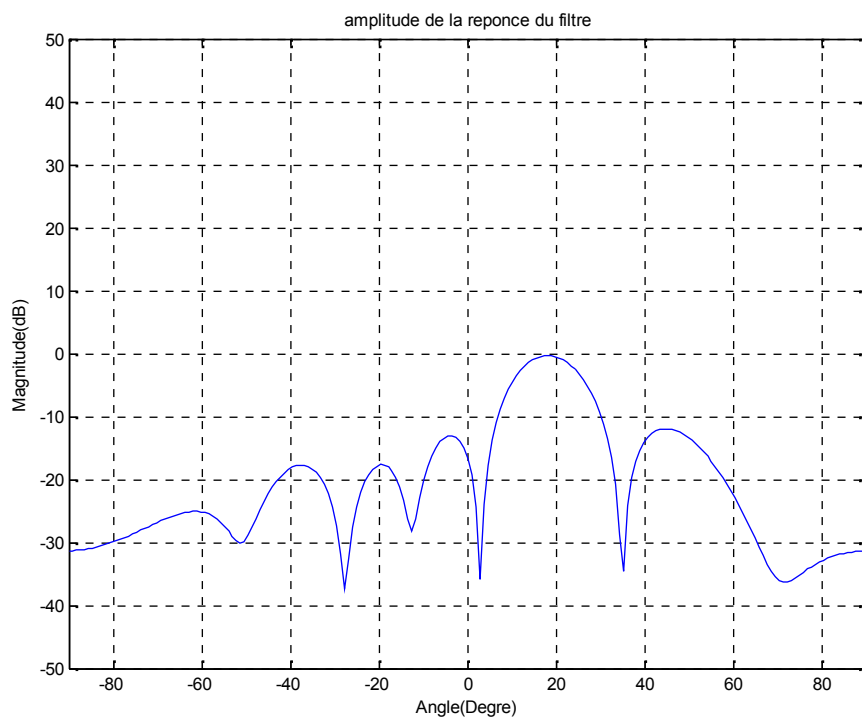


Figure.4.20- réponse du filtre LMS.

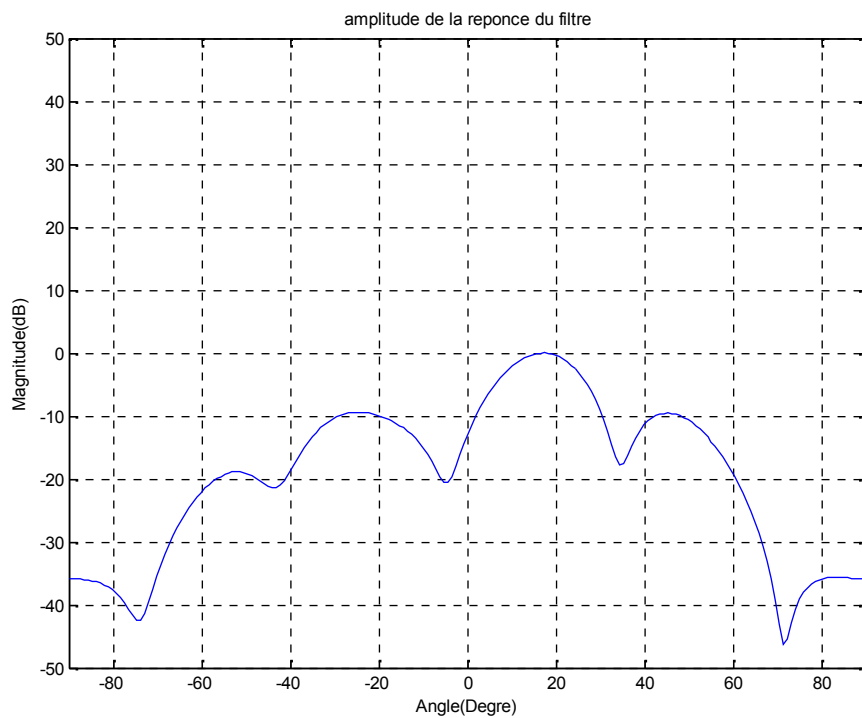


Figure.4.21- réponse du filtre RLS a facteur d'oublie constant.

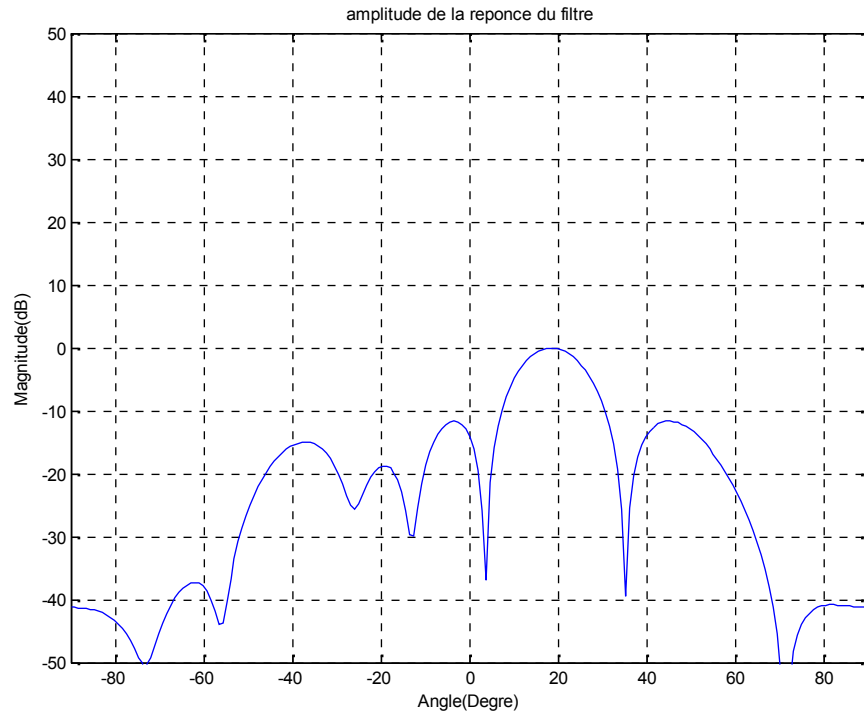


Figure.4.22- réponse du filtre RLS facteur d'oublie variable.

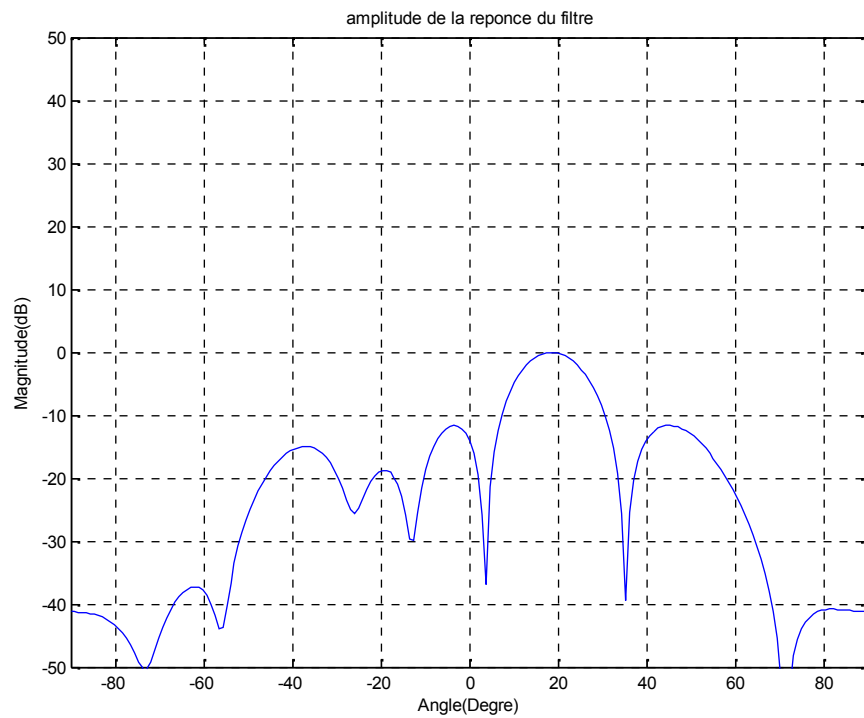


Figure.4.23- réponse du filtre RLS a trace constante.

Commentaires :

Dans la *Figure.4.20*, la réponse du filtre adaptatif LMS donne un pic maximal dans notre direction désiré(18°) et des minimums aux directions des interférences(-72° et 72°).

Ces pics sont plus net dans le cas des filtres adaptatifs a facteur d'oublie variable et trace constante (*Figure.4.22*, *Figure.4.23*), ce qui n'est pas le cas, du RLS a facteur d'oublie constant (*Figure.4.21*).

Le filtre adaptatif sans facteur d'oublie ne donne pas de réponse à l'échantillon 400, car il a perdue le signal à l'échantillon 230.

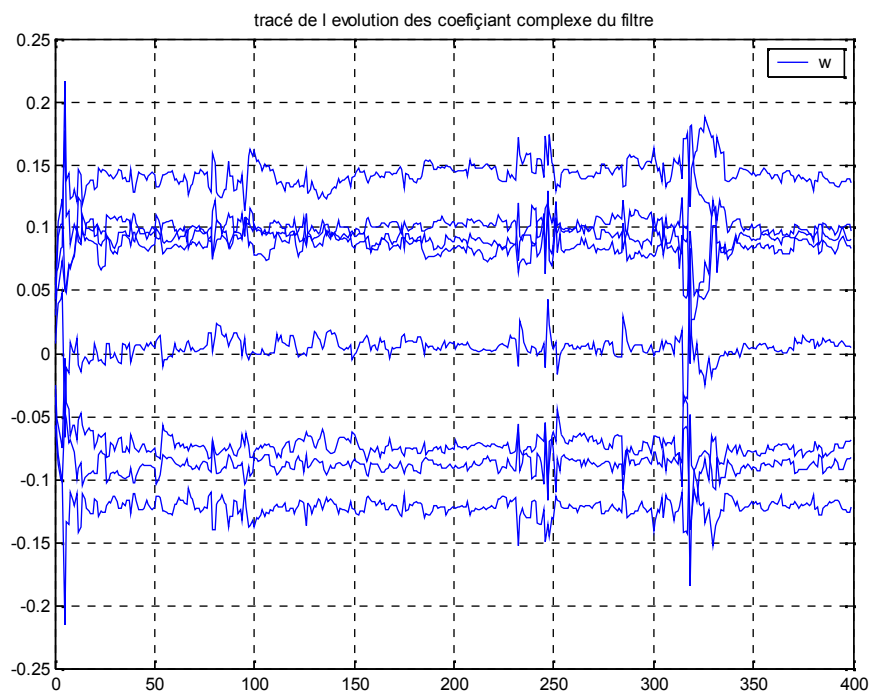
Evolution des coefficient complexe des filtres (convergence, stabilité) :

Figure.4.24- évolution des coefficient complexe du filtre LMS.

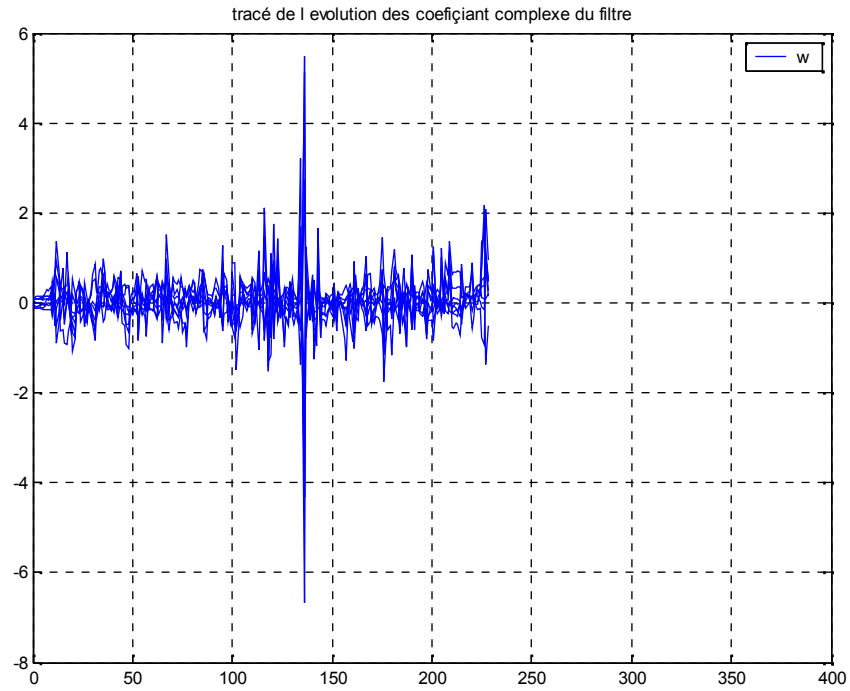


Figure.4.25- évolution des coefficiant complexe du filtre RLS sans oublier.

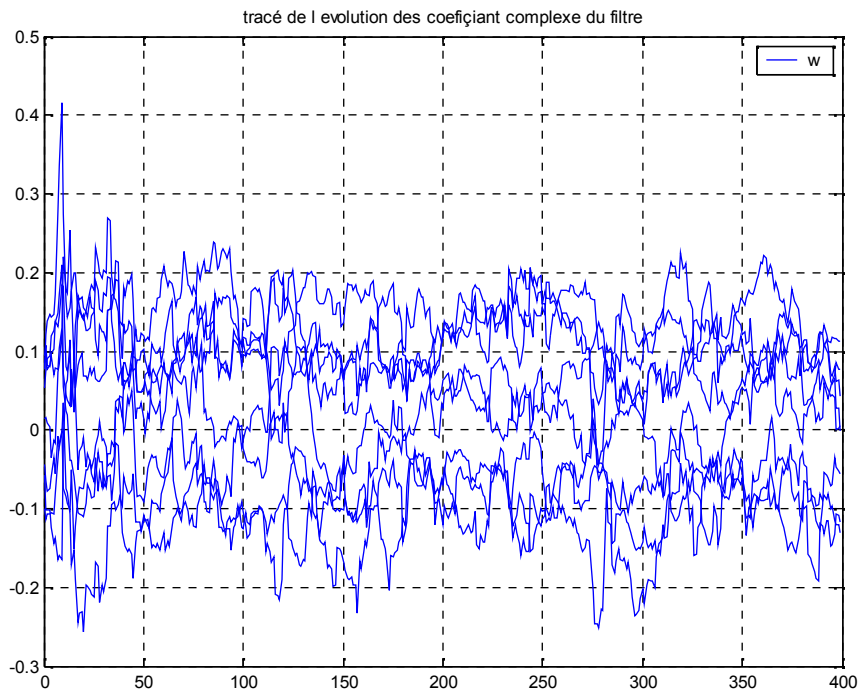


Figure.4.26- évolution des coefficiant complexe du filtre RLS a facteur d'oublie constant.

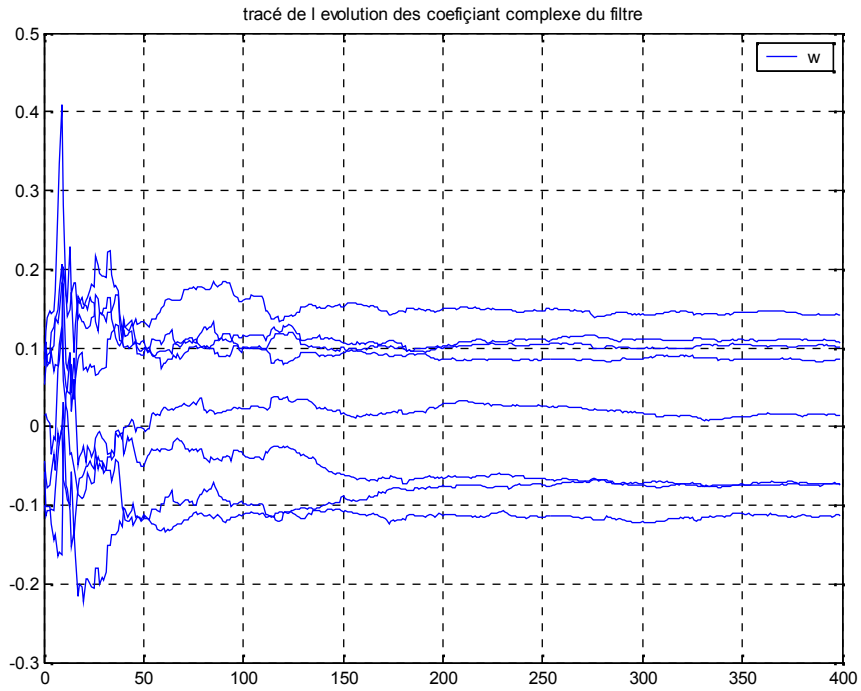


Figure.4.27- évolution des coefficant complexe du filtre RLS a facteur d'oublie variable.

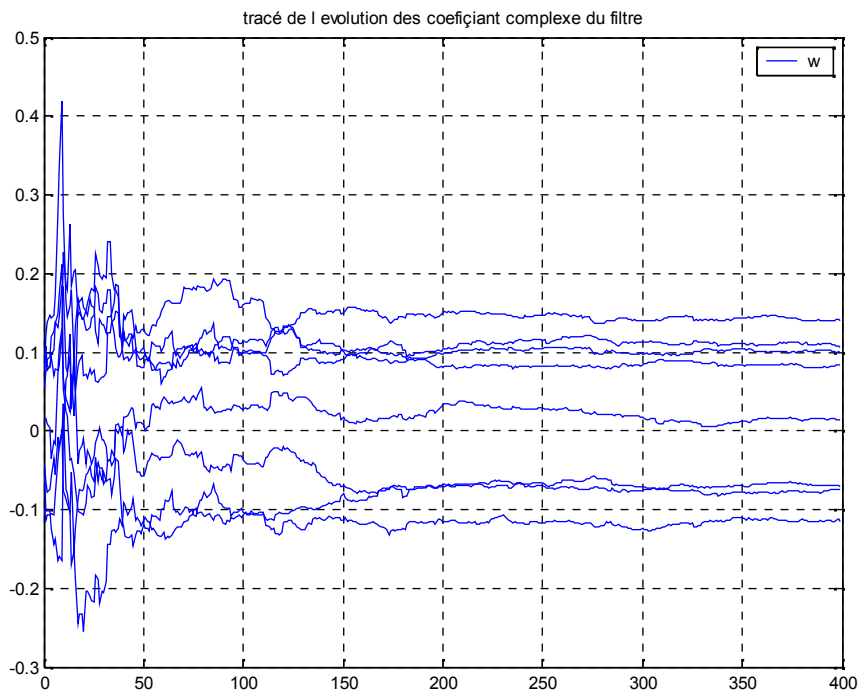


Figure.4.28- évolution des coefficant complexe du filtre RLS a trace constante.

Commentaires :

L'évolution des coefficients complexe nous donne une meilleure explication a la réponse de chaque filtre, on voit mieux la rapidité de convergence et la stabilité.

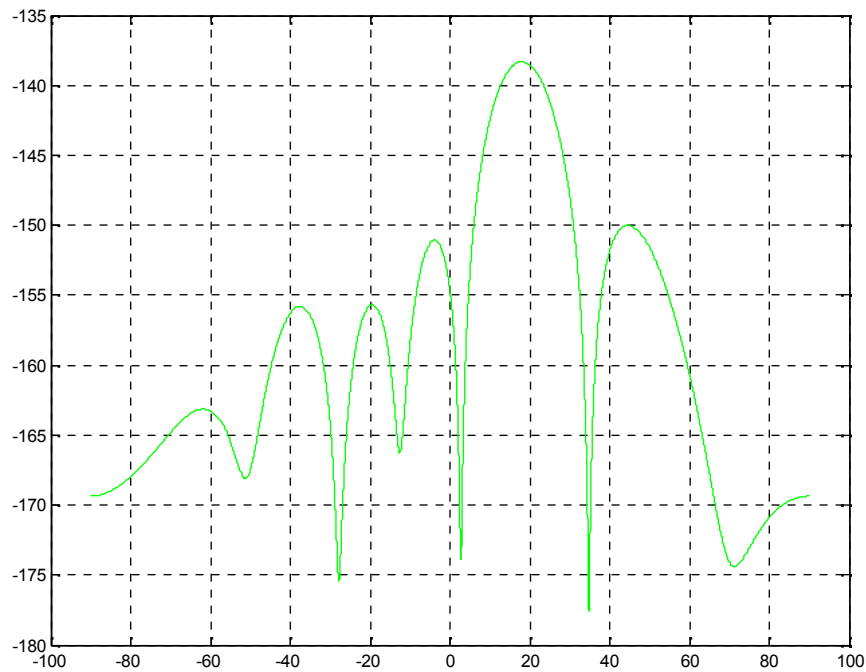
Tracé des réponses des antennes :

Figure.4.29- réponse des l'antennes au filtre LMS.

