

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE CONSTANTINE I
Faculté des Sciences de la Technologie
Département d'Electronique

Thèse

Présentée pour obtenir le diplôme de

Doctorat en Sciences

En Electronique

Option : Traitement du signal

Par:

BENCHABANE Abderrazak

Application des réseaux de neurones à l'estimation et à la prédiction des signaux

Devant le jury:

Président :	N. MANSOURI	Prof. Université de Constantine 1
Rapporteur :	A. BENNIA	Prof. Université de Constantine 1
Examineur:	A. CHAREF	Prof. Université de Constantine 1
Examineur :	M-F. HARKAT	Prof. Université de Annaba
Examineur :	A. LACHOURI	M.C.A. Université de Skikda

SOUTENUE LE 22 /10 /2014

*A mes parents, mes frères et mes sœurs,
à ma femme et mes deux petits fils Abderraouf et
Abderrahim*

Je dédie ce mémoire

Remerciements

Tout d'abord je tiens à remercier le Pr. N. MANSOURI, professeur à l'université de Constantine1 qui m'a fait l'honneur de présider le jury de ma thèse. Je remercie également les membres de jury, le Pr. A. CHAREF, professeur à l'université de Constantine1, le Pr. M-F. HARKAT, professeur à l'université de Annaba et le Dr. A. LACHOURI, maître de conférence à l'université de Skikda pour le temps qu'ils ont bien voulu consacrer à lire et à juger le travail en tant que examinateurs de ma thèse.

Mes vifs remerciements sont adressés au professeur A. BENNIA pour la confiance qu'il m'a témoignée en acceptant de diriger mes travaux de thèse. Je lui suis très reconnaissant de m'avoir fait bénéficier tout au long de ce travail de sa grande compétence, de sa rigueur intellectuelle, de sa disponibilité sans faille, et de son efficacité certaine que je n'oublierai jamais. Soyez assuré de ma profonde gratitude.

Je tiens à remercier aussi le professeur A. AHMED-TALEB de l'université de Valenciennes de m'avoir accueilli au sein de son laboratoire durant mon stage de courte durée.

Je remercie également tous les enseignants-chercheurs du département d'électronique et de télécommunication de l'université de Ouargla.

A titre personnel, je remercie chaleureusement mes parents, mes frères et sœurs et sans oublier ma femme pour sa collaboration, son soutien et son aide durant cette thèse.

Résumé

Cette thèse adresse l'utilisation des réseaux de neurones dynamiques dans le problème de l'estimation spectrale. La première phase de notre contribution sert à estimer le spectre de raies d'un signal modélisé par la somme d'un ensemble de sinusoïdes complexes dont les fréquences sont connues en utilisant le réseau de neurones de Hopfield à valeurs complexes. Cette approche a été généralisée pour estimer simultanément les trois paramètres du modèle du signal ; la fréquence, l'amplitude et la phase à partir des échantillons disponibles et sans aucune information a priori sur le signal utilisant un réseau de Hopfield modifié. Cette première phase a été suivie par l'utilisation d'un réseau de neurones dynamique proposé pour l'estimation de la densité spectrale autorégressive. Ensuite ce réseau a été introduit dans la méthode non paramétrique du minimum de variance. L'approche proposée consiste à calculer les coefficients AR par le réseau de Zhang en résolvant les équations de Yule Walker, puis utiliser l'algorithme de Marple *et al.* pour évaluer la densité spectrale de puissance. Suite à la structure Toeplitz du système à résoudre, une architecture réduite du réseau de Zhang a été également proposée. Notre contribution a été achevée par l'utilisation du réseau de Zhang pour inverser la matrice de corrélation pour estimer le spectre d'amplitude d'un signal multidimensionnel.

Les résultats de simulation montrent une supériorité des approches proposées du point de vue rapidité et simplicité d'implémentation par rapport aux méthodes conventionnelles ce qui qualifie ces techniques aux applications en temps réel.

Mots clés : Estimation spectrale, Réseau de Hopfield, Réseau de Zhang, Matrice Toeplitz, Algorithme de Musicus, Algorithme de Marple.

Abstract

This thesis addresses the use of dynamic neural networks in the spectral estimation problem. The first phase of our contribution is used to estimate the line spectrum of a signal modeled by the sum of a set of complex sinusoids whose frequencies are known using the complex-valued Hopfield neural network. This approach has been generalized to estimate the three parameters of the model signal simultaneously; frequency, amplitude and phase from the available samples and without any a priori information on the signal using a modified Hopfield network. This phase was followed by the use of a dynamic neural network proposed for estimating the autoregressive spectral density. Then the last network has been introduced into the non-parametric method of minimum variance. This approach consists in calculating the AR coefficients by the Zhang neural network in solving the Yule Walker equations, and then uses the Marple *et al.* algorithm to assess the power spectral density. Following the Toeplitz structure of the system to be solved, a reduced architecture of the Zhang neural network has also been proposed. Our contribution was completed by using the Zhang network to reverse the correlation matrix to estimate the amplitude of a multidimensional signal spectrum. Simulation results show the superiority of the proposed approaches for the speed and the ease of implementation compared to the conventional methods which qualify these technics for real-time applications.

Key-words: Spectral estimation, Hopfield neural network, Zhang neural network, Toeplitz matrix, Marple algorithm.

ملخص

تتناول هذه الأطروحة استخدام الشبكات العصبية الحيوية في مشكلة التقدير الطيفي. في المرحلة الأولى من مساهمتنا ، استخدمنا الشبكة العصبية لهوبفيلد ذات القيم المركبة في تقدير الخط الطيفي لإشارة مكونة من مجموعة من الجيوب المعقدة ذات الترددات المعروفة. وقد تم تعميم هذا النهج لتقدير المعلومات الثلاث للإشارة ؛ التردد، السعة والطور في وقت واحد من العينات المتاحة و دون أي معلومات مسبقة على الإشارة باستخدام شبكة هوبفيلد المعدلة . وأعقبت هذه المرحلة اقتراح استخدام الشبكة العصبية الحيوية لتقدير الكثافة الطيفية ذات الانحدار الذاتي . ثم تم إدخال هذه الشبكة في الطريقة الغير معلمة للحد الأدنى للتباين. اذ يتكون النهج المقترح في حساب معاملات AR بواسطة شبكة تشانغ في حل معادلات يول- وولكر، ومن ثم استخدام خوارزمية ماربل لتقييم كثافة الطاقة الطيفية . وفقا لهيكل طوبليتز لنظام المعادلات المراد حلها ، تم اقتراح تخفيض بنية شبكة تشانغ . و في الأخير تم إدراج شبكة تشانغ لعكس مصفوفة الارتباط لتقدير اتساع طيف إشارة متعددة الأبعاد. تظهر نتائج المحاكاة تفوق النهج المقترح من وجهة نظر السرعة و سهولة التنفيذ مقارنة بالطرق التقليدية مما يؤهلها للتطبيقات الفورية.

كلمات مفتاحية: التقدير الطيفي، الشبكة العصبية لهوبفيلد، الشبكة العصبية لتشانغ، مصفوفة طوبليتز، خوارزمية ماربل.

Liste des figures

1.1. Méthode de Bartlett	9
1.2. Méthode de Welch	9
1.3. Méthode de Welch symétrique.....	10
1.4. Méthode de Welch anti-symétrique.....	10
1.5. Structure d'un processus $ARMA(p, q)$	11
1.6. Structure d'un processus $AR(p)$	12
1.7. Structure d'un processus $MA(q)$	13
1.8. Segmentation du vecteur d'observations.....	21
2.1. Le modèle de neurone formel.....	27
2.2. Structure des réseaux de neurones.....	28
2.3. Structure du réseau de Hopfield.....	29
2.4. Architecture du réseau de Hopfield...	32
2.5. Moyenne et écart-type des amplitudes estimées.	35
2.6. Erreur quadratique moyenne des amplitudes estimées.....	36
2.7. Trajectoire des amplitudes estimées.....	36
2.8. Trajectoire des amplitudes estimées en utilisant la fonction sigmoïde.....	37
2.9. Schéma synoptique du réseau de neurones	40
2.10. Architecture de chaque estimateur neuronal de la figure (2.9).....	40
2.11. Surfaces et contours de la fonction de coût	41
2.12. Performances du réseau vis-à-vis le SNR pour l'estimation des amplitudes	42
2.13. Trajectoires des amplitudes estimées et erreur résiduelle pour un $SNR = 30dB$	42
2.14. Performances du réseau vis-à-vis le SNR pour l'estimation des phases	43
2.15. Trajectoires des phases estimées et erreur résiduelle pour un $SNR = 30dB$	43
2.16. Performances du réseau vis-à-vis le SNR pour l'estimation des fréquences.....	44
2.17. Trajectoires des fréquences estimées et erreur résiduelle pour un $SNR = 30dB$	44
2.18. Performances du réseau vis-à-vis le SNR pour l'estimation des trois paramètres...	45
3.1. Architecture du réseau RNG.....	50
3.2. Architecture du réseau RNG pour l'estimation spectrale.....	50