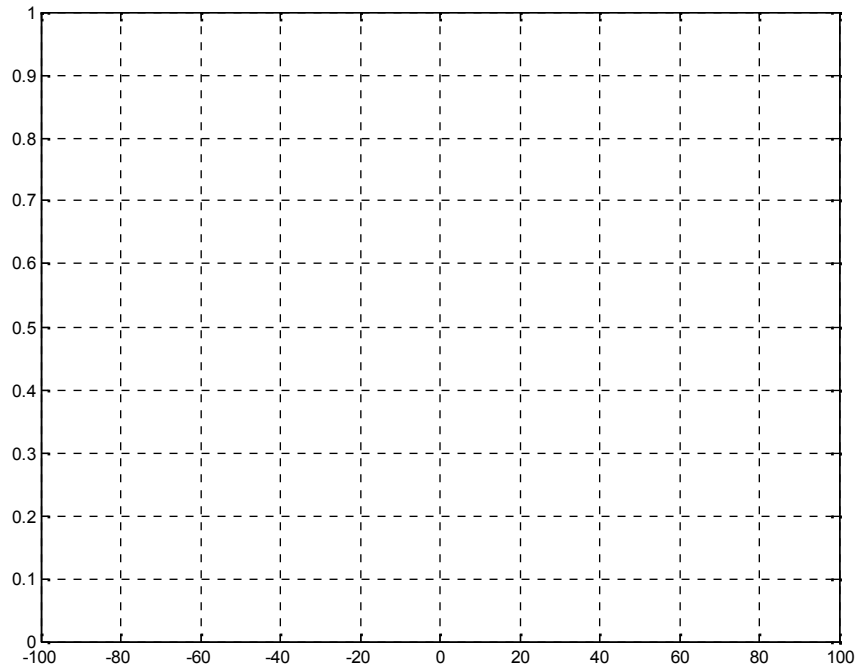


Figure.4.30- réponse des l'antennes au filtre RLS sans oublier.

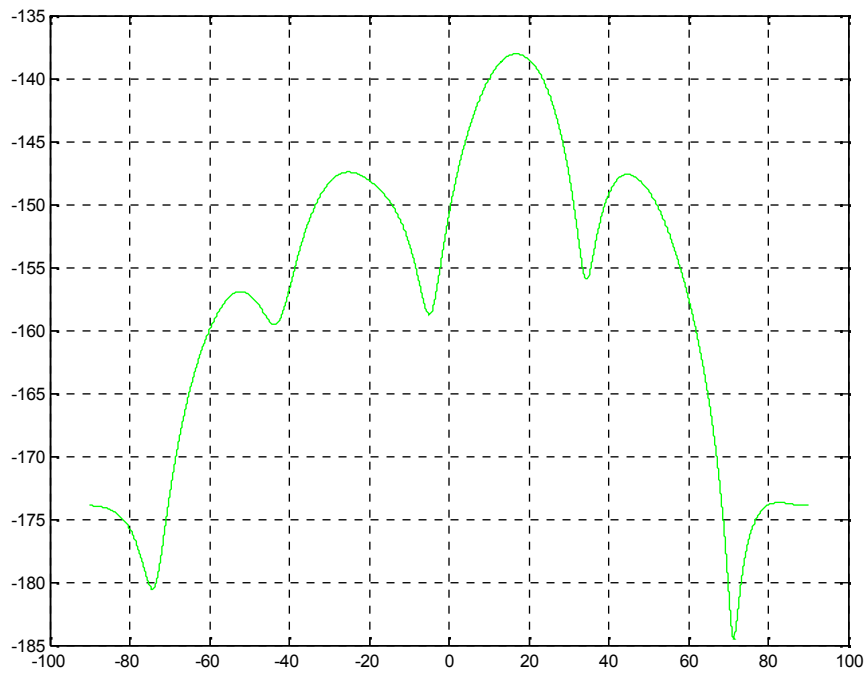


Figure.4.31- réponse des antennes au filtre RLS a facteur d'oublie constant.

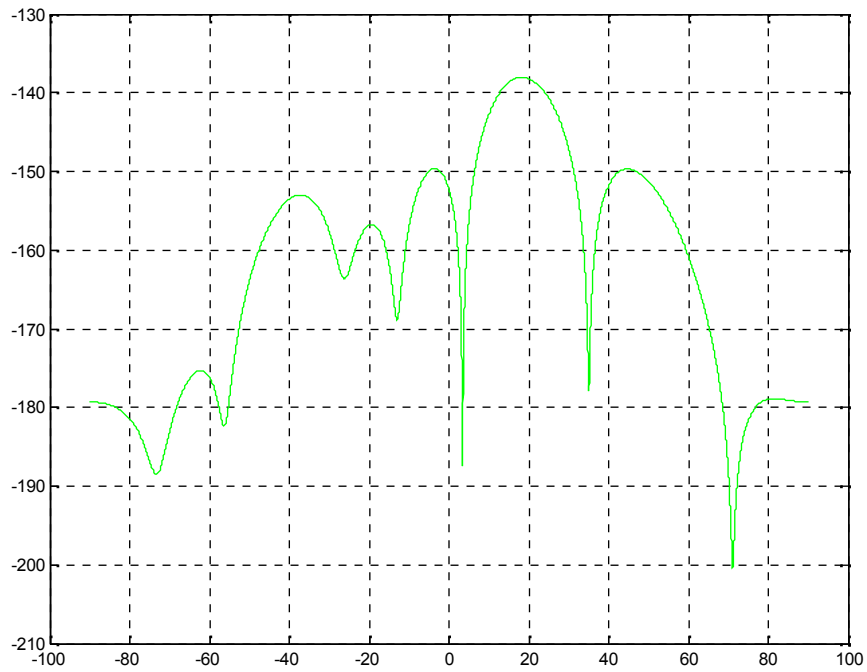


Figure.4.32- réponse des antennes au filtre RLS a facteur d'oublie variable.

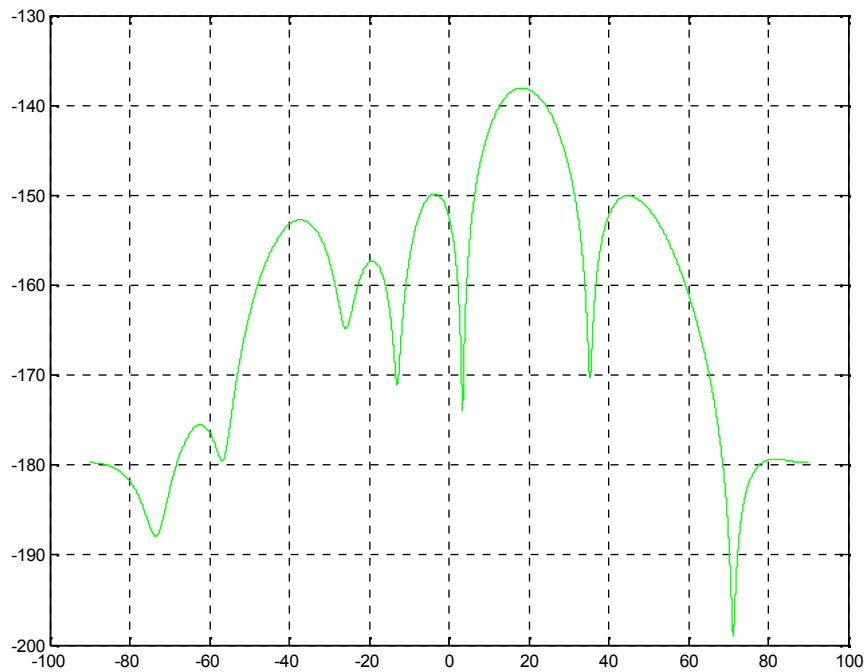


Figure.4.33- réponse des antennes au filtre RLS a trace constante.

Commentaires :

La réponse des antennes est identique a celle des filtres pour chaque algorithme adaptatif.

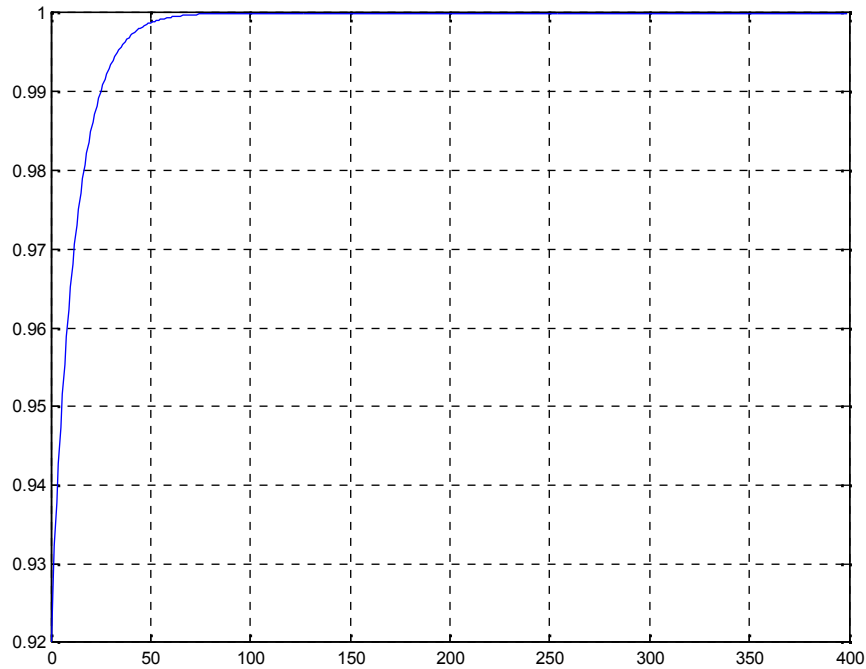
Evolution du facteur d'oublie variable :

Figure.4.34- évolution du facteur d'oublie variable.

Commentaires :

Le facteur d'oublie a une évolution croissante et tend vers 1, et reste dans l'intervalle $[0.90,1]$ comme dans l'étude théorique.

Evolution du gain d'adaptation :

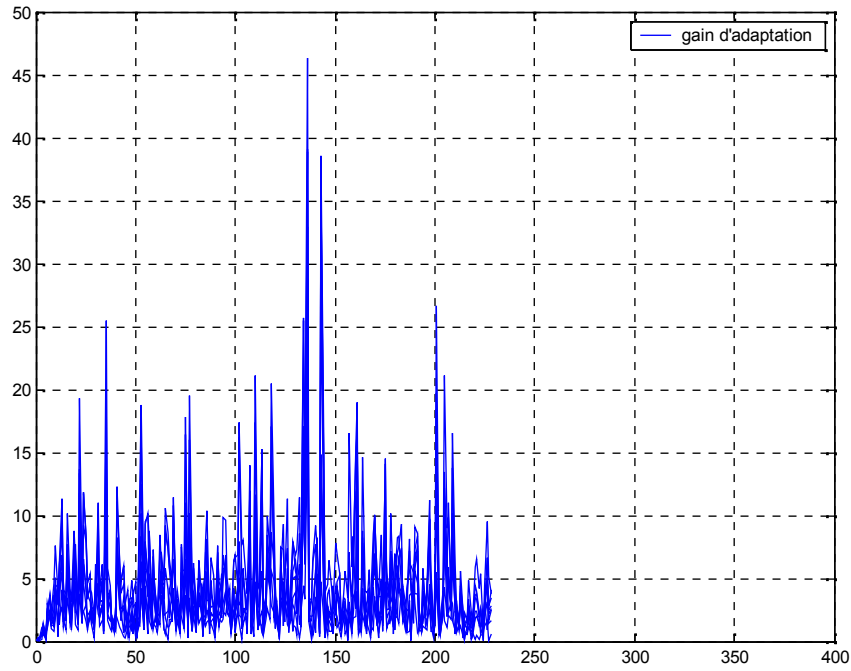


Figure.4.35- évolution du gain d'adaptation pour RLS sans oublier.

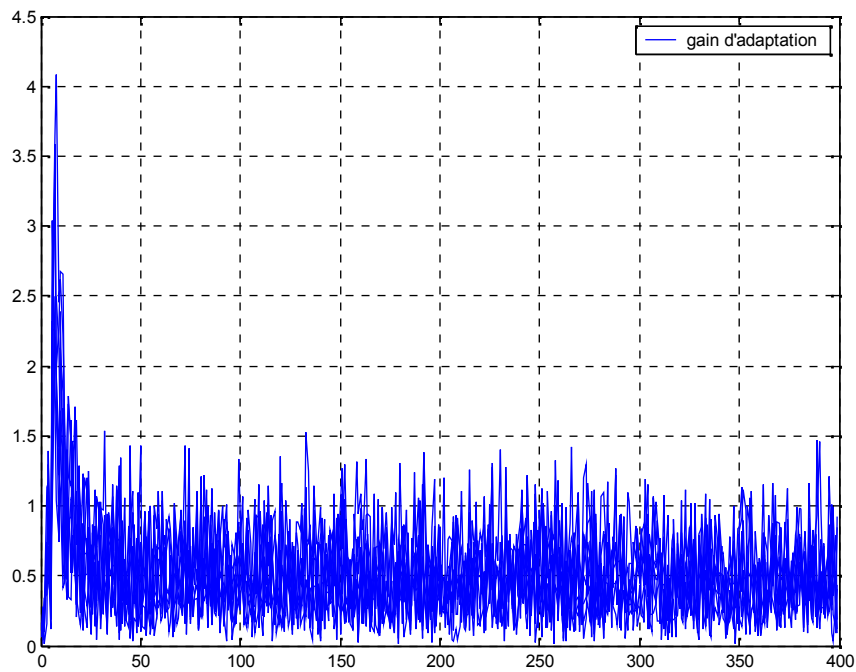


Figure.4.36- évolution du gain d'adaptation pour RLS a facteur d'oublie constant.

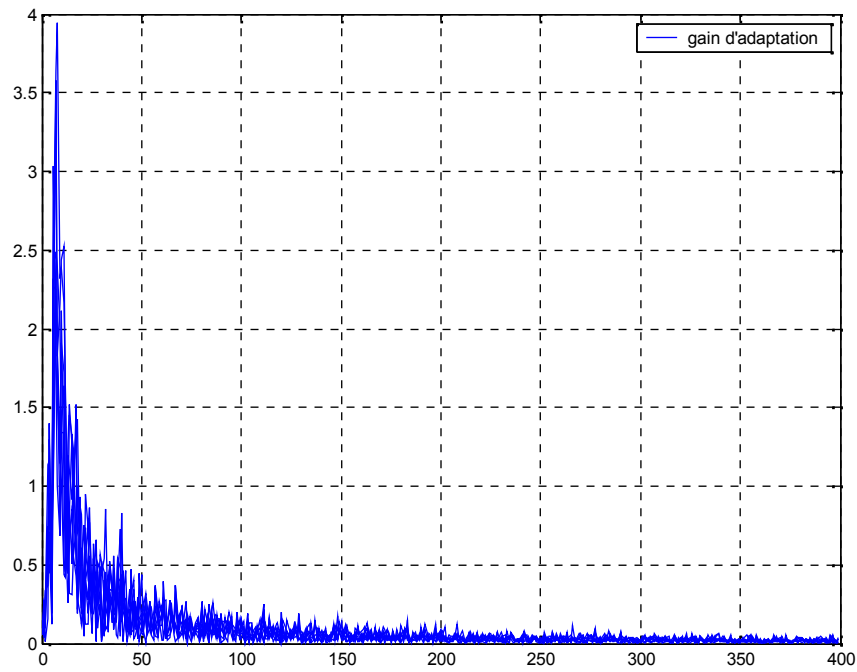


Figure.4.37- évolution du gain d'adaptation pour RLS a facteur d'oublie variable.

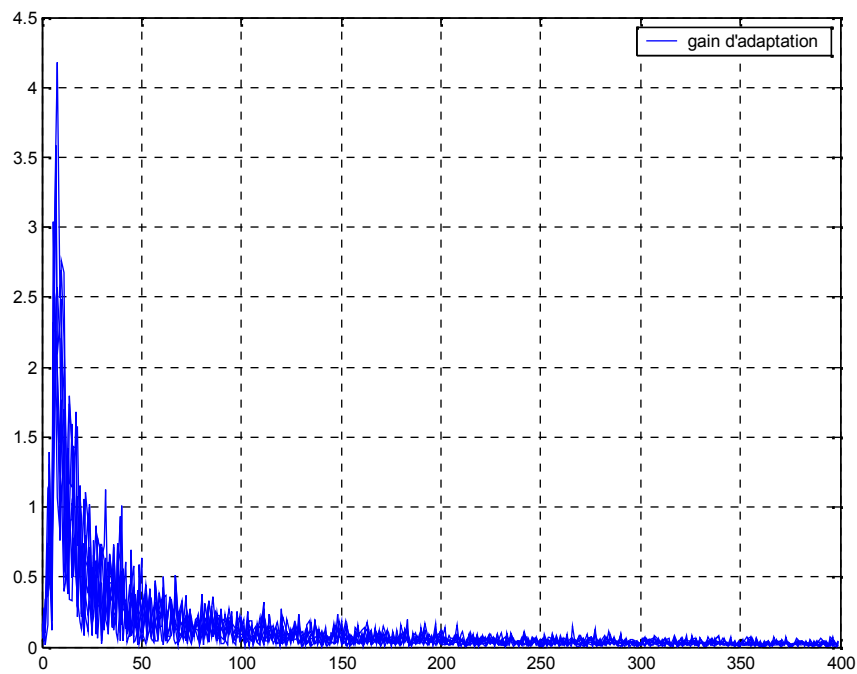


Figure.4.38- évolution du gain d'adaptation pour RLS a trace constante.

Commentaires :

Dans la *Figure.4.35*, on voit une évolution irrégulière du gain d'adaptation du RLS sans facteur d'oubli, ce qui explique l'irrégularité de l'évolution de l'erreur et l'instabilité des coefficients au cours de la simulation.

la *Figure.4.36*, donne une explication pour la maigre stabilité des coefficients du filtre adaptatif a facteur d'oubli constant.

Les Figures 4.37 et .4.38, montre une meilleure adaptation pour un facteur d'oubli variable.

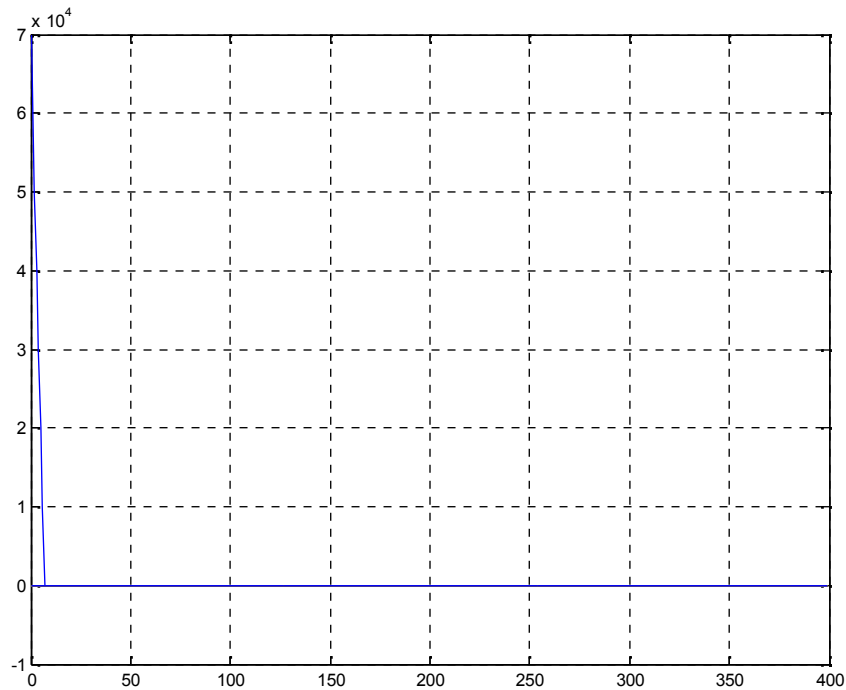
Tracé de l'évolution de trace de la matrice d'autocorrelation :

Figure.4.39- évolution de trace pour RLS sans facteur d'oubli.

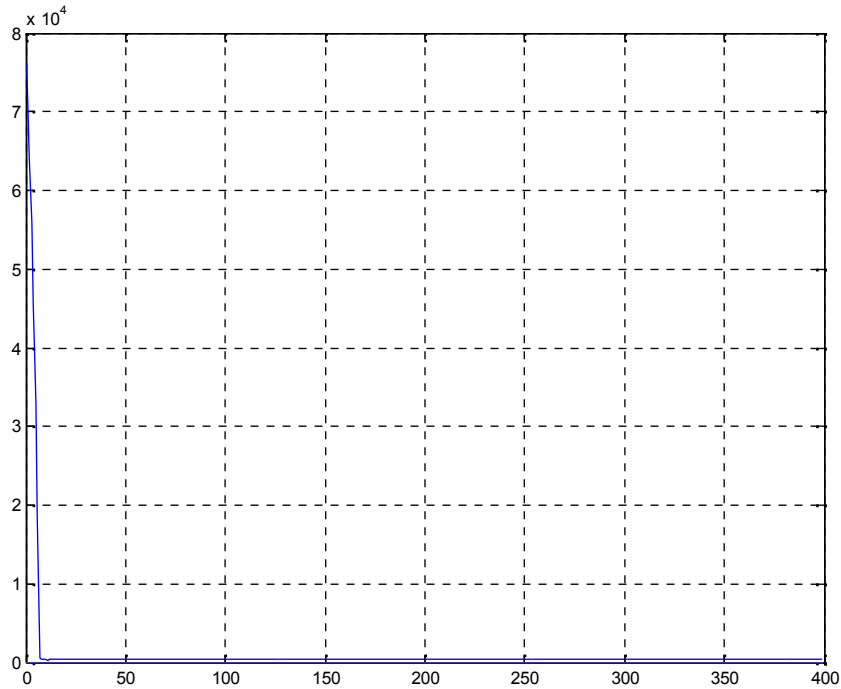


Figure.4.40- évolution de la trace pour RLS a trace constante.

Commentaires :

La *Figure.4.39*, montre l'évolution de la trace de la matrice d'autocorrelation pour l'algorithme RLS sans facteur d'oubli, et nous donne l'explication de la perte de poursuite de cet algorithme, due a la chute de sa trace a zéros ce qui a provoquer une devisions par zéros dans la mise a jour des coefficient du filtre ainsi pour le gain d'adaptation.

La *Figure.4.40*, nous montre l'évolution de la trace de la matrice d'autocorrelation pour l'RLS a trace constante, qui assure une trace constante qui mit a l'abrie notre algorithme d'une devisions par zéros et empêche de décrochage de la poursuite même pour un nombre illimité d'echontillon .c'est pour cela que cet algorithme est meilleur que tous les autres algorithmes.

Tracé des lobes et des angles vrais et estimés (vert, rouge):

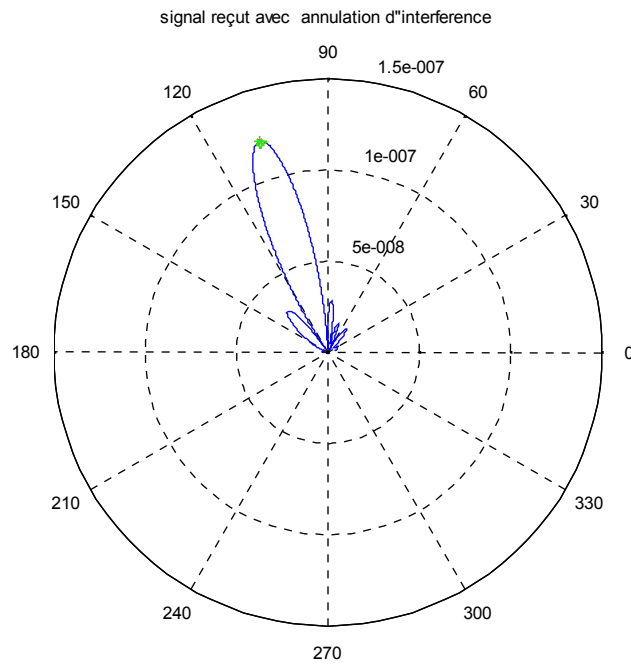


Figure.4.41- tracé des lobes pour une adaptation LMS.

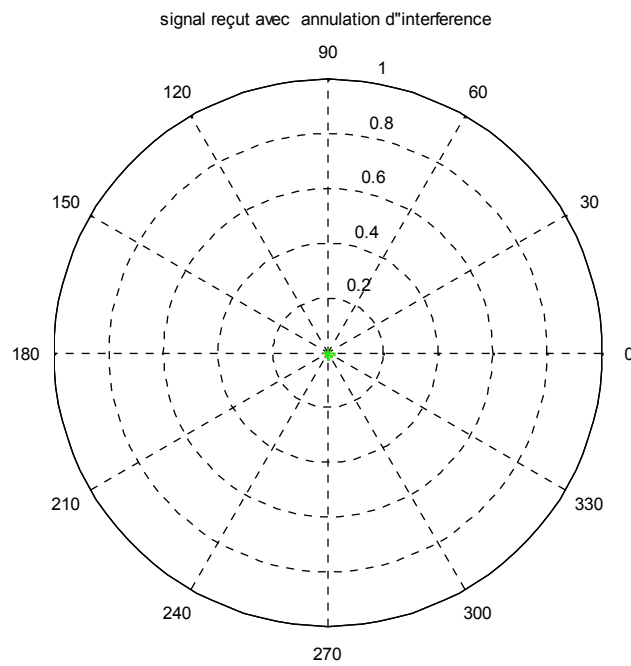


Figure.4.42- tracé des lobes pour une adaptation LMS.

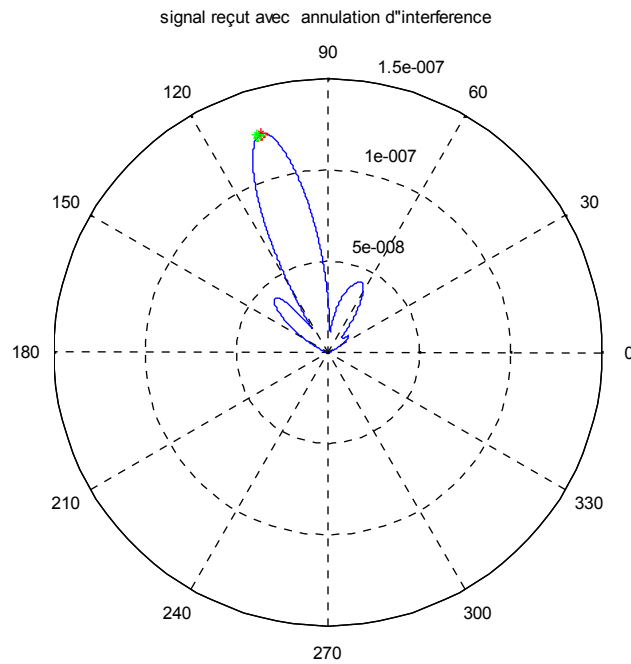


Figure.4.43- tracé des lobes pour une adaptation RLS a facteur d'oublie constant.

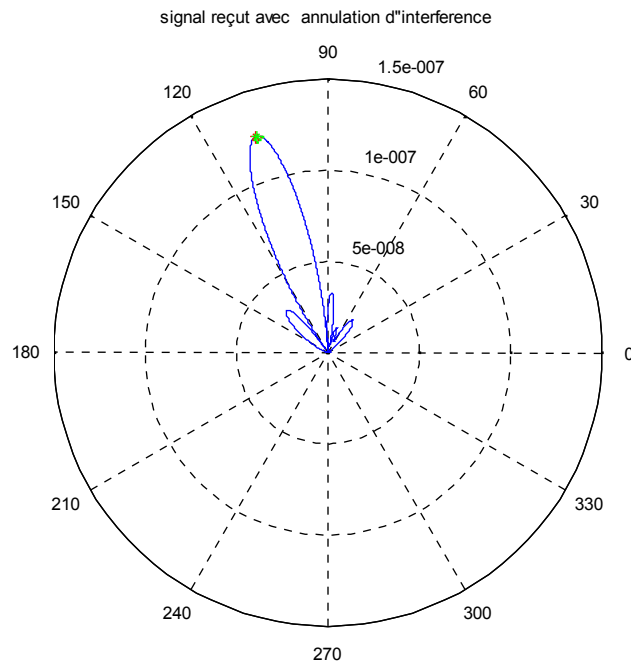


Figure.4.44- tracé des lobes pour une adaptation RLS a facteur d'oublie variable.

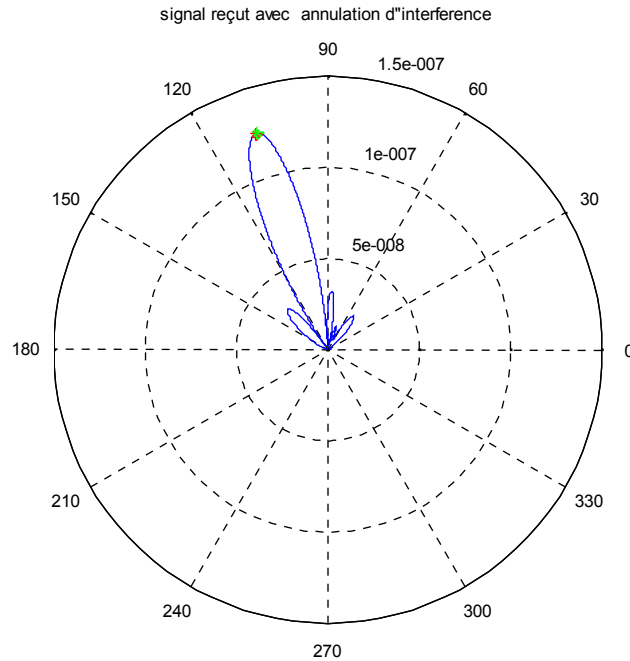


Figure.4.45- tracé des lobes pour une adaptation RLS a trace constante.

algorithme	LMS a pas constant	RLS sans facteur d'oublie	RLS a facteur d'oublie constant	RLS a facteur d'oublie variable	RLS a trace de la matrice d'autocorrelation constante
Angle vrai (°)	18	18	18	18	18
Angle estimé (°)	17.9800	----	16.9200	18.3200	18.4000

Tableau.4.2-valeurs des angles vrais et estimés pour la simulation n°02

Commentaires :

Dans les figure de 4.41 a 4.45, on montre le tracé des lobes pour chaque algorithme a l'échantillon 400, avec les cibles vrais (vert) et estimé (rouge). et le **tableau 4.2** nous donne les valeurs exacte de chaque estimation.

4.4.2. Simulation n°02 :

4.4.2. 1. Partie 1 : Limite de la poursuite des signaux par l’LMS et l’RLS a facteur d’oublie variable en convergence et stabilité

Initialisation n°02(partie1) :

Nombre d’antenne	8
Nombre d’information (itération /4)	10
Angle d’arrivée du signal désiré (°)	36
Angle d’arrivée d’interférence 1 (°)	72
Angle d’arrivée d’interférence 1 (°)	-72

Tableau.4.3-initialisation de la simulation n°02(partie1)

Résultat n°02(partie1) :

Poursuite de la phase :

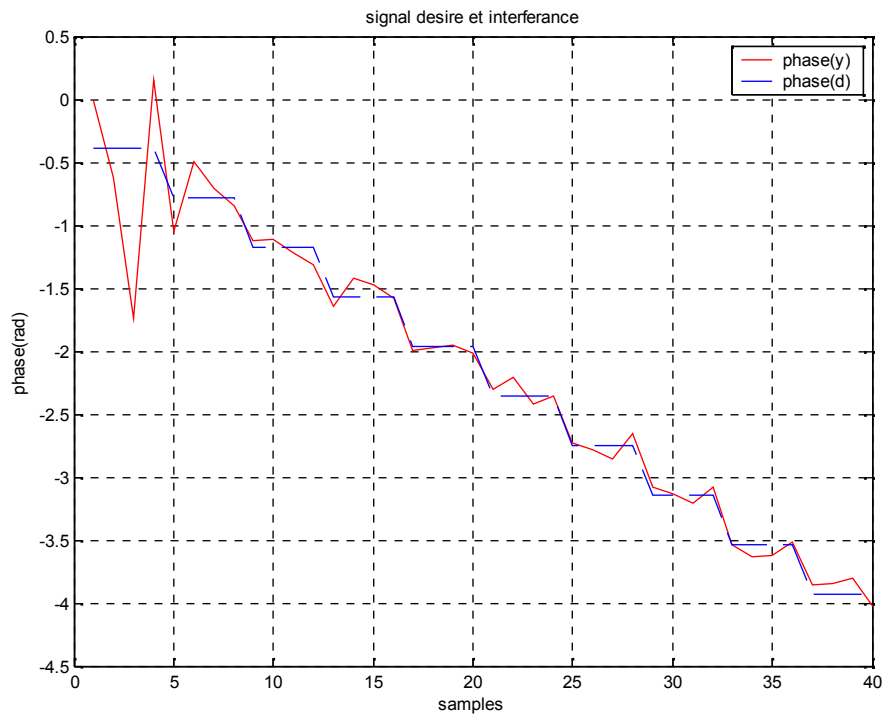


Figure.4.46- poursuite de la phase par LMS.

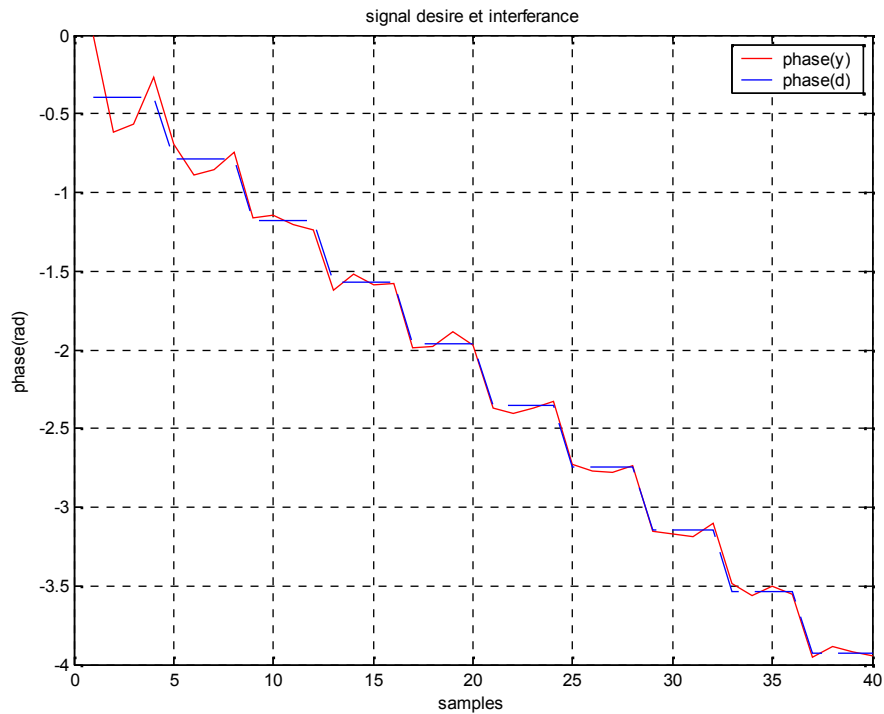


Figure.4.47- poursuite de la phase par RLS a facteur d'oubli variable.

Poursuite de la magnitude :

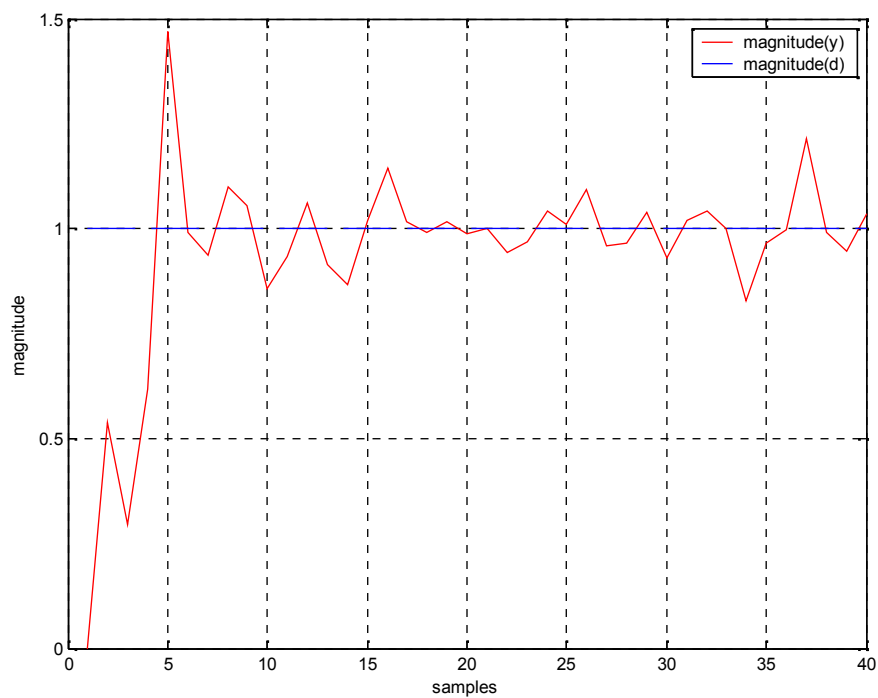


Figure.4.48- poursuite de la magnitude par LMS.

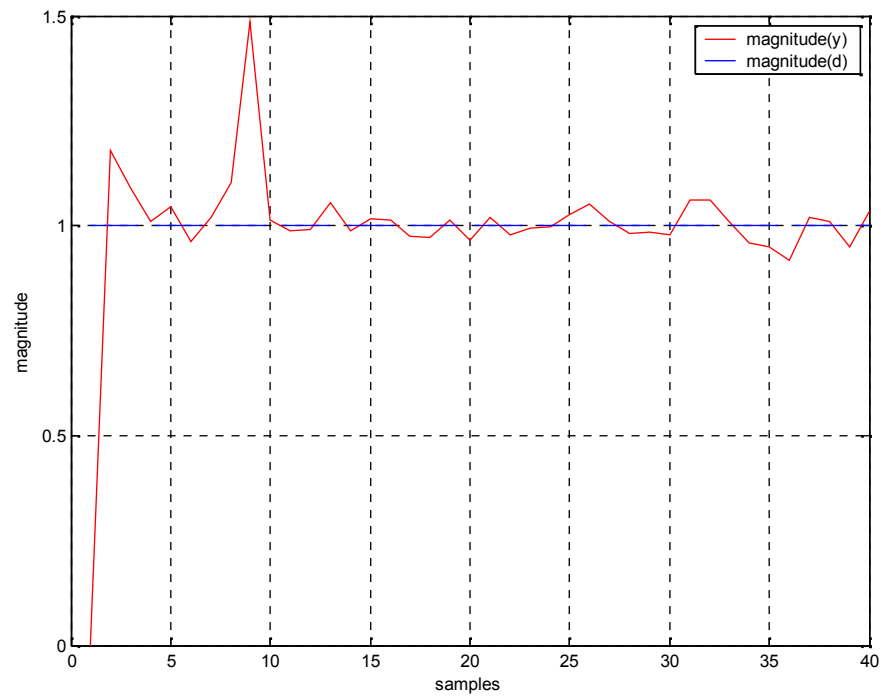


Figure.4.49- poursuite de la magnitude par RLS a facteur d'oublie variable.

Evolution de l'erreur :

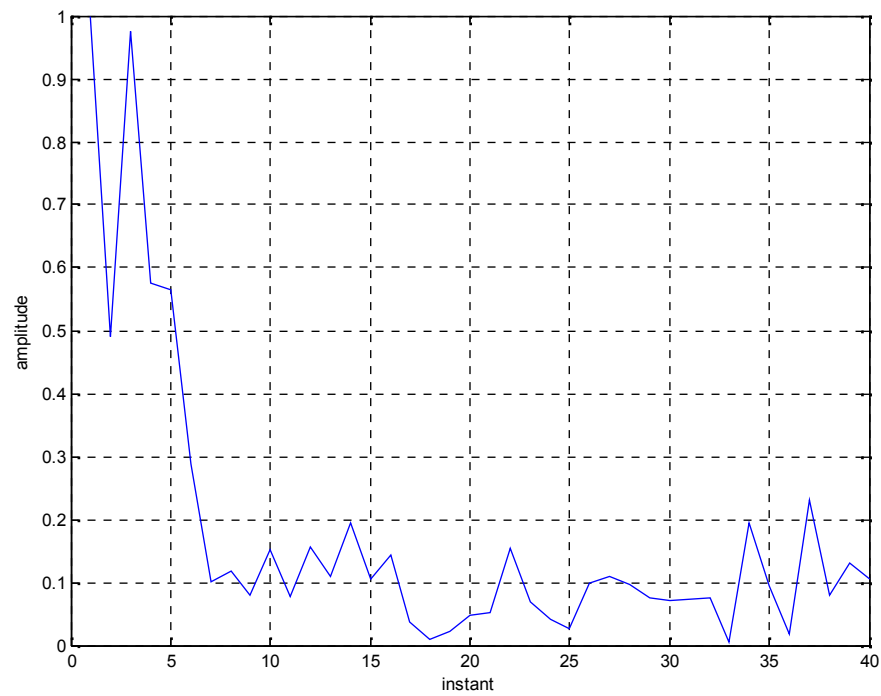


Figure.4.50- évolution de l'erreur par LMS.

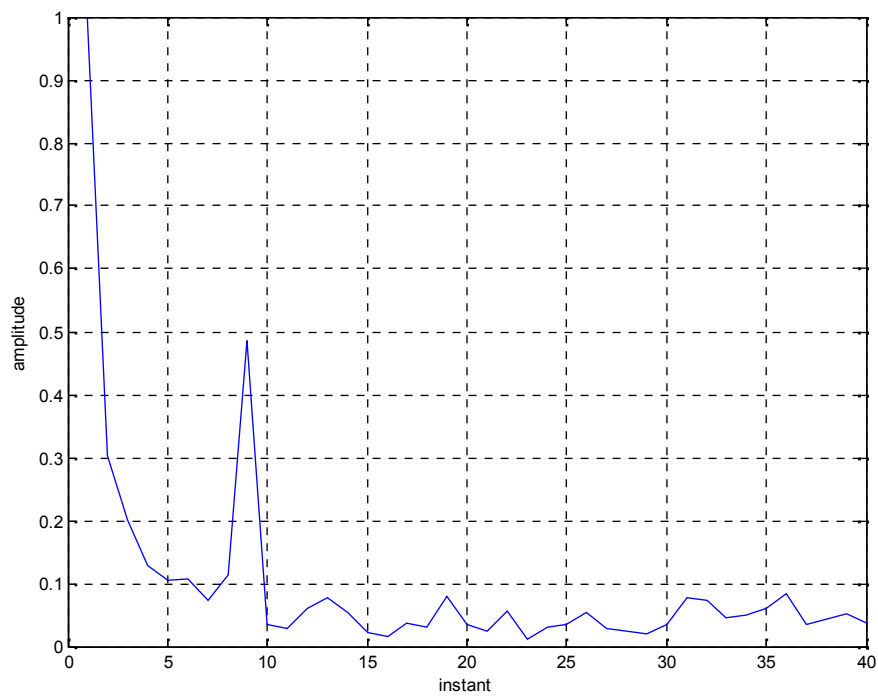


Figure.4.51- évolution de l'erreur par RLS a facteur d'oublie variable.