3.3. Architecture du réseau RNZ.....	52
3.4. Architecture du réseau RNZ pour l'estimation spectrale.....	52
3.5. Fonction sigmoïde.....	53
3.6. Fonction signe-puissance.....	53
3.7. Schéma d'implémentation du modèle basé sur le gradient.....	54
3.8. Schéma d'implémentation du modèle de Zhang.....	54
3.9. Performances du réseau RNZ pour $\sigma^2 = 10^{-4}$	55
3.10. Performances du réseau RNZ pour $\sigma^2 = 10^{-2}$	55
3.11. Performances du RNZ avec la fonction linéaire pour différentes valeurs de γ	56
3.12. Performances du RNZ avec la fonction sigmoïde pour différentes valeurs de ξ	57
3.13. Performances du réseau basé sur le gradient suivant la valeur de γ	58
3.14. Pas de discrétisation optimale pour atteindre une erreur de convergence de 10^{-10} ...	59
3.15. Performances du réseau de Zhang suivant la valeur de σ_e^2	60
3.16. Performances du réseau RNG suivant la valeur de σ_e^2	61
3.17. Implémentation du réseau de Zhang pour l'estimation spectrale.....	63
3.18. Schéma de simulation pour comparer les performances des deux réseaux.....	63
3.19. Performances des deux réseaux.	64
3.20. Densité spectrale estimée par les deux réseaux.	64
3.21. Implémentation du réseau de Zhang pour l'estimation spectrale.....	65
3.22. comparaison des trajectoires des coefficients obtenus avec la méthode de Burg....	55
3.23. Densité spectrale estimée par les différentes méthodes après 313 segments.....	66
4.1. Schéma synoptique de l'estimateur neuronal AR/Marple.....	73
4.2. Architecture du réseau RNZD.....	73
4.3. Schéma de simulation du réseau RNZD-Marple.....	74
4.4. Résultats de simulation du modèle RNZD-Marple pour $p = 4$ et différentes valeurs de SNR	75
4.5. Résultats de simulation du modèle RNZD-Marple pour $p = 8$ et différentes valeurs de SNR	76
4.6. Schéma synoptique de l'algorithme rapide pour calculer \mathbf{Ax}	79
4.7. Réseau RNZD réduit.....	89

4.8. Nombre de multiplications nécessaires pour RNZD et sa version réduite.....	80
4.9. Schéma de simulation du réseau RNZD réduit	81
4.10. Résultats de simulation des modèles RNZD-Marple et RNZDR-Marple.....	81
4.11. Complexité du réseau RNZD réduit.....	81
5.1. Segmentation de la matrice d'observations.....	86
5.2. Architecture du réseau RNZD1.....	92
5.3. Architecture du réseau RNZD2.....	94
5.4. Schéma synoptique de l'estimateur neuronal basé sur le réseau RNZD2.....	96
5.5. Schéma d'implémentation du réseau RNZD1.....	97
5.6. Schéma d'implémentation du réseau RNZD2.....	98
5.7. Densité spectrale du signal 1D estimée $M = 8$ et $SNR = 30dB$	101
5.8. Erreur de convergence pour le cas des signaux 1D.....	101
5.9. Signal de test.....	101
5.10. Erreur de convergence pour le cas des signaux 2D.....	102
5.11. Densité spectrale du signal 2D estimée $M \times \bar{M} = 8 \times 8$ et $SNR = 30dB$	102
5.12. Module, phase et spectre des données de l'avion MIG-25.....	103
5.13. Erreur de convergence pour le cas des signaux 2D réels.....	103
5.14. Image reconstruite avec un filtre $M \times \bar{M} = 4 \times 4$	104
5.15. Projections de la densité spectrale estimée via la méthode de Capon 3D.....	106
5.16. Projections de la densité spectrale estimée via la méthode APES 3D.....	107
5.17. Erreur de convergence pour le cas des signaux 3D.....	107

Liste des tableaux

3.1. Les valeurs de k_{\min} correspondant aux γ_{opt} pour différentes valeurs de σ_e^2	61
3.2. Résumé des résultats obtenus avec les réseaux proposés et autres méthodes.....	62
3.3. Propriétés statistiques des coefficients estimés.....	66
4.1. Comparaison de la complexité avec d'autres méthodes [20].....	82
5.1. Influence du coefficient τ et la longueur du filtre M sur la convergence du réseau RNZD1.....	99
5.2. Fréquences estimées pour les quatre méthodes avec $N = 16$, $M = \bar{M} = \overline{\bar{M}} = 8$	105

Table des matières

Remerciements	i
Résumé	ii
Abstract	iii
ملخص	iv
Liste des figures	v
Liste des tableaux	viii
Table des matières	ix
Introduction Générale	1

Chapitre 1

Méthodes d'estimation Spectrale : Etat de l'art

1.1. Introduction	6
1.2. Méthodes non-paramétriques.....	7
1.2.1. Périodogramme	7
1.2.2. Corrélogramme.....	7
1.2.3. Périodogramme modifié.....	8
1.2.4. Méthode de Blackman-Tukey	8
1.2.5. Méthode de Bartlett.....	8
1.2.6. Méthode de Welch.....	9
1.2.7. Méthode de Welch modifiée.....	10
1.3. Méthodes paramétriques.....	11
1.3.1. Modèles à fonction de transfert rationnelle	11
1.3.2. Modèle à décomposition harmonique.....	19
1.4. Méthodes non-paramétriques basé sur le filtrage adaptatif.....	20
1.4.1. Méthode de Capon.....	22
1.4.2. Méthode APES.....	24
1.5. Conclusion.....	25

Chapitre 2

Estimation du spectre de raies par le réseau de neurones de Hopfield

2.1. Introduction.....	26
2.2. Réseau de neurones.....	26
2.2.1. Définition des réseaux de neurones.....	26
2.2.2. Types de réseaux de neurones.....	27
2.2.3. Réseaux de neurones de Hopfield pour l'optimisation.....	28

2.3. Formulation du problème de l'estimation du spectre de raies.....	29
2.4. Estimateur neuronal.....	31
2.4.1. Réseau de Hopfield à valeurs réelles.....	31
2.4.2. Réseau de Hopfield à valeurs complexes.....	33
2.5. Résultats de simulation.....	34
2.5.1. Fonction d'activation linéaire.....	35
2.5.2. Fonction d'activation sigmoïde.....	36
2.6. Réseau de Hopfield modifié.....	37
2.6.1. Architecture du réseau de Hopfield modifié.....	39
2.6.2. Résultats de Simulation.....	40
2.7. Conclusion.	45

Chapitre 3

Estimateur neuronal d'un modèle auto régressif

3.1. Introduction.....	47
3.2. Positionnement du problème	47
3.3. Estimateur neuronal AR par gradient.....	48
3.4. Estimateur AR par le réseau de Zhang.....	51
3.4.1. Fonctions d'activation.....	52
3.5. Résultats de simulation pour l'estimation des coefficients AR.....	53
3.5.1. Choix de la fonction d'activation.....	54
3.5.2. Choix du coefficient de convergence γ	55
3.5.3. Choix du pas de discrétisation h	59
3.5.4. Effet de la variance du bruit.....	60
3.5.5. Comparaison avec d'autres méthodes.....	62
3.6. Estimation de la densité spectrale.....	62
3.7. Conclusion.....	66

Chapitre 04

Estimateur neuronal par minimum de variance

4.1. Introduction.....	67
4.2. Estimateur spectral par minimum de variance.....	67
4.2.1. Principe.	67
4.2.2. Algorithme de Musicus..	69
4.2.3. Algorithme de Marple..	71