Evolution des coefficient complexe des filtres (convergence, stabilité) :

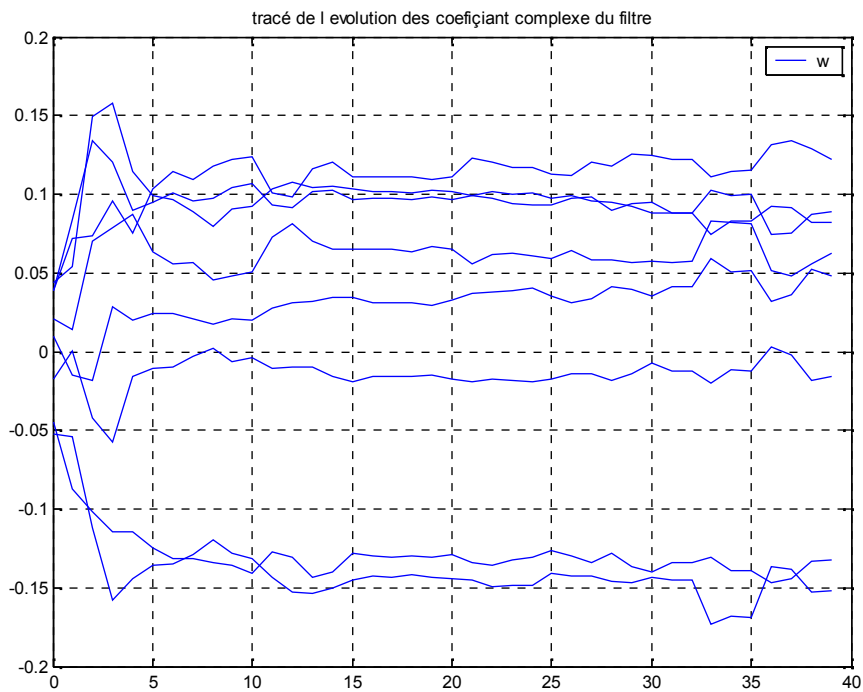


Figure.4.52- évolution des coefficient complexe du filtre LMS .

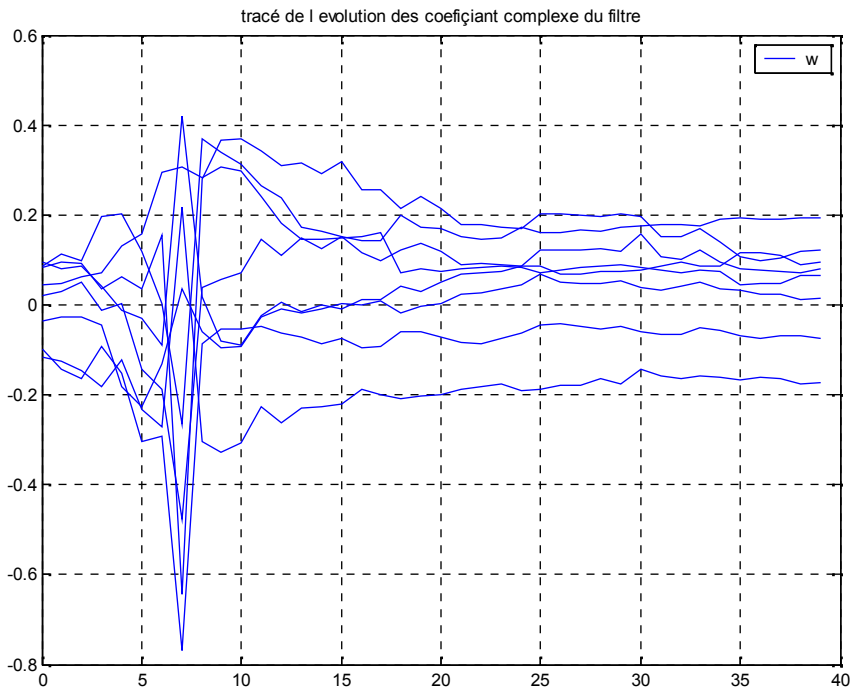


Figure.4.53- évolution des coeffçiant complexe du filtre RLS a facteur d'oublie variable.

Tracé des lobes et des angles vrais et estimés (vert, rouge):

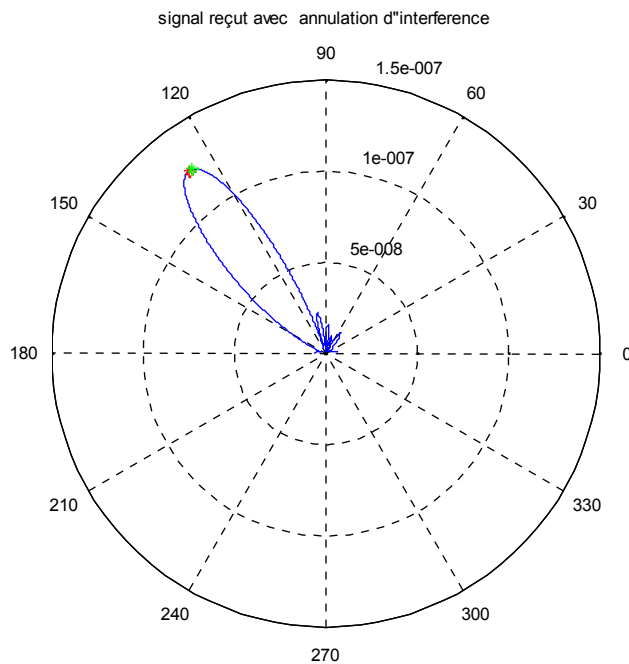


Figure.4.54- tracé des lobes pour une adaptation LMS.

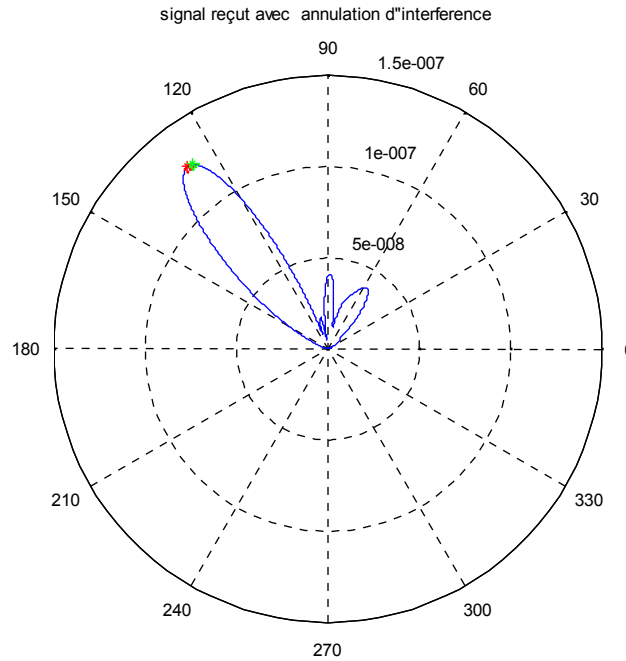


Figure.4.55- tracé des lobes pour une adaptation RLS a facteur d'oublie variable.

algorithme	LMS a pas constant	RLS a facteur d'oublie variable
Angle vrai (°)	36	36
Angle estimé (°)	36.8600	37.5800

Tableau.4.4-valeurs des angles vrais et estimés pour la simulation n°02(partie1).

Commentaires :

Cette simulation nous montre la limite du LMS à converger et poursuivre le signal désiré pour un nombre d'échantillon limité car c'est le facteur le plus influant dans le calcul LMS, pour remédier à cette contrainte, on a dans le RLS le facteur le plus influant est le nombre de capteur et il suffit d'avoir un nombre d'échantillon supérieur à ce nombre de capteur pour avoir une bonne poursuite et stabilité.

4.4.2. 2. Partie 2 : Limite de la poursuite des signaux par l’LMS et l’RLS a facteur d’oublie variable en robustesse aux interférences

Initialisation n°02(partie2) :

Nombre d’antenne	15
Nombre d’information (itération /4)	10
Angle d’arrivée du signal désiré (°)	36
Angle d’arrivée d’interférence 1 (°)	72
Angle d’arrivée d’interférence 1 (°)	29

Tableau.4.5-initialisation de la simulation n°02 (partie2).

Résultat n°02(partie2) :

Poursuite de la phase :

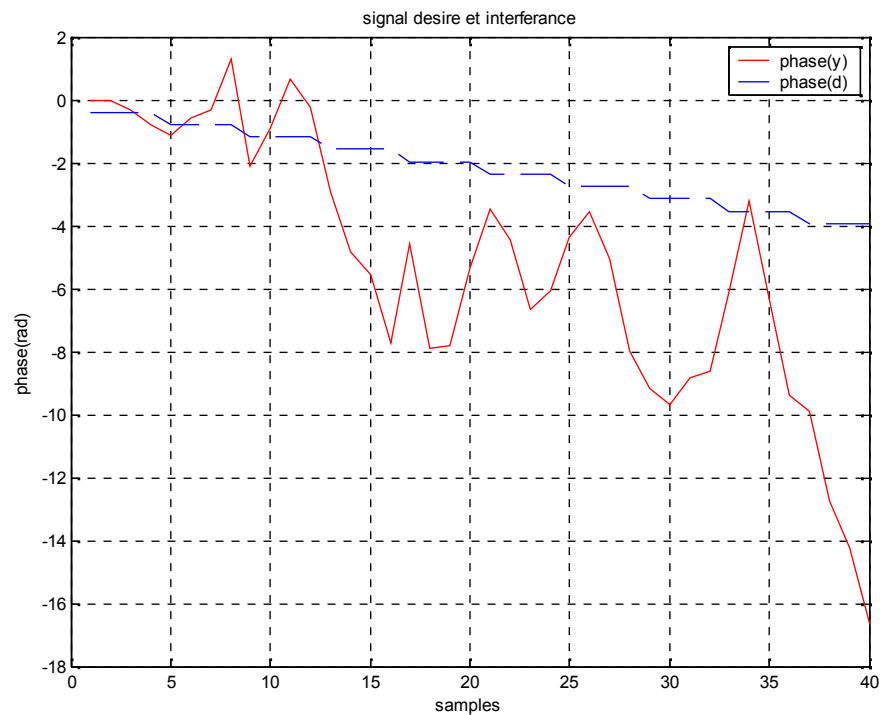


Figure.4.56- poursuite de la phase par LMS.

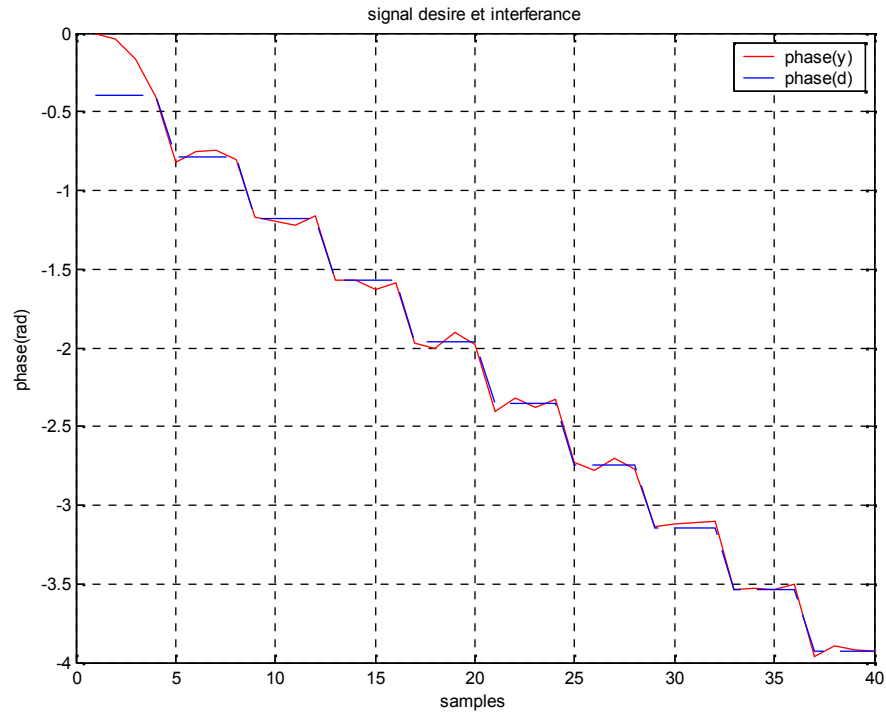


Figure.4.57- poursuite de la phase par RLS a facteur d'oublie variable.

Evolution de l'erreur quadratique :

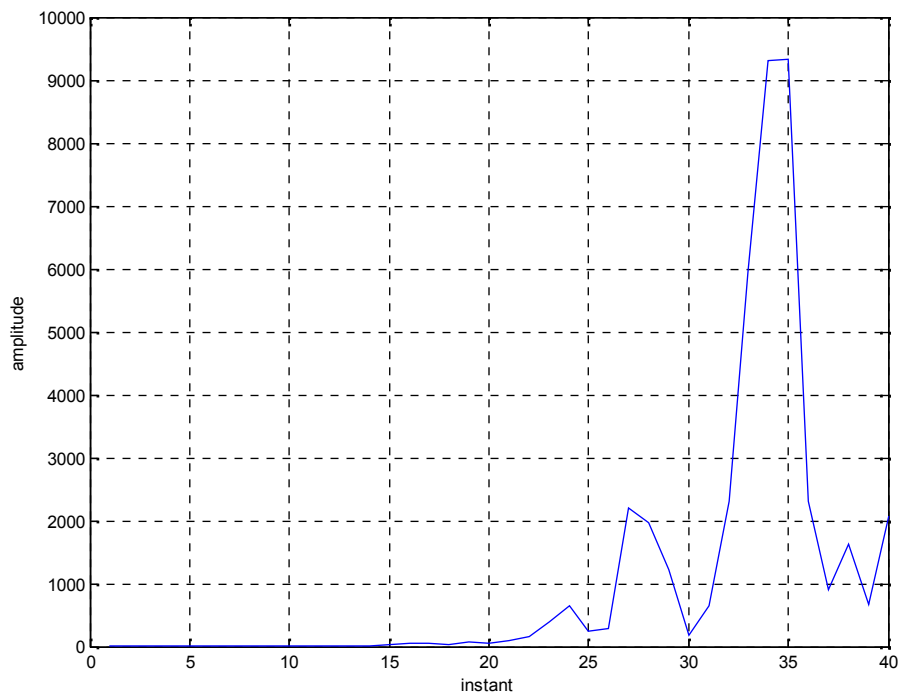


Figure.4.58- évolution de l'erreur par LMS.

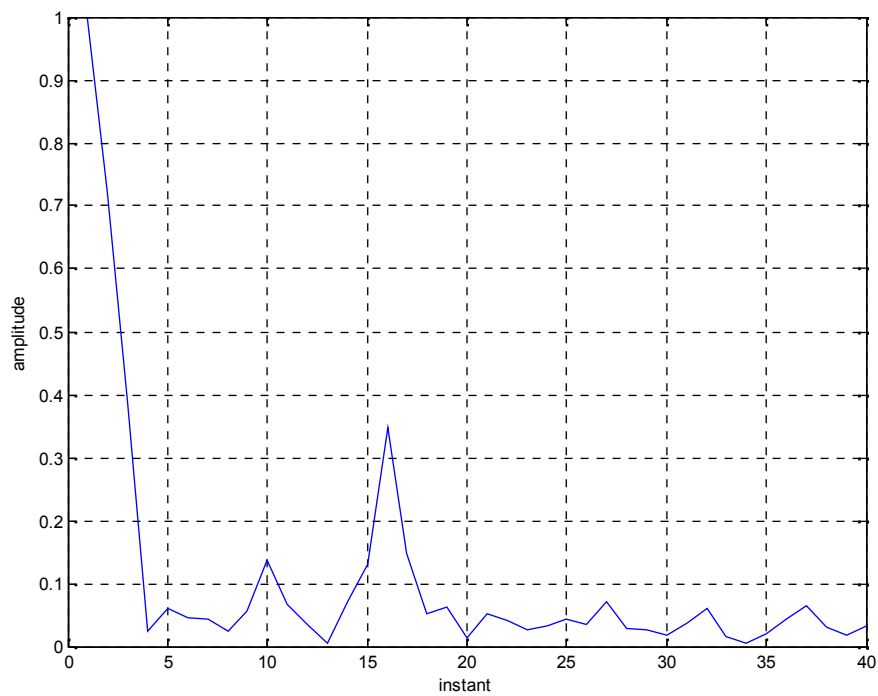


Figure.4.59- évolution de l'erreur par RLS a facteur d'oubli variable.

Tracé des lobes et des angles vrais et estimés (vert, rouge) :

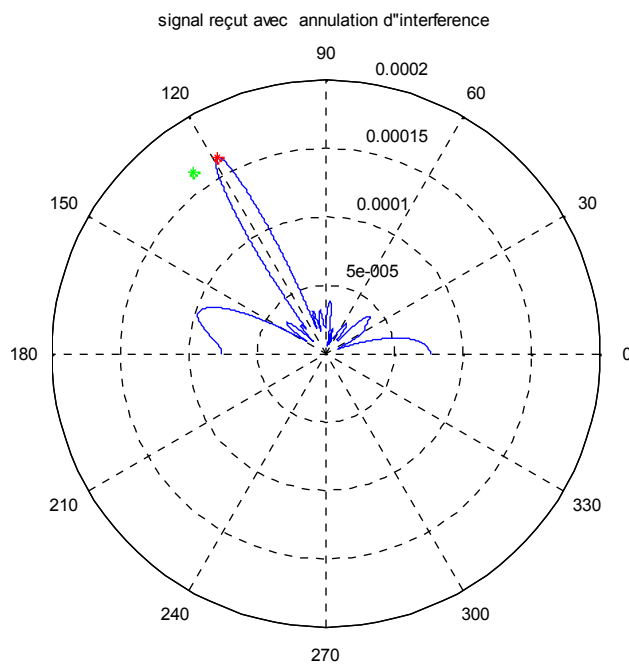


Figure.4.60- tracé des lobes pour une adaptation LMS.

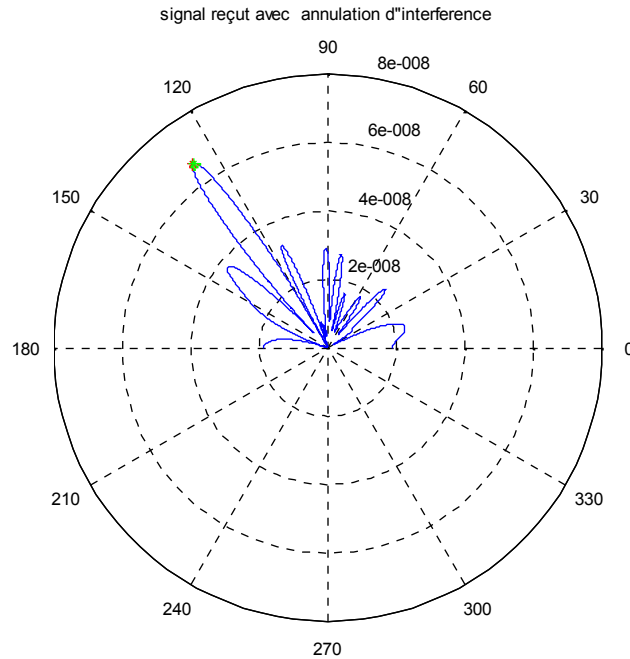


Figure.4.61- tracé des lobes pour une adaptation RLS a facteur d'oublie variable.

algorithme	LMS a pas constant	RLS a facteur d'oublie variable
Angle vrai (°)	36	36
Angle estimé (°)	29.0600	36.1400

Tableau.4.6-valeurs des angles vrais et estimés pour la simulation n°02(partie2).

Commentaires :

Dans cette simulation on a montré l'incapacité du LMS à éliminer un bruit non désiré (interférence) proche a la direction désirée, avec un nombre limité d'echontillon, ce qui explique la perte de le mauvais accrochage du signal et l'erreur d'estimation par rapport au simulation précédente, pour le RLS il a suffit d'augmenté le nombre d'antenne pour remédier a ce problème.

4.5. Simulation et commentaires pour AOA (DOA) variable :

Dans cette partie on simule Beamforming adaptatif pour un angle d'arrivée variable, On variant a chaque fois les angle d'arrivée du départ des interférence, le nombre d'itération, le nombre d'antenne.



Figure 4.62 - Initialisation des données pour AOA variable.

Et voir l'évolution de l'adaptation beamformer par l'algorithme :

- RLS de trace constante.

4.5. 1. Simulation n°03 : Poursuite de la variation de l'angle d'arriver d'une cible par filtrage adaptatif beamformer basé sur RLS a facteur d'oublie de trace constante

4.5. 1. 1. Partie1 : interférence situer en dehors de l'intervalle de variation de la AOA

Initialisation n°03(partie1) :

Nombre de variation de l'AOA	5
Nombre d'antenne	8
Nombre d'information (itération /4)	50
Angle d'arrivée de départ du signal désiré (°)	16
Angle d'arrivée d'interférence 1 (°)	86
Angle d'arrivée d'interférence 1 (°)	-52

Tableau.4.7-initialisation de la simulation n°03(partie1).

Résultats n°03(partie1) :

Tracé du balayage des lobes :

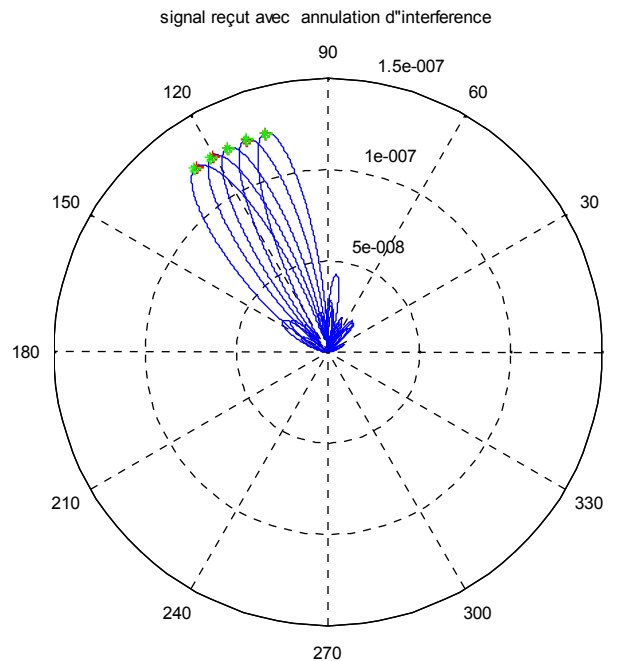


Figure.4.63- tracé super posé du balayage des lobes pour une adaptation RLS a trace constante.

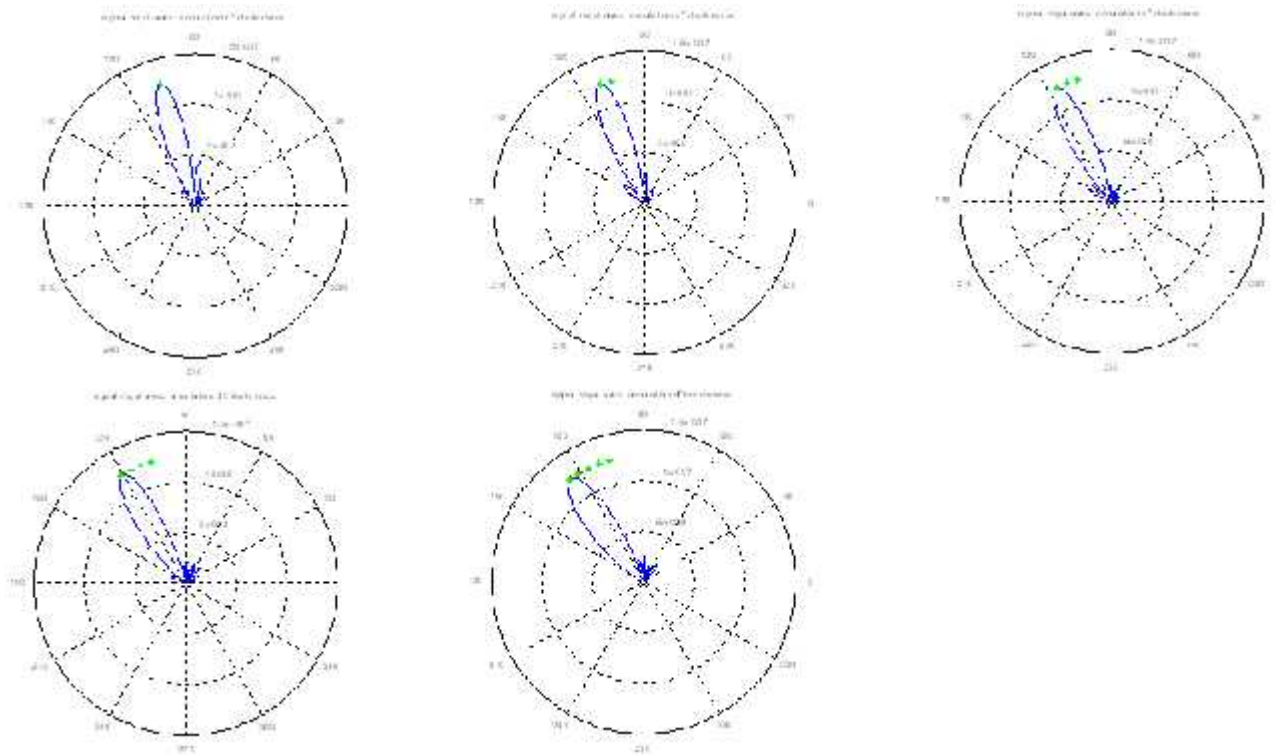


Figure.4.64- tracé du balayage des lobes a chaque variation pour une adaptation RLS a trace constante.

	Départ	Variation 1	Variation 2	Variation 3	Variation 4
Angle estimé	15.9200	21.1600	26.2600	30.4400	35.3800
Angle vrai	16.0000	21.0000	26.0000	31.0000	36.0000

Tableau.4.8-valeurs des angles vrais et estimés pour la simulation n°03(partie1).

Commentaires :

Dans cette simulation ou les interférences sont situés en dehors de l'intervalle de la variation de la AOA, on a eu une bonne poursuite de la cible par le lobe de l'antenne intelligente, et peu de lobes secondaire.

4.5. 1. 1. **Partie2** : interférence situer dans l'intervalle de variation de la AOA

Initialisation n°03(partie2) :

Nombre de variation de l'AOA	5
Nombre d'antenne	8
Nombre d'information (itération /4)	50
Angle d'arrivée de départ du signal désiré (°)	18
Angle d'arrivée d'interférence 1 (°)	22
Angle d'arrivée d'interférence 1 (°)	31

Tableau.4.9-initialisation de la simulation n°03(partie2).

Résultats n°03(partie2) :

Tracé du balayage des lobes :

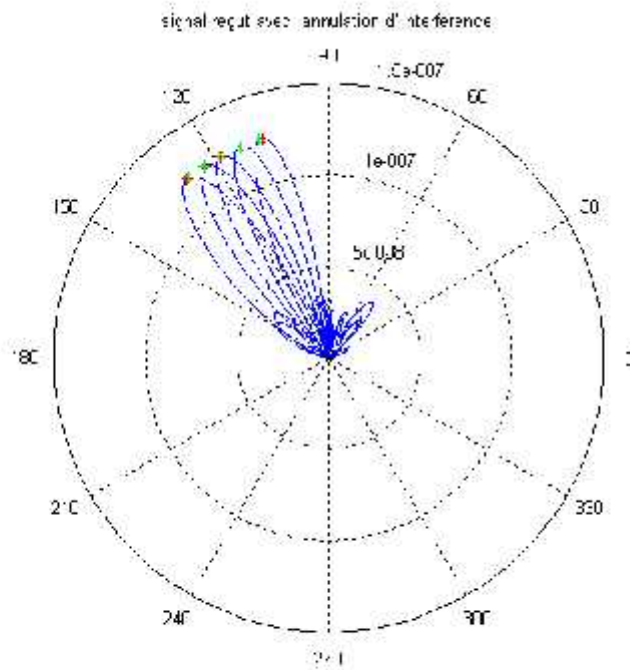


Figure.4.65- tracé super poser du balayage des lobes pour une adaptation RLS a trace constante.

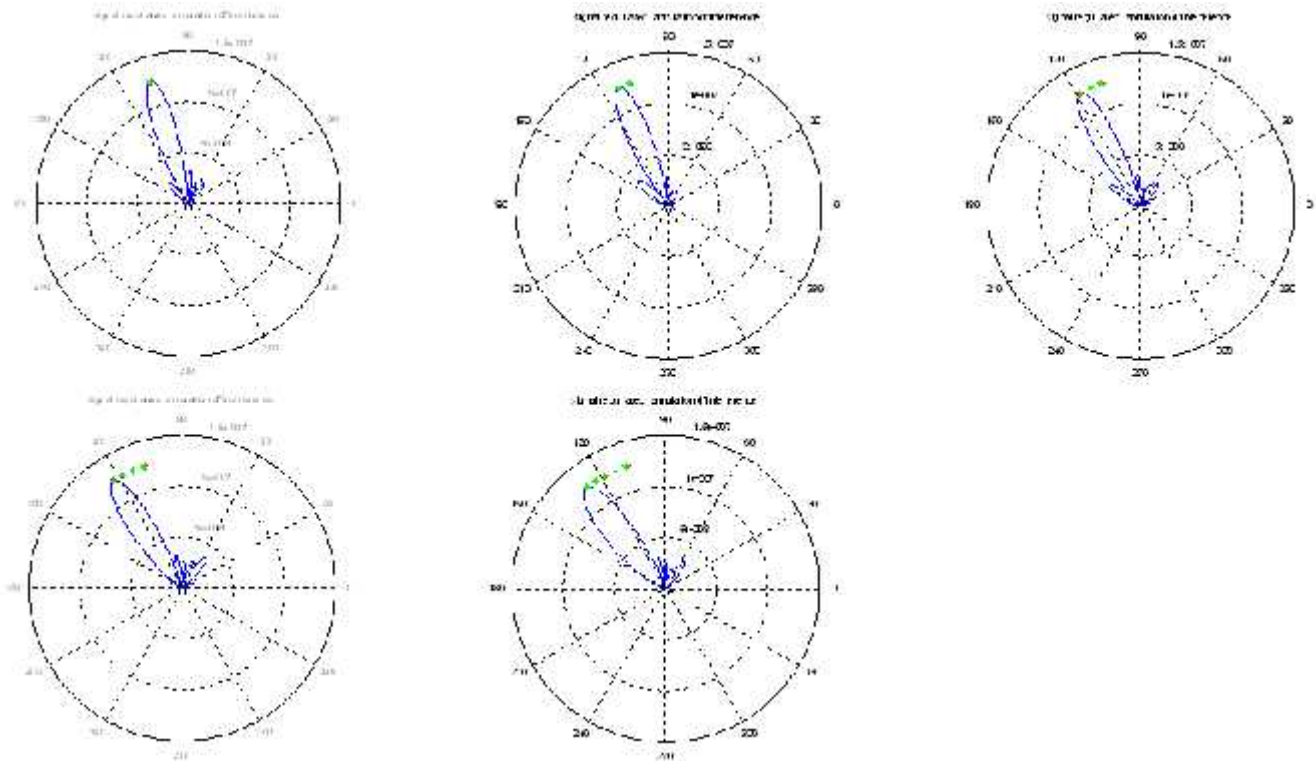


Figure.4.66- tracé du balayage des lobes a chaque variation pour une adaptation RLS a trace constante.

	Départ	Variation 1	Variation 2	Variation 3	Variation 4
Angle estimé	17.2000	22.9200	28.3000	32.8600	38.4200
Angle vrai	18	23	28	33	38

Tableau.4.10-valeurs des angles vrais et estimés pour la simulation n°03(partie2).

Commentaires :

Dans cette simulation ou les interférences sont situées dans l’intervalle de la variation de la AOA, on a une bonne poursuite de la cible par le lobe de l’antenne intelligente.

4.6. Conclusion :

Au bout de notre simulation on a pue :

- 1-Montrer l’évolution des différents paramètres pour chaque algorithme.
- 2-Montrer que le LMS nécessite un nombre d’échantillon élevé pour assuré la poursuite des signaux,ou une variation de son pas d’adaptation.
- 3-Le RLS sans facteur d’oublie ne peut pas assuré la poursuite car il n y a pas d’action sur sa mémoire.
- 4-Le RLS a facteur d’oublie constant donne une solution acceptable, mais pas aussi bonne que dans le RLS a facteur d’oublie variable, car on fixe l’action sur la mémoire a une valeur constante.
- 5-Montrer que le RLS à trace constante de la matrice d’autocorrelation et le RLS a facteur d’oublie variable, assure une très bonne poursuite du signal et de la variation de variation de la AOA avec une stabilité et rapidité de convergence admirable.
- 6-Appliquer ces algorithmes à la rangée d’antenne linéaire et aboutir a une antenne intelligente, qui assure de façon autonome la poursuite de la cible par son lobe.

2.1.3. Choix de l'algorithme :

Le choix de l'algorithme se fera en fonction des critères suivants :

- La rapidité de convergence qui sera le nombre d'itérations nécessaires pour converger « assez près » de la solution optimale de Wiener dans le cas stationnaire.
- La mesure de cette « proximité » entre cette solution optimale et la solution obtenue.
- La capacité de poursuite (tracking) des variations (non stationnarités) du processus. On examinera quels sont les algorithmes vraiment adaptatifs.
- La robustesse au bruit
- La complexité (en nombre de MIPS)
- La structure (se prêtant plus ou moins à l'implémentation en VLSI)
- Les propriétés numériques (stabilité – précision) dans le cas d'une précision limitée sur les données et les coefficients (problèmes d'implémentation en virgule fixe).

Nous ne nous intéresserons dans le cadre de cette thèse qu'aux trois premiers critères de choix.

Remarques :

1. La théorie des filtres incluant celle des filtres de Wiener et Kalman a été développée pour des signaux à temps continu et à temps discret mais nous considérerons par la suite uniquement le cas discret où les signaux sont échantillonnés et les filtres sont numériques. Nous nous limiterons de plus aux filtres à réponse impulsionnelle finie (FIR)

2. Le début de cette thèse est écrit pour des signaux complexes. Les formules obtenues étant un peu lourdes, la suite du cours est écrite pour des signaux réels. Vous pouvez donc, lors de la lecture, ignorer les conjugaisons $()^*$ et translater les conjuguées hermitiennes $()^+$ ou $()^H$ en transposées simples $()^T$.

2.2. Filtrage linéaire optimal Filtrage de Weiner :

Considérons la figure suivante :

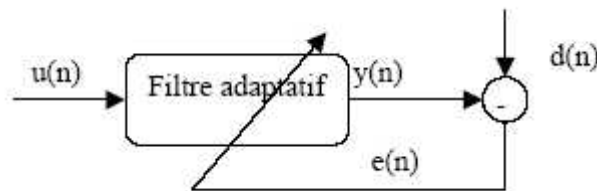


Figure.2.1- filtrage optimal.

Le problème du filtrage optimal de trouver le « meilleur » filtre c'est à dire celui permettant d'obtenir en sortie une réponse $y(n)$ la plus « proche » possible d'une réponse désirée $d(n)$ lorsque l'entrée est une certaine séquence $u(n)$, On note :

$$e(n) = y(n) - d(n) \quad (2.1)$$

L'erreur entre la réponse désirée $d(n)$ et la sortie $y(n)$. On note également $w(n)$ la réponse pulsionnelle du filtre. On note :

$$y(n) = \hat{d}(n / \underline{X}_n)$$

Où \underline{X}_n est un vecteur d'état contenant toute l'information utilisée pour prédire $d(n)$:

$$\underline{X}_n = [u(0), \dots, u(n), e(0), \dots, e(n-1), \dots, w(0), \dots, w(M-1), y(0), \dots, y(n-1)] \quad (2.2)$$

Le problème consiste donc à rechercher le filtre assurant l'erreur la plus faible $e(n)$, au sens d'une certaine fonction de coût :

$$w = \arg \min J(e(n)) \quad (2.3)$$

De nombreux choix sont possibles en ce qui concerne la fonction de coût, l'erreur quadratique moyenne est la plus utilisée, car elle conduit à des développements mathématiques complets et simples, fournit la solution en fonction des caractéristiques au second ordre des variables aléatoires, caractéristiques qui sont les plus simples à estimer, et enfin fournit une solution unique. C'est sur l'estimation linéaire en moyenne quadratique que repose le filtrage de Wiener.

2.2.1. Relations d'orthogonalité - Equation de Wiener-Hopf :

La sortie du filtre s'écrit :

$$y(n) = \sum_{k=0}^{M-1} w^* u(n-k) \quad (2.4)$$

Et l'erreur : $e(n) = y(n) - d(n)$

Le filtre de Wiener est celui qui minimise l'erreur quadratique moyenne (**EQM** ou **MSE** anglais) :

$$J = E\{(e(n))^2\} \quad (2.5)$$

En introduisant les vecteurs :

$$\underline{w}^T = [w_0, \dots, w_{M-1}] \text{ et } \underline{u}^T(n) = [u(n), u(n-1), \dots, u(n-M+1)]$$

On a alors :

$$e(n) = d(n) - \underline{w}^H \underline{u}(n) = d(n) - \underline{u}^T(n) \underline{w}^* \quad (2.6)$$

D'où :

$$\begin{aligned} J &= E\{(d(n) - \underline{w}^H \underline{u}(n))(d^*(n) - \underline{w}^T \underline{u}^*(n))\} \\ &= E\{d(n)^2\} - \underline{w}^H E\{\underline{u}(n) d^*(n)\} - \underline{w}^T E\{\underline{u}^*(n) d(n)\} + \underline{w}^H E\{\underline{u}(n) \underline{u}^*(n)\} \underline{w} \end{aligned}$$

Soit :

$$J = \dagger_d^2 - \underline{w}^H \underline{R}_{ud} - \underline{w}^T \underline{R}_{ud}^* + \underline{w}^H \underline{R}_{uu} \underline{w} \quad (2.7)$$

Avec \underline{R}_{uu} qui est la matrice d'autocorrélation de l'entrée $u(n)$. Cette matrice est définie positive, de Toeplitz et à symétrie hermitienne : $\underline{R}_{uu} = \underline{R}_{uu}^H$, on aboutie au même résultats par le théorème projection (**ANNEXE 2**).

Et $\underline{R}_{ud} = E\{\underline{u}(n) d^*(n)\}$ le vecteur d'intercorrélacion entre la sortie désirée $d(n)$ et l'entrée $u(n)$.

Le vecteur optimum \underline{w} est celui qui annule le gradient du critère :

$$\frac{dJ}{d\underline{w}} = \underline{0} \tag{2.8}$$

En écrivant J sous la forme $J = E \{ e_n \underline{e}_n^* \}$ on a :

$$\frac{dJ}{d\underline{w}} = E \left\{ e_n \frac{de_n^*}{d\underline{w}} \right\} + E \left\{ e_n^* \frac{de_n}{d\underline{w}} \right\} \tag{2.9}$$

Or le gradient vectoriel complexe (ANNEXE 3). nous donne :

$$\frac{de_n^*}{d\underline{w}} = \underline{0} \quad \text{et} \quad \frac{de_n}{d\underline{w}} = 2\underline{u}(n)$$

Par conséquent, à l'optimum, on a :

$$E \{ e_n^* \underline{u}(n) \} = \underline{0} \tag{2.10}$$

C'est le principe d'orthogonalité signifiant que toutes les entrées $u(n)$ sont décorrélées de e_n^* .

En développant cette dernière équation on obtient :

$$E \left\{ \underline{u}(n) \left(d^*(n) - \underline{u}^H(n) \underline{w} \right) \right\}$$

Alors on a :

$$\underline{R}_{uu} \underline{w} = \underline{R}_{ud} \tag{2.11}$$

Cette relation est appelée Formule de Wiener ou équation de Wiener Hopf. La solution est le filtre optimal :

$$\underline{w} = \left(\underline{R}_{uu} \right)^{-1} \underline{R}_{ud} \tag{2.12}$$

Pour ce vecteur optimal on obtient à partir de (2.7) et (2.11) l'Erreur Quadratique Moyenne minimale (Annexe 3):

$$J_{\min} = \sigma_d^2 - \underline{w}^H \underline{R}_{ud} = \sigma_d^2 - \sigma_{\hat{d}}^2 \tag{2.13}$$

On note :

$$\hat{d} = \underline{w}^H \underline{u}(n) \text{ Et } \sigma_{\hat{d}}^2$$

Signal filtré optimal et la variance de ce signal.

Cette relation montre que pour le filtre optimal, l'EQM est la différence entre la variance du signal désiré et celle de l'estimée de ce signal produite par le filtre.

2.2.2. Application :

Le filtrage de Wiener adaptatif est un outil puissant en traitement du signal, communications numériques et contrôle automatique. Les applications sont diverses mais présentent toujours les caractéristiques exposées plus haut : on dispose d'une entrée u ainsi que de la réponse désirée d et l'erreur e entre la sortie y et d sert à contrôler (adapter) les valeurs des coefficients du filtre w . Ce qui différencie essentiellement les applications provient de la façon de définir la réponse désirée d .

On peut distinguer quatre grandes classes d'applications :

- L'identification de systèmes
- La prédiction
- La modélisation inverse
- L'annulation d'interférences

Nous donnons ci après les schémas correspondant à ces quatre classes :

2.2.2.1. Identification de Systèmes :

$d(n)$ Est la sortie du système que l'on souhaite identifier

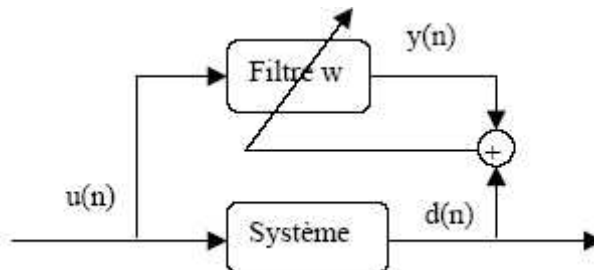


Figure.2.2-Identification de Systèmes.

2.2.2.2. Prédiction :

$d(n)$ Est le signal à l'instant n et $y(n)$ le signal prédit à partir du signal aux instants précédents.

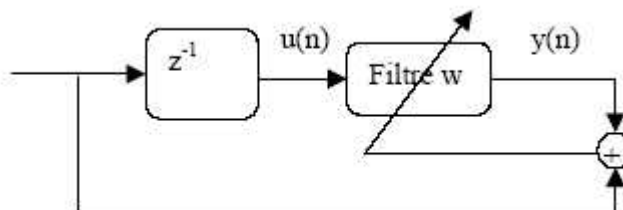


Figure.2.3-Prédiction.

2.2.2.3. Modélisation inverse (égalisation, déconvolution) :

$d(n)$ Est l'entrée (retardée) du système que l'on cherche à « inverser ».