4.3. Réseau de neurones AR/Marple.	72
4.3.1. Principe du réseau proposé.	72
4.3.2. Résultats de simulation	73
4.4. Architecture réduite proposée	77
4.4.1. Calcul rapide du produit matrice par vecteur	77
4.4.2. Réseau de neurones proposé.....	79
4.4.3. Complexité du réseau.....	80
4.5. Performances de la méthode proposée.....	80
4.5.1. Précision.....	80
4.5.2. Comparaison de la complexité.....	81
4.6. Conclusion.....	83

Chapitre 5

Estimation spectrale multidimensionnelle par le réseau de Zhang

5.1. Introduction	84
5.2. Méthodes de Capon et APES multidimensionnelles.....	84
5.2.1. Spectre des signaux 2D.....	84
5.2.2. Spectre des signaux 3D.....	88
5.3. Inversion de la matrice de corrélation par le réseau de Zhang.....	90
5.3.1. Réseau de neurones à dynamique explicite.....	90
5.3.2. Réseau de neurones à dynamique implicite.....	93
5.4. Architecture de l'estimateur spectral.....	94
5.5. Implémentation de l'estimateur proposé sous Simulink.....	96
5.5.1. Estimateur spectral via RNZD1.....	96
5.5.2. Estimateur spectral via RNZD2.....	97
5.6. Exemples de simulation.....	98
5.6.1. Signaux monodimensionnels.....	99
5.6.2. Signaux bidimensionnels..	101
5.6.3. Signaux tridimensionnels.....	105
5.7. Conclusion.....	108
Conclusion Générale.....	109
Bibliographie.....	111

Introduction Générale

L'objectif majeur de l'estimation spectrale consiste à extraire le contenu fréquentiel d'un processus aléatoire [1]. L'estimation spectrale fût apparue dès le 18^{ème} siècle avec la naissance de la célèbre théorie de Fourier. Selon cette théorie, tout signal peut être représenté par une somme infinie de fonctions sinus et cosinus. La théorie statistique de l'estimation spectrale n'a commencé que dès le début des années cinquante quand Tukey a introduit les méthodes numériques pour le calcul du spectre à partir un nombre d'échantillons du signal. La découverte de l'algorithme rapide de la transformée de Fourier discrète a enrichi énormément le domaine de l'estimation spectrale [2]. Juste peu de temps après, Burg a proposé une nouvelle approche de l'estimation spectrale basée sur le principe du maximum d'entropie [2]. Cet algorithme s'est suivi par une grande diversité de méthodes de calcul dans les trois dernières décades [3-4].

D'une manière générale, les méthodes de l'estimation spectrale se divisent en deux grandes familles ; les méthodes non paramétriques et les méthodes paramétriques [4-5]. Les méthodes non paramétriques sont toutes basées sur la transformée de Fourier discrète d'où la résolution fréquentielle ne doit pas dépasser en aucun cas la limite de Fourier qui est fonction du nombre d'échantillons disponibles. Cependant, ces méthodes sont plus robustes du fait qu'elles n'exigent aucune connaissance a priori sur le signal [4]. La deuxième classe englobe les méthodes paramétriques où le processus est supposé issu d'un modèle dont on cherche ses paramètres. Ces méthodes sont connues par une haute résolution fréquentielle. Cependant l'inconvénient de ces méthodes réside dans le choix du modèle. Dans la pratique où souvent le modèle n'est pas connu, les méthodes non paramétriques sont avantagées à condition d'utiliser un nombre plus grand d'échantillons pour augmenter la résolution [5-6].

Bénéficiant de la résolution fréquentielle des méthodes paramétriques, et la robustesse des méthodes non paramétriques, une nouvelle classe de méthodes issues de la méthode du minimum de variance de Capon est née [7]. Dans cette classe, le processus est supposé issu d'un filtre adapté dont les coefficients sont calculés en minimisant la puissance à la sortie du filtre [8]. Dans beaucoup d'applications où l'information de la phase est primordiale, la méthode du minimum de variance n'est plus applicable, d'où des recherches récentes ont été

élaborées pour estimer le spectre d'amplitude et de phase ou en d'autres termes les amplitudes complexes des raies spectrales.