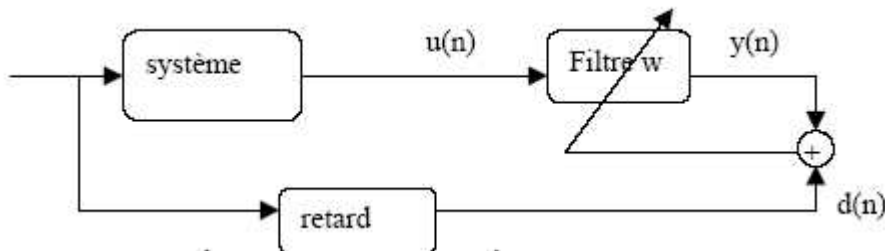


Figure.2.4-Modélisation inverse.

2.2.2.4. Annulation d'interférences (annulation d'écho, de bruit) :

$d(n)$ Est un signal primaire qui comporte les interférences à annuler. $u(n)$ Est le signal de référence dénué (ou presque) d'information et obtenu par un capteur proche de celui qui fournit $d(n)$.

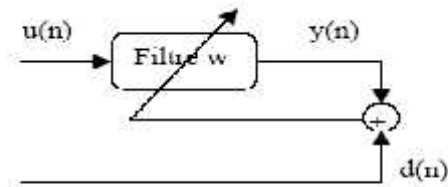


Figure.2.5-Annulation d'interférences.

2.3. Algorithme du gradient :

L'équation de Wiener Hopf (2.11) qui permet de calculer le filtre de Wiener optimal conduit à résoudre un système de M équations à M inconnues : $\underline{R}_{uu} \underline{w} = \underline{R}_{ud}$

Une méthode d'inversion directe est en $O(M^3)$, ce qui est très coûteux. Il peut être préférable de résoudre ce système par une méthode itérative, notamment en se souvenant que la fonction de coût est quadratique, ce qui entraîne que le minimum est unique.

La méthode du gradient consiste à :

- Choisir un vecteur initial de coefficients $\underline{w}(0)$.
- Ayant un vecteur candidat $\underline{w}(n)$ à l'étape n, obtenir un meilleur candidat en incrémentant $\underline{w}(n)$ dans la direction opposée au gradient du coût J.

$$\underline{w}(n+1) = \underline{w}(n) - \frac{1}{2} \tilde{\mu}_n \underline{\nabla} J \Big|_{\underline{w}=\underline{w}(n)} \tag{2.14}$$

La séquence $\{\tilde{\mu}_n\}$ est une série de coefficients appelés pas d'adaptation.

On distingue l'algorithme à pas d'adaptation constant $\tilde{\mu}_n = \tilde{\mu}$ et les algorithmes à pas décroissant.

Le gradient est égal à :

$$\underline{\nabla} J \Big|_{\underline{w}=\underline{w}(n)} = 2 \underline{R}_{uu} \underline{w} - 2 \underline{R}_{ud} \tag{2.15}$$

La forme explicite de l'algorithme est donc :

$$\underline{w}(n+1) = \underline{w}(n) - \tilde{\mu}_n \left(\underline{R}_{uu} \underline{w} - \underline{R}_{ud} \right) \tag{2.16}$$

On note :

$$v(n) = d(n) - \hat{d}(n)$$

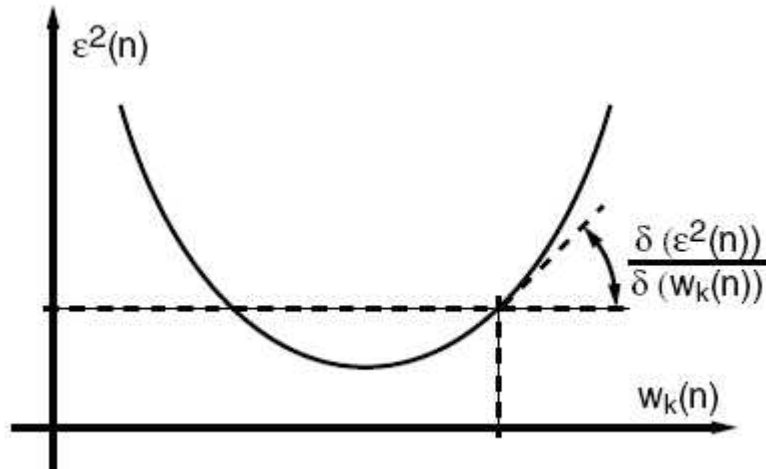


Figure 2.6-Derivee de l'erreur quadratique.

Notons que cette forme est déterministe et nécessite la connaissance des grandeurs statistiques et Le paramètre μ_n contrôle l'importance de la correction apportée au vecteur de coefficients \underline{w} lors du n itération. Le paragraphe suivant porte sur l'étude de la convergence de l'algorithme c'est à dire sur son aptitude à être stable et à tendre vers la solution optimale. Les deux facteurs influençant à priori cette convergence sont le pas d'adaptation μ_n et la matrice d'autocorrélation.

2.3.1. Convergence de l'algorithme du gradient :

Nous commençons en étudiant le comportement du vecteur d'erreur entre le vecteur optimal \underline{w} et le vecteur déterminé \underline{w} à la n itération de l'algorithme.

On note :

$$\underline{v}(n) = \underline{w}(n) - \underline{w}$$

A partir des équations (2.13) et (2.16), on obtient :

$$\underline{w}(n+1) = \underline{w}(n) - \mu_n \left(\underline{R}_{\underline{w}} \left[\underline{w} - \underline{w}(n) \right] \right) \tag{2.17}$$

D'où :

$$\underline{v}(n+1) = \left(I - \mu_n \underline{R}_{\underline{w}} \right) \underline{v}(n) \tag{2.18}$$

Introduisons maintenant la décomposition propre de la matrice d'autocorrélation :

$$\underline{R}_{\underline{w}} = \underline{Q} \underline{P} \underline{Q}^H,$$

Où \underline{Q} est unitaire.

Soit en multipliant les membres de l'équation (2.18) par \underline{Q} et en notant $\underline{r}(n) = \underline{Q}^H \underline{v}(n)$:

$$\underline{r}(n+1) = \left(I - \mu_n \underline{P} \right) \underline{r}(n) \tag{2.19}$$

De cette façon, on fait apparaître les modes propres de l'algorithme.

L'algorithme converge si $\underline{r}(n) \rightarrow 0$, ce qui entraîne bien que $\underline{v}(n) \rightarrow 0$, soit $\underline{w}(n) \rightarrow \underline{w}$.

Le problème est donc de donner les conditions qui assurent cette convergence, de quantifier la rapidité de convergence, et de choisir le pas d'adaptation.

2.3.1.1. Conditions de convergence pour un pas μ constant :

L'équation (2.19) nous donne par récurrence :

$$\underline{r}(n) = (I - \mu \underline{P})^n \underline{r}(0) \tag{2.20}$$

Chacune des composantes s'exprime alors sous la forme :

$$r_k(n) = (1 - \mu \lambda_k)^n r_k(0) \tag{2.21}$$

Où λ_k est la k^{ième} valeur propre.

Les différentes composantes convergent à 0 pourvu que nous en déduisons alors la condition de convergence de l'algorithme du gradient :

$$|1 - \mu \lambda_k| < 1, \forall k \in \{0, 1, \dots, M-1\} \tag{2.22}$$

On notant λ_{\max} la valeur propre maximale :

$$0 < \mu < \frac{2}{\lambda_{\max}} \tag{2.23}$$

2.3.1.2. Rapidité de convergence :

Nous nous intéressons maintenant à la vitesse de convergence et choisissons de suivre l'évolution du critère J à minimiser (nous aurions pu continuer de nous intéresser à la convergence des coefficients vers l'optimum):

$$\underline{R} \underline{w} = \underline{R}_{ud}$$

D'après l'équation (2.7) et avec, on peut écrire :

$$\begin{aligned} J &= \frac{1}{2} \underline{w}^H \underline{R} \underline{w} - \underline{w}^H \underline{R} \underline{w} + \underline{w}^H \underline{R} \underline{w} \\ &= \frac{1}{2} \underline{w}^H \underline{R} \underline{w} + (\underline{w} - \underline{w})^H \underline{R} (\underline{w} - \underline{w}) \end{aligned}$$

D'après l'équation (1.12), on a donc :

$$J = J_{\min} + (\underline{w} - \underline{w})^H \underline{R} (\underline{w} - \underline{w}) \tag{2.24}$$

En utilisant la décomposition propre de la matrice d'autocorrélation et en utilisant de nouveau :

$$\underline{r}(n) = \underline{Q} (\underline{w}(n) - \underline{w}(n))$$

On obtient :

$$J = J_{\min} + \underline{r}^H \underline{P} \underline{r}$$

En utilisant maintenant :

$$\underline{r}(n) = (I - \mu \underline{P})^n \underline{r}(0)$$

Nous pouvons écrire :

$$J = J_{\min} + r^H(0) \left(\prod_{k=0}^{M-1} (1 - \mu \lambda_k) \right)^n \left(\prod_{k=0}^{M-1} (1 - \mu \lambda_k) \right)^n r(0)$$

Toutes les matrices étant diagonales.

$$J(n) = J_{\min} + \sum_{k=0}^{M-1} \left\{ \lambda_k \left(1 - \mu \lambda_k \right)^{2n} \left| r_k(0) \right|^2 \right\} \quad (2.25)$$

Lorsque l'algorithme est convergent c'est à dire lorsque : $0 < \mu < \frac{2}{\lambda_{\max}}$ quelque soient les conditions initiales $w(0)$.

La courbe obtenue en traçant $J(n)$ en fonction du nombre d'itérations n est appelée courbe d'apprentissage. D'après l'équation (2.25) cette courbe consiste en une somme d'exponentielles décroissantes, chacune d'elle correspondant à un mode propre de l'algorithme. La vitesse de convergence du mode k est liée à $(1 - \mu \lambda_k)^{2n}$.

Le mode le plus lent est lié à la valeur propre la plus petite et le mode le plus rapide est lié à la valeur propre la plus grande.

A une valeur propre, on associe une constante de temps, telle que :

$$\left| 1 - \mu \lambda_k \right|^2 = \exp\left(\frac{-1}{\tau_k}\right)$$

La constante de temps de l'erreur quadratique est alors

$$\frac{-1}{2 \log(1 - \mu \lambda_{\max})} \leq \tau \leq \frac{-1}{2 \log(1 - \mu \lambda_{\min})} \quad (2.26)$$

Ceci montre que la convergence est d'autant plus lente que le pas est faible.

Remarque : pour $\mu \ll 1$ on peut approcher la constante de temps par :

$$\tau_k = \frac{1}{2 \mu \lambda_k}$$

2.3.1.3. Pas optimal :

La rapidité de convergence est gouvernée, pour chacun des modes, par $u_k = |1 - \mu \lambda_k|$

La solution optimale consiste à minimiser le plus grand des u_k : il s'agit donc d'un problème de type minimax :

$$\mu_{opt} = \arg \min_{\mu} \max_k (|1 - \mu \lambda_k|) \quad (2.27)$$

Pour : $\mu > \mu_{opt}$ $1 - \mu \lambda_k < 1 - \mu_{opt} \lambda_k$ $\max (|1 - \mu \lambda_k|) = \mu \lambda_{\max} - 1$

Pour : $\mu < \mu_{opt}$ $1 - \mu \lambda_k > 1 - \mu_{opt} \lambda_k$ $\max (|1 - \mu \lambda_k|) = 1 - \mu \lambda_{\min}$

Donc pour $\mu = \mu_{opt}$ on a $1 - \mu \lambda_k = 1 - \mu_{opt} \lambda_k$

On a par conséquent :

$$\mu_{opt} = \frac{2}{\lambda_{\min} + \lambda_{\max}} \quad (2.28)$$

On peut noter que ce pas appartient bien au domaine des pas qui assurent la convergence, c'est-à-dire : $\tilde{\alpha} \in \left[0, \frac{2}{\lambda_{\max}} \right]$

Cependant, calculer les valeurs propres de la matrice de corrélation est aussi compliqué que d'inverser la matrice.

Le choix du pas fait donc perdre l'avantage de prendre un algorithme « économique ».

Une mesure de compromis consiste à prendre :

$$\tilde{\alpha} = \frac{2}{\sum_{k=1}^M \lambda_k} = \frac{2}{\text{trace}(\underline{\underline{R}}_{uu})} \tag{2.29}$$

La trace de $\underline{\underline{R}}_{uu}$ est simple à calculer (somme des éléments sur la diagonale principale). La matrice étant de Toeplitz, on peut aussi voir que :

$$\text{trace} \underline{\underline{R}}_{uu} = M r_{uu}(0)$$

Et $r_{uu}(0)$ est la puissance de $u(n)$:

$$r_{uu}(0) = E \left\{ |u(n)|^2 \right\}$$

On constate facilement que $\tilde{\alpha} < \alpha_{\max}$ et, avec un peu de chance, n'est pas très loin de α_{opt} .

2.3.1.4. Pas optimal et conditionnement de la matrice d'autocorrélation :

D'après :

$$\underline{\underline{r}}(n) = (\underline{\underline{I}} - \tilde{\alpha} \underline{\underline{P}})^n \underline{\underline{r}}(0)$$

La convergence de $\underline{\underline{r}}(n)$ est gouvernée par la valeur propre la plus grande de :

$$\underline{\underline{A}} = \underline{\underline{I}} - \tilde{\alpha} \underline{\underline{P}}$$

On peut remarquer que pour n assez grand on a :

$$\underline{\underline{r}}(n) \approx \zeta (\underline{\underline{A}})^n \underline{\underline{r}}(0)$$

Où ζ est le rayon spectral (la plus grande valeur propre) de $\underline{\underline{A}}$. Pour $\tilde{\alpha} = \alpha_{opt}$ on a :

$$\zeta(\underline{\underline{A}}) = 1 - \tilde{\alpha} \lambda_{\min} = \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} = \frac{C(\underline{\underline{R}}) - 1}{C(\underline{\underline{R}}) + 1} \tag{2.30}$$

Où :

$$C(\underline{\underline{R}}) = \frac{\lambda_{\max}}{\lambda_{\min}}$$

Est le conditionnement de $\underline{\underline{R}}$.

Ce résultat montre que la vitesse de convergence est directement liée au conditionnement de $\underline{\underline{R}}$,

Si $C(\underline{\underline{R}})$ est proche de 1, $\zeta(\underline{\underline{A}})$ est proche de 0, et la vitesse de convergence est très élevée.

Au contraire, si $C(\underline{\underline{R}}) \rightarrow \infty$ alors $\zeta(\underline{\underline{A}}) \rightarrow 1$, et l'algorithme est très lent à converger.

En d'autres termes, l'algorithme converge d'autant plus lentement que la matrice de covariance est mal conditionnée.

2.3.2. Autre présentation de la méthode du gradient :

Cette autre présentation s'appuie sur le résultat suivant :

$$\forall \mu < \frac{1}{\lambda_{\max}(\underline{R})} \quad \underline{R}^{-1} = \sum_{k=0}^{\infty} (\underline{I} - \mu \underline{R})^k$$

On peut montrer cette égalité de la même façon que pour établir la somme d'une suite géométrique-convergente.

Dans ce cas :

$$\underline{w} = \underline{R}^{-1} \underline{R}_{du} = \sum_{k=0}^{\infty} (\underline{I} - \mu \underline{R})^k \underline{R}_{du}$$

En posant alors :

$$\underline{w}(n) = \sum_{k=0}^n (\underline{I} - \mu \underline{R})^k \underline{R}_{du} \tag{2.31}$$

On a :

$$\begin{aligned} \underline{w}(n+1) &= \sum_{k=0}^{n+1} (\underline{I} - \mu \underline{R})^k \underline{R}_{du} \\ &= (\underline{I} - \mu \underline{R}) \left[\sum_{k=0}^n (\underline{I} - \mu \underline{R})^k \underline{R}_{du} + \mu (\underline{I} - \mu \underline{R})^0 \underline{R}_{du} \right] \\ &= (\underline{I} - \mu \underline{R}) \underline{w}(n) + \mu \underline{R}_{du} \end{aligned}$$

Soit :

$$\underline{w}(n+1) = \underline{w}(n) + \mu (\underline{R}_{du} - \underline{R}_{du} \underline{w}(n)) \tag{2.32}$$

On retrouve donc l'algorithme du gradient, qui, vu comme ceci, consiste à inverser la matrice \underline{R}_{uu} à l'aide du développement en série de \underline{R}_{uu}^{-1} .

2.4. Algorithme du gradient stochastique :

L'algorithme du gradient stochastique, ou plus exactement la famille des algorithmes de gradient stochastique, consiste à remplacer le gradient :

$$\nabla J|_{\underline{w}=\underline{w}(n)} = 2 \underline{R}_{uu} \underline{w}(n) - 2 \underline{R}_{du}$$

Quantité déterministe, exacte, par une approximation du gradient calculée à partir des données disponibles. Les données étant considérées comme aléatoires, le gradient estimé devient lui aussi une quantité aléatoire et l'algorithme devient stochastique. Parmi cette famille d'algorithmes, le plus connu est l'algorithme LMS (Least Mean Square).

Le problème majeur dans le gradient déterministe est que \underline{R}_{uu} et \underline{R}_{du} sont évidemment inconnus, on approchera donc ces grandeurs déterministes par des estimées $\underline{R}_{uu}(n)$ et $\underline{R}_{du}(n)$ à l'instant n.

Dans le cas du **LMS**, on choisit les estimées les plus simples possibles, à savoir :

$$\underline{R}_{uu}(n) = \underline{u}(n)\underline{u}^H(n) \text{ Et } \underline{R}_{du}(n) = d(n)\underline{u}(n)$$

Ce sont simplement les estimées instantanées des corrélations.

La relation permettant la mise à jour du filtre \underline{w} est ainsi maintenant :

$$\underline{w}(n+1) = \underline{w}(n) + \mu \underline{u}(n) (d^*(n) - \underline{u}^H(n)\underline{w}(n)) \quad (2.33)$$

Il est immédiat de montrer que cette approche est celle qui consiste à prendre l'algorithme du gradient sur le critère instantané : minimiser $|e(n)|^2$ (au lieu du critère statistique : minimiser $E\{|e(n)|^2\}$).

L'algorithme peut en fait se décomposer en :

- Calcul de la sortie du filtre : $y(n) = \underline{w}^H(n)\underline{u}(n)$
- Calcul de l'erreur d'estimation : $e(n) = d(n) - y(n)$
- Mise à jour : $\underline{w}(n+1) = \underline{w}(n) + \mu e^*(n)\underline{u}(n)$

Au temps n , on calcule donc la sortie du filtre et l'erreur d'estimation, en utilisant le filtre courant $\underline{w}(n)$.

Le filtre est ensuite mis à jour en ajoutant au filtre courant, le terme correctif :

$$\mu e^*(n)\underline{u}(n)$$

On remarquera que $\underline{w}(n)$ est maintenant un processus aléatoire, ceci signifie que pour des jeux de données différents, les trajectoires obtenues seront différentes.

On peut s'intéresser à la convergence de la trajectoire moyenne $E\{\underline{w}(n)\}$ et étudier:

$$E\{\underline{w}(n+1)\} = E\{\underline{w}(n)\} + \mu (\underline{R}_{du} - \underline{R}_{uu} E\{\underline{v}(n)\})$$

Si on suppose que $\underline{w}(n)$ est indépendant de $\underline{u}(n)$ (ce qui est très faux) on a :

$$E\{\underline{u}(n)\underline{u}^H(n)\underline{w}(n)\} = \underline{R}_{uu} E\{\underline{w}(n)\}$$

et la trajectoire moyenne suit l'algorithme du gradient déterministe.

Dans les autres cas (de façon réaliste), on n'a pas :

$$E\{\underline{u}(n)\underline{u}^H(n)\underline{w}(n)\} = \underline{R}_{uu} E\{\underline{w}(n)\}$$

Car $\underline{w}(n)$ a toutes les chances d'être corrélé à $\underline{u}(n)$.

La trajectoire moyenne n'est donc pas nécessairement celle du gradient déterministe. On peut cependant espérer qu'elle en soit proche, et que le caractère aléatoire se traduise par des fluctuations autour de cette valeur moyenne.

Comme les estimées instantanées sont très variables, on pourrait penser que l'algorithme LMS a des performances très pauvres. Cependant, comme l'algorithme est par nature récursif, il moyenne ces estimées au cours des itérations, ce qui permet d'obtenir des performances acceptables.

2.4.1. Convergence et stabilité :

On étudie la convergence en s'appuyant sur l'hypothèse d'indépendance entre $\underline{u}(n)$ et $\underline{w}(n)$. Rappelons que cette hypothèse est atrocement fautive. Les résultats corrects (très techniques) sont établis dans le livre d'O. Macchi [5].

Dans ce cas (indépendance entre $\underline{u}(n)$ et $\underline{w}(n)$) si on s'intéresse à la convergence de la trajectoire moyenne $E\{\underline{w}(n)\}$ on a simplement :

$$E\{\underline{w}(n+1)\} = E\{\underline{w}(n)\} + \mu E\{\underline{u}(n)(d^*(n) - \underline{u}^H(n)\underline{w}(n))\}$$

Ceci étant simplement l'algorithme du gradient déterministe, on aura convergence si :

$$\mu < \frac{2}{\lambda_{\max}} \quad (\text{Pour un pas fixe}).$$

Une étude plus précise (en moyenne quadratique), mais qui repose également sur des hypothèses contestables conduit à la condition plus contraignante :

$$\mu < \frac{2}{\text{Trace}(R_{uu})}$$

Où l'on pourra estimer la trace comme $\frac{1}{M} \text{Trace}(M)$ si M est la dimension de la matrice. En pratique, on prendra Γ de l'ordre de 2 ou 3.

2.4.1.1. Comportement de l'erreur :

L'erreur définie par $e(n) = d(n) - \underline{u}^T(n)\underline{w}$ peut être exprimée en fonction de l'erreur optimale obtenue à partir du filtre optimal :

$$\hat{e}(n) = d(n) - \underline{u}^T(n)\underline{y}$$

D'où :

$$e(n) = \hat{e}(n) - \underline{u}^T(n)(\underline{w} - \underline{y}) = \hat{e}(n) - \underline{u}^T(n)\underline{v}(n)$$

Où $\underline{v}(n)$ est l'écart ou filtre optimal, Cet écart vérifie l'équation récursive :

$$\underline{v}(n+1) = (\underline{I} - \mu \underline{u}(n)\underline{u}^T(n))\underline{v}(n) + \mu \hat{e}(n)\underline{u}(n)$$

La solution s'écrit sous la forme :

$$\underline{v}(n+1) = \underline{Q}_n \underline{v}(0) + \underline{F}_{n+1} \tag{2.34}$$

Avec :

$$\underline{Q}_n = \prod_{j=0}^{j=n} \left((\underline{I} - \mu \underline{u}(j)\underline{u}^T(j)) \right)$$

Et \underline{F}_n qui suit :

$$\underline{F}(n+1) = (\underline{I} - \mu \underline{u}(n)\underline{u}^T(n))\underline{F}(n) + \mu \hat{e}(n)\underline{u}(n)$$

Cette relation fait apparaître deux termes :

- $Q \underline{v}(0)$ représente la contribution des conditions initiales et correspond à une réponse transitoire.
- E_{n+1} résulte de l'excitation permanente et représente une fluctuation autour de la solution optimale.

On note alors :

$$\underline{v}(n+1) = \underline{v}(n+1)^{tr} + \underline{v}(n+1)^{fl} \tag{2.35}$$

L'erreur d'estimation devient alors :

$$e(n) = \hat{e}(n) - \underline{u}^T(n) \underline{v}(n)^{tr} - \underline{u}^T(n) \underline{v}(n)^{fl} \tag{2.36}$$

On montre que le transitoire tend vers 0 lorsque n tend vers l'infini presque sûrement.

L'erreur moyenne quadratique, lorsque n tend vers l'infini est alors :

$$\begin{aligned} J &= E \left\{ \hat{e}(n)^2 \right\} + E \left\{ \left| \underline{u}^T(n) \underline{v}(n)^{fl} \right|^2 \right\} \\ &= J_{\min} + E \left\{ e^{fl}(n)^2 \right\} \end{aligned} \tag{2.37}$$

L'erreur quadratique liée à la fluctuation est bornée :

$$E \left\{ e^{fl}(n)^2 \right\} \leq \sim J_{\min} \frac{Tr(\underline{R}_{uu})}{2 - \sim \underline{\alpha} Tr(\underline{R}_{uu})} \quad r \geq 1 \tag{2.38}$$

Notons bien que cette fluctuation provient du fait que la trajectoire est aléatoire, et que c'est la moyenne de $\underline{w}(n)$ qui converge vers \underline{w} .

En effet, si à l'étape n, n étant aussi grand que l'on veut, $\underline{w}(n) = \underline{w}$ alors, à l'étape (n+1) :

$$\underline{w}(n+1) = \underline{w} + \sim \left(\underline{u}(n) \underline{d}(n) - \underline{u}^T(n) \underline{w} \right) \tag{2.39}$$

Qui est différent de \underline{w} , sauf cas très particulier du fait que l'excitation est aléatoire.

De ces résultats, on déduit que :

- plus \sim est grand, mais vérifiant la condition de stabilité, plus la convergence est rapide, mais plus la variance résiduelle est importante, ce qui se traduit par des fluctuations importantes autour de la trajectoire moyenne.
- Pour de faibles valeurs de \sim , la convergence est lente, mais l'erreur résiduelle a une faible variance, c'est-à-dire que l'on a une trajectoire « presque » déterministe.
- La convergence de l'algorithme requiert donc un compromis entre **vitesse et fidélité**.

2.4.1.2. Convergence de l'algorithme à pas variable :

Afin d'accélérer la convergence, il est intéressant de faire varier le pas d'adaptation \sim :

- \sim doit être grand au départ, lorsque l'on est loin de l'optimum
- \sim doit être faible lorsque l'on se retrouve au voisinage de l'optimum.

On peut alors utiliser une séquence $\{ \sim_n \}$ de pas variable.

Dans ce cas, l'algorithme converge en moyenne quadratique, c'est à dire que :

$$E \left\{ \left| \underline{w}(n) - \underline{w} \right|^2 \right\} \xrightarrow{n \rightarrow \infty} 0$$

Si les conditions suffisantes de Robbin et Monroe sont satisfaite :

- $\sim_n > 0$ quelque soit n
- $\sum \sim_n$ diverge
- $\sum \sim_n^2$ converge

2.4.2. Poursuite – Adaptativité :

L'algorithme présenté jusqu'ici a été présenté sous l'angle d'une implantation récursive d'un filtre de Wiener, pour lequel on a implicitement supposé les signaux stationnaires. On s'est alors intéressé à la convergence vers la solution \underline{w} .

A partir de l'implantation récursive, il est également possible de voir l'algorithme comme un algorithme adaptatif, c'est-à-dire capable de suivre des modifications lentes, en permanence, du filtre optimal.

Celui-ci est alors dépendant du temps :

$$\underline{w} \longrightarrow \underline{w}(n)$$

On peut tout de suite noter que si l'on désire que le filtre soit adaptatif, on ne peut pas le laisser converger en prenant une séquence $\{\sim_n\}$, avec :

$$\sim_n \longrightarrow 0 \text{ Lorsque } n \longrightarrow \infty$$

L'algorithme doit conserver une « capacité de réaction » permettant la poursuite des non stationnarités.

On considère donc que $\underline{w}(n)$ évolue maintenant suivant une marche aléatoire :

$$\underline{w}(n) = \underline{w}(n-1) + \underline{v}(n) \tag{2.40}$$

Où $\underline{v}(n)$ est un bruit vectoriel, de matrice de covariance $\underline{\Gamma}$. On définit alors un nouvel écart au filtre optimal :

$$\underline{v}(n) = \underline{w}(n) - \underline{w}(n)$$

On a toujours :

$$\begin{aligned} e(n) &= d(n) - \underline{u}^T(n) \underline{w}(n) - \underline{u}^T(n) (\underline{w}(n) - \underline{w}(n)) \\ &= \hat{e}(n) - \underline{u}^T(n) \underline{v}(n) \end{aligned}$$

En résolvant récursivement, on a :

$$\underline{v}(n+1) = \underline{Q}_{\underline{v}} \underline{v}(0) + \underline{F}_{n+1} + \underline{E}_{n+1}$$

Où \underline{E}_{n+1} est la suite aléatoire « supplémentaire », correspondant à l'ajout de $\underline{v}(n)$ dans le modèle et qui obéit à l'équation récursive :

$$\underline{E}(n+1) = \left(\underline{I} - \underline{u}(n) \underline{u}^T(n) \right) \underline{E}(n) - \underline{v}(n+1)$$

Il s'en suit que :

$$e(n) = \hat{e}(n) + e^{tr}(n) + e^{fl}(n) + e^r(n)$$

Et une fois disparu le transitoire :

$$J = J_{\min} + E\{e^{fl}(n)^2\} + E\{e^r(n)^2\} \tag{2.41}$$

Avec :

$$E\{e^r(n)^2\} = \frac{Tr(\underline{\Gamma})}{2 - \mu \cdot Tr(\underline{R}_{uu})} \quad r \geq 1$$

On rappelle que :

$$E\{e^{fl}(n)^2\} \leq J_{\min} \frac{Tr(\underline{\Gamma})}{2 - \mu \cdot Tr(\underline{R}_{uu})} \quad r \geq 1$$

La minimisation de l'erreur quadratique conduit alors à choisir un pas optimal qui effectue un compromis entre ces deux erreurs.

2.4.3. Algorithme du signe :

Parmi les variantes simplifiées de l'algorithme LMS, l'algorithme du signe est intéressant, On remplace simplement l'erreur d'estimation $e(n)$ par son signe, ce qui conduit à la récurrence :

$$\underline{w}(n+1) = \underline{w}(n) + \mu(n) \cdot \text{signe}(e(n)) \tag{2.42}$$

On évite ainsi d'effectuer la multiplication dans la relation d'actualisation des coefficients, ce qui conduit ainsi à une complexité de M multiplications par itération.

Il est à noter qu'en dépit de cette simplification extraordinaire, la convergence reste convenable dans beaucoup d'applications, et en particulier en égalisation adaptative.

2.4.4. Algorithme LMS normalisé :

Dans la récurrence de l'algorithme LMS standard (avec des coefficients de filtre réels)

$$\underline{w}(n+1) = \underline{w}(n) + \mu(n) \cdot (\underline{d}(n) - \underline{u}^H(n) \cdot \underline{w}(n)) \tag{2.43}$$

La dynamique des fluctuations dépend de la dynamique du signal d'entrée $\underline{u}(n)$. On dit qu'on a une amplification du bruit par le gradient.

Reprenons l'équation du gradient déterministe :

$$\underline{w}(n+1) = \underline{w}(n) - \frac{1}{2} \cdot \nabla J \Big|_{\underline{w}=\underline{w}(n)}$$

Avec :

$$\nabla J \Big|_{\underline{w}=\underline{w}(n)} = 2\underline{R}_{uu} \cdot \underline{w} - 2\underline{R}_{ud} = \underline{\nabla}_n$$

A l'instant $n+1$, on a :

$$J(n+1) = E\{e(n+1)^2\}$$

Avec :

$$\begin{aligned} e(n+1) &= d(n+1) - \underline{w}^T \underline{u}(n+1) \\ &= d(n+1) - \underline{w}^T \underline{u}(n+1) + \frac{1}{2} \tilde{\mu}_n \nabla_n^T u(n+1) \end{aligned}$$

Si on cherche le pas optimal $\tilde{\mu}_n$ qui minimise $J(n+1)$, on doit avoir :

$$\frac{dJ(n+1)}{d\tilde{\mu}_n} = 2E \left\{ e(n+1) \frac{de(n+1)}{d\tilde{\mu}_n} \right\} = 0$$

Soit :

$$E \left\{ \nabla_n^T [d(n+1)\underline{u}(n+1) - \underline{w}_n^T \underline{u}(n+1)] \underline{u}^T(n+1) \nabla_n + \frac{1}{2} \mu_n \underline{u}(n+1) \underline{u}^T(n+1) \nabla_n \right\} = 0$$

Dans l'algorithme du gradient déterministe ∇_n est certain, la relation précédente s'écrit donc :

$$\nabla_n^T \underline{R}_{du} - \underline{w}_n^T \underline{R}_{uu} \nabla_n + \frac{1}{2} \mu_n \nabla_n^T \underline{R}_{uu} \nabla_n = 0$$

Soit finalement :

$$\tilde{\mu}_n = \frac{2(\underline{w}_n^T \underline{R}_{uu} \nabla_n - \nabla_n^T \underline{R}_{du})}{\underline{w}_n^T \underline{R}_{uu} \nabla_n}$$

L'estimée instantanée de \underline{R}_{uu} est :

$$\underline{R}_{uu}(n) = \underline{u}(n) \underline{u}^T(n)$$

Celle de \underline{R}_{du} est :

$$\underline{R}_{du} = d(n) \underline{u}_n$$

Celle de ∇_n est :

$$\nabla_n = 2(d(n) \underline{u}_n - \underline{u}_n \underline{u}_n^T \underline{w}_n)$$

En remplaçant \underline{R}_{uu} , \underline{R}_{du} et ∇_n par leurs estimées respectives il vient :

$$\tilde{\mu}_n = \frac{1}{\underline{u}_n^T \underline{u}_n} \tag{2.44}$$

Ce qui revient à prendre un pas inversement proportionnel à la puissance « instantanée » du signal d'entrée.

2.5. Formulation du filtrage de Wiener dans le cadre des moindres carrés :

Nous avons résolu précédemment le problème du filtrage de Wiener en nous appuyant sur la fonction de coût déterministe :

$$J = E \{ e(n)^2 \}$$

L'inconvénient de cette approche est que les grandeurs statistiques sont difficiles à obtenir. D'autre part comme nous l'avons déjà signalé, la solution optimale obtenue n'est valable que pour un signal stationnaire. L'approche du gradient stochastique nous a permis, à partir de ce critère statistique de nous affranchir de ces inconvénients mais nous pouvons choisir de reformuler le problème dans le cadre des moindres carrés, c'est-à-dire rechercher le filtre qui minimise la somme des carrés des erreurs, jusqu'à l'instant considéré n .

Nous donnerons dans le paragraphe « Algorithme des moindres carrés récursifs » une approche récursive de ce problème (autorisant une évolution pour un fonctionnement adaptatif).

Dans un premier temps nous allons considérer le cas dans lequel on connaît le signal d'entrée u_n sur N points et trouver le filtre optimal pour ce bloc de données, pour simplifier les équations nous considérerons maintenant le cas d'un filtre à coefficients réels.

2.5.1. Critère des moindres carrés pour un bloc de données de N points (u_0, u_1, \dots, u_{N-1}) :

$$J_{MC} = \sum_i (e(i)^2) \quad (2.45)$$

Avec :

$$y(i) = \sum_{l=0}^{M-1} w_l u(i-l)$$

Pour préciser les bornes de la sommation sur i , il faut prendre en compte le fait que pour $i < M - 1$ et $i > N - 1$, $y(i)$ dépend de valeurs d'entrée inconnues.

Différentes hypothèses sont possibles concernant les valeurs inconnues de l'entrée :

- Soit considérer $u(i) = 0$ pour $i < 0$
- Soit considérer $u(i) = 0$ pour $i > N - 1$
- Soit ignorer ces valeurs de l'entrée

On note la relation précédente :

$$\underline{y} = \underline{A}^T \underline{w}$$

La matrice \underline{A} est appelée matrice de données (ANNEXE 4).

Suivant les hypothèses, on pourra extraire plusieurs matrices :

- Si on refuse d'affecter les valeurs inconnues, on extrait une matrice de Toeplitz de dimension $(M)(N - M + 1)$, qui ne contient que les valeurs connues de l'entrée. En d'autres termes, on ne pourra calculer l'erreur que pour $i = M - 1$ à $i = N - 1$. Cette forme est appelée « forme covariance ».
- Si on affecte la valeur 0 aux échantillons de l'entrée pour $i < 0$, on parle de préfenêtrage, et on extrait la forme préfenêtrée, de dimension $M \times N$. On peut calculer $e(i)$ pour $i = 0$ à $N - 1$

- Si on affecte 0 aux valeurs inconnues pour $i > N-1$, on a une forme post-fenêtrée, et on peut calculer $e(i)$ pour $i = M-1$ à $N+M-2$.
- Enfin, en utilisant à la fois le pré et post fenêtrage, on obtient la forme «autocorrélation » de dimension $M \times (N+M-2)$, et $e(i)$ est calculable pour $i = 0$ à $N+M-2$.

Les termes de covariance et autocorrélation sont consacrés, en particulier en traitement de la parole. Cependant, il n’y a là aucun (ou peu) de rapport avec les définitions des fonctions et matrice de covariance et d’autocorrélation.

Notons de plus que le fait de compléter la séquence par des zéros n’est pas la seule solution. En particulier, il est possible de périodiser la séquence, ce qui conduit à des matrices circulantes. Suivant les hypothèses faites sur le fenêtrage, on ne minimisera donc pas nécessairement l’erreur sur tout l’horizon :

$$J = \sum_{i=M-1}^{N-1} e(i)^2$$

Pour la forme covariance, par exemple.

On note \underline{d} le vecteur de la sortie désirée, \underline{y} le vecteur de sortie, et \underline{A} la matrice de donnée.

Le vecteur d’erreur \underline{e} vaut :

$$\underline{e} = \underline{y} - \underline{d}$$

Le critère J s’écrit :

$$J = \underline{e}^T \underline{e} = (\underline{y} - \underline{d})^T (\underline{y} - \underline{d})$$

Il ne reste plus qu’à exprimer le gradient de J par rapport à \underline{w} :

$$\nabla J|_{\underline{w}} = 2\underline{A}(\underline{A}^T \underline{w} - \underline{d})$$

Et on en déduit que :

$$\underline{A} \underline{A}^T \underline{w} = \underline{A} \underline{d}$$

En développant $\underline{A} \underline{A}^T$ et $\underline{A} \underline{d}$, il est facile de voir que les termes génériques valent :

$$(\underline{A} \underline{A}^T)_{ij} = \sum_l u(l-i)u(l-i) = \hat{r}_{uu}(i-j)$$

$$\underline{A} \underline{d} = \sum_l u(l-i)d(l) = \hat{r}_{du}(i)$$

On retrouve ainsi les équations normales (équation de Wiener Hopf) en utilisant des corrélations empiriques (ou temporelles).

Notons que suivant les hypothèses effectuées sur le fenêtrage, ces estimées ne seront pas identiques.

En particulier, la matrice de corrélation empirique est de Toeplitz avec la méthode d’autocorrélation, alors qu’elle ne l’est pas forcément dans les autres hypothèses.

On peut encore noter que :

$$\underline{R}_{uu} = \underline{A} \underline{A}^T$$

Est par construction définie non négative (c’est un « carré » matriciel) et symétrique (à symétrie hermitienne).

2.5.1.1. Théorème de projection :

A l'optimum, le gradient s'annule :

$$\underline{A}(\underline{A}^T \underline{w} - \underline{d}) = \underline{0} \Leftrightarrow \underline{A} \underline{e} = \underline{0}$$

Par conséquent, en multipliant par \underline{w}^T on obtient :

$$\underline{y}^T \underline{e} = 0$$

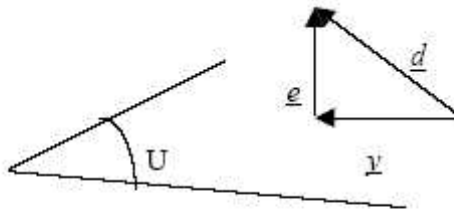


Figure.2.7- Théorème de projection.

2.5.2. Solution de l'équation normale :

A partir de :

$$\underline{A} \underline{A}^T \underline{w} = \underline{A} \underline{d}$$

On obtient :

$$\underline{w}_{MC} = (\underline{A} \underline{A}^T)^{-1} \underline{A} \underline{d}$$

Si $\underline{A} \underline{A}^T$ est inversible, donc le filtre optimal pour le critère des moindres carrés J_{MC} pour le bloc de données de N points $(u_0, u_1, \dots, u_{N-1})$.

Les inconvénients de cette approche sont les suivants :

- La solution optimale s'obtient en inversant la matrice ce qui représente une grosse charge de calcul.
- Le fonctionnement en bloc ne permet pas à priori de prendre en compte une nouvelle donnée u_N pour ajuster le filtre sans tout recalculer.
- L'algorithme peut être vu comme adaptatif si on considère des blocs successifs de N points mais ne peut pas suivre des non stationnarités à évolution rapide.

Ces différents inconvénients justifient la recherche d'une approche récursive pour ce critère des moindres carrés.

2.6. Algorithme des moindres carrés récursifs :

L'approche des moindres carrés consiste à minimiser la somme des carrés des erreurs :

$$J_{MC} = \sum_{i=1}^n (e(i)^2).$$

On obtient alors la solution des moindres carrés :

$$\underline{w}_{MC} = \arg \min_{\underline{w}} \left(\underline{A}^T \underline{w} - \underline{d} \right)^T \left(\underline{A}^T \underline{w} - \underline{d} \right) \quad \text{Où } \underline{A} \text{ est la matrice de données.}$$

La solution exacte s'écrit :

$$\underline{w}_{MC} = \left(\underline{A} \underline{A}^T \right)^{-1} \underline{A} \underline{d}$$

Nous avons vu que la matrice $\left(\underline{A} \underline{A}^T \right)$ (correspond à une estimation de la matrice de corrélation \underline{R}_{uu} , et le produit $\underline{A} \underline{d}$ est l'estimée de l'intercorrélation \underline{R}_{ud} .

L'approche des moindres carrés est par essence une approche bloc, c'est-à-dire qu'on traite un bloc de données pour construire \underline{A} , puis en déduire \underline{w}_{MC} .

L'inversion de $\left(\underline{A} \underline{A}^T \right)$ est coûteuse (de l'ordre de M^3).

Par ailleurs, la nature « bloc » de l'algorithme empêche à priori de prendre en compte simplement une nouvelle donnée, en profitant du résultat déjà obtenu :

Si on a résolu $\underline{w}_{MC} = \arg \min_{\underline{w}} \sum_{i=1}^n (e(i)^2)$ Peut-on relier $\underline{w}_{MC}^{(n+1)} = \arg \min_{\underline{w}} \sum_{i=1}^{n+1} (e(i)^2)$ à $\underline{w}_{MC}^{(n)}$?

La réponse est bien entendu affirmative. L'algorithme correspondant est l'algorithme des moindres carrés récursifs, qui permet de mettre à jour la solution au cours du temps. Notons dès à présent qu'il ne s'agit pas, tel quel, d'un algorithme adaptatif : dans la solution est contenue toute l'histoire du signal jusqu'à l'instant n, avec une hypothèse implicite de stationnarité. Pour obtenir un algorithme adaptatif on pourra introduire un « facteur d'oubli » sur les données les plus anciennes (voir paragraphe correspondant).

2.6.1. Construction de l'algorithme des MCR :

On pose :

$$\underline{w}_{MC}^{(n)} = \left(\underline{R}_{uu}^{(n)} \right)^{-1} \underline{R}_{ud}^{(n)}$$

Où $\underline{R}_{uu}^{(n)}$ et $\underline{R}_{ud}^{(n)}$ sont la matrice $\left(\underline{A} \underline{A}^T \right)$ et le vecteur $\underline{A} \underline{d}$ construits à partir des données jusqu'à l'instant n .

Dorénavant nous considérerons de plus que $i_1 = 0$ (hypothèse préférée)

La matrice $\underline{R}_{uu}^{(n)}$ possède alors la structure suivante :

$$\underline{R}_{uu}^{(n)} = \sum_{i=0}^n \underline{u}(i) \underline{u}^T(i)$$

Dans ces conditions, à l'instant $(n+1)$:

$$\underline{\underline{R}}^{(n+1)} = \underline{\underline{R}}^{(n)} + \underline{u}(n+1)\underline{u}^T(n+1)$$

Et on dispose d'une équation récursive pour estimer la matrice de corrélation.

Le problème est maintenant que pour obtenir une formulation récursive pour le vecteur optimal, il faut avoir une formulation récursive sur l'inverse de la matrice de corrélation. Pour cela on utilise, le **lemme d'inversion matriciel**.

2.6.1.1. Lemmes d'inversion matriciels :

Si \underline{A} \underline{B} \underline{C} \underline{D} sont des matrices de dimensions convenables :

$$\left(\underline{A} + \underline{B}\underline{C}\underline{D}\right)^{-1} = \underline{A}^{-1} - \underline{A}^{-1}\underline{B}\left(\underline{C}^{-1} + \underline{D}\underline{A}^{-1}\underline{B}\right)^{-1}\underline{D}\underline{A}^{-1} \tag{2.46}$$

Ce lemme se démontre par vérification directe (ANNEXE 5).

Dans notre cas, il s'agit d'inverser :

$$\underline{\underline{R}}^{(n+1)} = \underline{\underline{R}}^{(n)} + \underline{u}(n+1)\underline{u}^T(n+1)$$

On prendra donc :

$$\underline{A} = \underline{\underline{R}}^{(n)}, \underline{B} = \underline{u}(n+1), \underline{C} = 1, \underline{D} = \underline{u}^T(n+1)$$

En appliquant le lemme d'inversion matricielle, il vient alors :

$$\left(\underline{\underline{R}}^{(n+1)}\right)^{-1} = \left(\underline{\underline{R}}^{(n)}\right)^{-1} - \frac{\left(\underline{\underline{R}}^{(n)}\right)^{-1}\underline{u}(n+1)\underline{u}^T(n+1)\left(\underline{\underline{R}}^{(n)}\right)^{-1}}{1 + \underline{u}^T(n+1)\left(\underline{\underline{R}}^{(n)}\right)^{-1}\underline{u}(n+1)}$$

Pour alléger l'écriture, on pose :

$$\underline{\underline{K}}_n = \left(\underline{\underline{R}}^{(n)}\right)^{-1}$$

On a alors :

$$\underline{\underline{K}}_{n+1} = \underline{\underline{K}}_n - \frac{\underline{\underline{K}}_n \underline{u}(n+1)\underline{u}^T(n+1)\underline{\underline{K}}_n}{1 + \underline{u}^T(n+1)\underline{\underline{K}}_n \underline{u}(n+1)} \tag{2.47}$$

On notera que cette formule permet de calculer l'inverse de la matrice de corrélation, sans calcul explicite d'inverse.

Le vecteur d'intercorrélation $\underline{R}_{ud}^{(n)}$ s'écrit quant à lui, sous forme récursive, selon :

$$\underline{R}_{ud}^{(n+1)} = \underline{R}_{ud}^{(n)} + d(n+1)\underline{u}(n+1)$$

Le filtre optimal à l'itération $(n+1)$ s'écrit donc :

$$\underline{w}_{MC}^{(n+1)} = \left(\underline{\underline{R}}^{(n+1)}\right)^{-1} \underline{R}_{ud}^{(n+1)} = \underline{\underline{K}}_{n+1} \left(\underline{R}_{ud}^{(n)} + d(n+1)\underline{u}(n+1)\right)$$

Pour simplifier plus loin nous explicitons le dernier terme de cette expression : $\underline{\underline{K}}_{n+1} \underline{u}(n+1)$

$$\begin{aligned}
\underline{K}_{n+1} \underline{u}(n+1) &= \left(\underline{K}_{n+1} - \frac{\underline{K}_{n+1} \underline{u}(n+1) \underline{u}^T(n+1) \underline{K}_{n+1}}{1 + \underline{u}^T(n+1) \underline{K}_{n+1} \underline{u}(n+1)} \right) \underline{u}(n+1) \\
&= \underline{K}_{n+1} \underline{u}(n+1) \left(1 - \frac{\underline{u}^T(n+1) \underline{K}_{n+1} \underline{u}(n+1)}{1 + \underline{u}^T(n+1) \underline{K}_{n+1} \underline{u}(n+1)} \right) \\
&= \frac{\underline{K}_{n+1} \underline{u}(n+1)}{1 + \underline{u}^T(n+1) \underline{K}_{n+1} \underline{u}(n+1)}
\end{aligned}$$

Nous avons donc :

$$\begin{aligned}
\underline{w}^{(n+1)} &= \underline{K}_{n+1} \left(\underline{R}_{ind}^{(n)} + d(n+1) \right) \underline{u}(n+1) \\
&= \left(\underline{K}_{n+1} - \frac{\underline{K}_{n+1} \underline{u}(n+1) \underline{u}^T(n+1) \underline{K}_{n+1}}{1 + \underline{u}^T(n+1) \underline{K}_{n+1} \underline{u}(n+1)} \right) \left(\underline{R}_{ind}^{(n)} + d(n+1) \right) \underline{u}(n+1) \\
&= \underline{w}^{(n)} - \frac{\underline{K}_{n+1} \underline{u}(n+1) \underline{u}^T(n+1) \underline{K}_{n+1} \left(\underline{R}_{ind}^{(n)} + d(n+1) \right)}{1 + \underline{u}^T(n+1) \underline{K}_{n+1} \underline{u}(n+1)} \\
&= \underline{w}^{(n)} - \frac{\underline{K}_{n+1} \underline{u}(n+1)}{1 + \underline{u}^T(n+1) \underline{K}_{n+1} \underline{u}(n+1)} \left(\underline{u}^T(n+1) \underline{K}_{n+1} \underline{R}_{ind}^{(n)} - d(n+1) \right)
\end{aligned}$$

Soit finalement :

$$\underline{w}^{(n+1)} = \underline{w}^{(n)} - \underline{K}_{n+1} \underline{u}(n+1) \left(d(n+1) - \underline{u}^T(n+1) \underline{w}^{(n)} \right) \quad (2.48)$$

Cette formule, définissant le nouveau filtre optimal à partir du filtre est aussi appelée, par analogie, formule de Kalman.

Les équations (2.47) et (2.48) constituent l'**algorithme des moindres carrés récursifs**.

Dans l'algorithme développé ci-dessus, on n'a pas utilisé toutes les caractéristiques de la matrice de corrélation, et en particulier le fait que la matrice de corrélation (dans le cas de l'hypothèse autocorrélation) est de Toeplitz.

On peut alors obtenir une formule de renouvellement plus simple.

Par ailleurs, on peut effectuer à la fois une récurrence sur le temps, mais aussi sur l'ordre (Ordre Réursive Adaptive Least Squares).

On peut également trouver une récurrence sur le gain $\underline{K}_{n+1} \underline{u}(n+1)$ plutôt que sur l'inverse de la matrice de corrélation. Les algorithmes obtenus sont les algorithmes des moindres carrés récursifs rapides, qui peuvent poser des problèmes d'instabilité numérique.

On notera que le terme d'erreur s'écrit :

$$e(n+1) = d(n+1) - \underline{u}^T(n+1) \underline{w}^{(n)}$$

Alors qu'on avait dans le cas des algorithmes du gradient :

$$e(n+1) = d(n+1) - \underline{u}^T(n+1) \underline{w}(n+1)$$

La première erreur est l'erreur à priori, c'est-à-dire l'erreur effectuée avant d'avoir mis à jour le filtre avec la nouvelle donnée $u(n+1)$.

Cette erreur est différente de l'erreur à posteriori, dans laquelle le filtre a été mis à jour.

2.6.2. Problèmes d'initialisation :

La mise en oeuvre pratique des algorithmes des MCR requiert le choix des valeurs initiales \underline{K}_0 et $\underline{w}^{(0)}$.

La matrice de corrélation, à l'étape n s'exprimant comme une somme de dyades :

$$\underline{R}_{uu}^{(n)} = \sum_{i=0}^n u(i) u^T(i)$$

Cette matrice est singulière tant que $n < M$, ce qui empêche de démarrer la récurrence.

Deux solutions sont alors possibles :

- évaluer la matrice de corrélation et son inverse pour un $n > M$
- modifier légèrement la formule d'estimation de en introduisant un « talon » :

$$\underline{R}_{uu}^{(n)} = \sum_{i=0}^n u(i) u^T(i) + \sigma \underline{I}$$

Où σ est un scalaire positif très faible.

Il s'en suit que :

$$\underline{R}_{uu}^{(n)} = \sigma \underline{I} \quad \text{et} \quad \underline{K}_n = \frac{1}{\sigma} \underline{I}$$

Ce choix consiste en fait à modifier le critère initial en :

$$J' = \sigma \|\underline{w}^{(n)}\|^2 + \sum_{i=0}^n e(i)^2$$

L'intervention de la constante σ introduit une régularisation de l'algorithme, en pénalisant la norme du vecteur optimal (ce qui est équivalent à effectuer la minimisation de $J_{MC}(n)$, sous contrainte).

Il reste à choisir $\underline{w}^{(0)}$, que l'on choisit usuellement comme le vecteur nul.

2.6.3. Version adaptative – pondération exponentielle :

L’algorithme précédent n’est pas adaptatif. Pour le rendre adaptatif, il faut « oublier » le passé du signal lors de l’évolution de l’algorithme. Pour cela, une approche classique consiste à donner un poids plus important aux valeurs les plus récentes : on modifie le critère en introduisant une pondération exponentielle :

$$J' = \sum_{i=0}^n \lambda^{n-i} e(i)^2$$

On montre alors aisément que seule l’équation de renouvellement de l’inverse de la corrélation est modifiée :

$$K_{n+1} = \frac{1}{\lambda} K_n - \frac{1}{\lambda} \frac{K_n \underline{u}(n+1) \underline{u}^T(n+1) K_n}{1 + \frac{1}{\lambda} \underline{u}^T(n+1) K_n \underline{u}(n+1)} \tag{2.49}$$

Cet algorithme ne converge plus (la matrice de corrélation reste aléatoire lorsque $n \rightarrow \infty$, mais ceci est normal puisque l’on se place dans une perspective adaptative, en permettant à l’algorithme de réagir à des non stationnarités.

2.6.4. Equation récursive sur le critère somme quadratique :

A l’instant n , pour le vecteur optimal, on a :

$$\begin{aligned} J(n) &= \sum_{i=0}^n \lambda^{n-i} e(i)^2 \\ &= \sum_{i=0}^n \lambda^{n-i} \left(d(i) - \underline{w}^{T(n)} \underline{u}(i) \right)^2 \\ &= \sum_{i=0}^n \lambda^{n-i} \left(d(i)^2 - \underline{R}_{ud}^{T(n)} \underline{w}^{(n)} \right) \end{aligned}$$

Où $\underline{R}_{ud}^{T(n)}$ est l’intercorrélation « exponentielle » :

$$\underline{R}_{ud}^{T(n)} = \sum_{i=0}^n \lambda^{n-i} \underline{u}(i) d(i).$$

En posant :

$$\dagger_d^2(n) = \sum_{i=0}^n \lambda^{n-i} (d(i))^2$$

On a :

$$\dagger_d^2(n) = \lambda \dagger_d^2(n-1) + (d(n))^2$$

Et :

$$\underline{R}_{ud}^T(n) = \lambda \underline{R}_{ud}^T(n-1) + d(n) \underline{u}(n)$$

En utilisant ces deux relations, il vient :

$$J(n) = \lambda J(n-1) + \zeta_n e_n \tag{2.50}$$

Notons que :

$$\begin{aligned} e(n) &= d(n) - \underline{w}^T(n) \underline{u}(n) \\ &= d(n) - \left(\underline{w}^T(n-1) + \underline{K}_n \underline{u}(n) \right) \underline{u}(n) \end{aligned}$$

Soit :

$$e(n) = \left(1 - \underline{u}^T(n) \underline{K}_n \underline{u}(n) \right) \xi_n$$

Le rapport entre les erreurs à posteriori et à priori s'appelle facteur de conversion :

$$x(n) = \frac{e(n)}{\xi(n)} = 1 - \underline{u}^T(n) \underline{K}_n \underline{u}(n)$$

2.6.5. Convergence de l'algorithme, en moyenne :

On supposera ici que :

$$d(n) = \underline{w}^T \underline{u}(n) + \hat{e}(n)$$

Où \underline{w} est le vecteur filtre exact et un bruit d'observation blanc centré, décorrélé de $\underline{u}(n)$ (notez que ce modèle est restrictif : rien ne dit, dans le filtrage de Wiener que $d(n)$ est issu d'une filtrée linéaire de $u(n)$).

En écrivant l'algorithme, on a :

$$\underline{w}^n = \underline{w}^{n+1} + \underline{K}_n \underline{u}(n) \left(d(n) - \underline{u}^T(n) \underline{w}^{n-1} \right)$$

Exprimons $\underline{R}_{ud}^{(n)}$ en remplaçant $d(i)$ par son expression :

$$\begin{aligned} \underline{R}_{ud}^{(n)} &= \sum_{i=0}^n \underline{w}^T \underline{u}(i) \underline{u}(i) + \hat{e}(i) \underline{u}(i) \\ &= \underline{R}_{uu}^{(n)} \underline{w} + \sum_{i=0}^n \hat{e}(i) \underline{u}(i) \end{aligned}$$

En multipliant par \underline{K}_n il vient :

$$\underline{w}^{(n)} = \underline{w} + \underline{K}_n \sum_{i=0}^n \hat{e}(i) \underline{u}(i)$$

En utilisant maintenant la composition des espérances :

$$E_{X,Y}(X(Y)) = E_Y(E_{X/Y}(X/Y))$$

Et grâce à la décorrélation (l'indépendance) entre $e(n)$ et $u(n)$, et au fait que soit centré, on obtient :

$$E(\underline{w}(n)) = \underline{w} \tag{2.51}$$

L'algorithme converge donc en moyenne, ce qui est rassurant, et il converge à nombre d'itérations fini : il n'est pas nécessaire d'avoir $n \longrightarrow \infty$ pour obtenir la convergence (cas du LMS).

2.6.6. Erreur quadratique moyenne :

Avec des hypothèses irréalistes, mais qui permettent d'aboutir à un résultat bien vérifié en pratique, on montre que :

$$E\left\{\left|w(n) - \underline{w}\right|^2\right\} \propto \frac{1}{n} \sum_{i=0}^{M-1} \frac{1}{\lambda_i}$$

Ce résultat montre que l'algorithme converge en $1/n$, et que l'erreur quadratique est inversement proportionnelle aux valeurs propres. Une fois de plus, le mauvais conditionnement limite la vitesse de convergence.

2.6.7. Algorithmes sous optimaux :

L'idée des algorithmes sous optimaux est de simplifier l'équation de renouvellement de \underline{K}_n . En particulier, on obtient des algorithmes du type LMS en effectuant la simplification ultime :

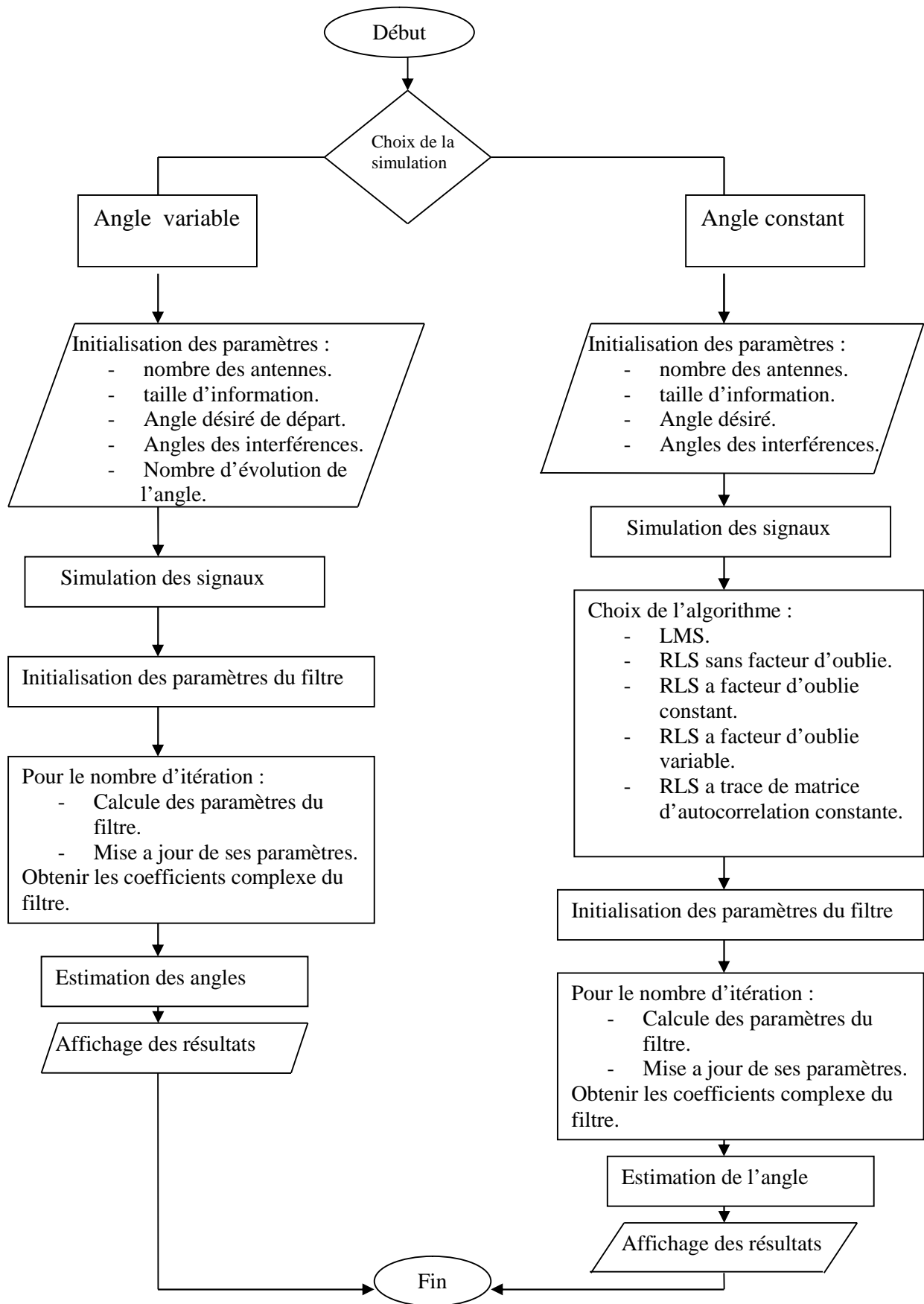
$$\underline{K}_{n+1} = \sim_{n+1} \underline{I}$$

2.7. Conclusion :

Les algorithmes adaptatifs utilisés pour minimiser l'erreur quadratique sont le **LMS** (" **Least Mean Square**" ou moindre carré moyen), le **RLS** (" **récurif least square**" ou moindre carré récurif)

L'algorithme **LMS** est un membre important de la famille des algorithmes basés sur le gradient stochastique dont la caractéristique principale est sa simplicité.

L'algorithme **RLS**, basé sur les moindres carrés, possède la caractéristique très intéressante de converger plus rapidement que le **LMS**.



figure

Conclusion et perspective

Plusieurs approches d'estimation et poursuite de la AOA sont possibles et le beamformer est l'une d'entre elles, il consiste à combiner les signaux reçus par les éléments d'une antenne réseau.

Cette méthode de traitement de l'information fait références aux systèmes utilisant des antennes intelligentes.

L'intérêt de ce système est sa capacité à réagir automatiquement à un environnement complexe dont l'interférence est connue à priori.

Il permet de réduire les niveaux des lobes secondaires existants dans la direction de l'interférence, tout en maintenant le lobe principal en direction utile.

Ce système repose sur des antennes réseau, des dispositifs pour calculer les angles d'arrivées et des outils numériques de synthèse qui attribuent des poids aux éléments de l'antenne réseau a fin d'optimiser le signal de sortie selon des techniques de contrôle prédéfinies pour la formation des voies et l'annulation d'interférents.

Une antenne réseau adaptative peut donc être définie comme un réseau capable de modifier son diagramme de rayonnement grâce à un logiciel de synthèse apte à répondre aux spécifications désirées.

L'objectif de notre étude est d'élaborer une technique de synthèse pour la formation de voies et l'annulation d'interférents qui réponde aux spécifications imposées par le système adaptatif.

Les systèmes d'antennes intelligentes nécessitent en générale un calcul des coefficients du réseau en temps réel ce qui n'est pas possible avec une méthode de synthèse classique.

Nous adoptons une nouvelle méthode de synthèse basée sur un modèle adaptatif, le développement de cette méthode est présenté au troisième chapitre, cet outil présente une grande performance au niveau de sa rapidité de convergence.