Stoica *et al* ont proposé deux méthodes qui découlent directement de la méthode du minimum de variance, la première à l'avantage de localiser les fréquences avec une grande précision, cependant les amplitudes des raies estimées sont fortement biaisées [9]. Quant à la deuxième méthode APES (amplitude and phase estimation), elle donne des pics éloignés des raies réels mais elle permet une estimation des amplitudes plus précises aux vraies fréquences [9-10]. En combinant ces deux méthodes, Stoica *et al* ont proposé une autre méthode CAPES (Capon-amplitude and phase estimation) qui estime en premier lieu les fréquences par la méthode de Capon, puis utiliser la méthode APES pour estimer les amplitudes aux fréquences trouvées [11].

On note ici que la méthode du minimum de variance nécessite l'inversion d'une matrice Toeplitz, tandis que les deux autres méthodes, elles nécessitent l'inversion d'une matrice quazi-Toeplitz. Du fait que ces méthodes nécessitent une inversion de matrice, dont le temps de calcul est important surtout pour les matrices de grandes tailles, plusieurs algorithmes rapides ont été proposés [12-20]. Musicus [12] est le premier qui a utilisé la structure Toeplitz de la matrice de corrélation pour calculer son inverse utilisant la formule de Simenculi. Son algorithme se déroule en trois étapes ; le calcul des coefficients autorégressifs (AR) par l'algorithme de Levinson, puis le calcul de la corrélation de ces paramètres et enfin l'évaluation de la densité spectrale utilisant la transformée de Fourier rapide.

Récemment, Marple *et al.* [20] ont proposé un algorithme ultra rapide pour le calcul de la DSP, cet algorithme se fait seulement en deux étapes, la première sert à calculer les coefficients AR comme dans l'algorithme de Musicus. La deuxième étape permet d'évaluer la densité spectrale de puissance (DSP) en utilisant la transformée de Fourier rapide et en évitant totalement le calcul de la corrélation des coefficients AR. Ces deux derniers algorithmes concernent la méthode du minimum de variance suite à la structure Toeplitz de la matrice de corrélation. Pour la méthode de Capon et APES pour l'estimation du spectre d'amplitude et où la matrice à inverser est Toeplitz prés, Li & all [14-15] ont proposé un algorithme rapide utilisant la formule de Cholesky. Quant à Glentis [18-19], il a proposé d'utiliser le rang de déplacement des matrices structurées pour calculer la matrice inverse.

Les réseaux de neurones constituent un outil de calcul très puissant suite à leur nature parallèle, permettent d'exécuter un nombre de calculs énorme dans un temps extrêmement court [21-25]. Suite à cette rapidité, les réseaux de neurones offrent une supériorité par rapport aux méthodes classiques dans plusieurs applications [26-40]. Dans les trois dernières décades, les réseaux de neurones ont été appliqués avec un grand succès dans une variété de domaines d'engineering comme le traitement du signal [26-28], le traitement d'image [29-31] l'estimation spectrale [32-40] et autres.

Objectif de la thèse

Notre contribution consiste à introduire les réseaux de neurones récurrents dans le problème de l'estimation spectrale utilisant les méthodes paramétriques (le modèle AR et le modèle de sinusoides) et les méthodes non paramétriques utilisant la méthode du minimum de variance et ces dérivées [7,9].

La première démarche sert à estimer le spectre de raies d'un signal modélisé par la somme d'un ensemble de sinusoides complexes dont les fréquences sont connues en utilisant un réseau de neurones de Hopfield à valeurs complexes [41-42]. Ceci est réalisé en concevant une fonction de coût à minimiser semblable à l'énergie du réseau de Hopfield et à partir de laquelle on identifie les paramètres du réseau ; les entrées, les sorties et poids de connections. Puisque les paramètres du réseau sont des valeurs complexes, deux architectures sont élaborées, la première est similaire à celle du réseau à valeurs réelles obtenue par la construction d'un vecteur de données réelles à partir de la partie réelle et la partie imaginaire du signal [43]. La deuxième version est constituée de deux réseaux à valeurs réelles, identiques, l'un pour la partie réelle, l'autre pour la partie imaginaire. Puis cette première démarche a été généralisée pour estimer simultanément les trois paramètres du modèle du signal ; la fréquence, l'amplitude et la phase à partir des échantillons disponibles et sans aucune information a priori sur le signal utilisant un réseau de Hopfield modifié. Suite à l'existence d'un grand nombre de minimums locaux, les fréquences initiales doivent être initialisées par des valeurs très proches des vraies fréquences. La deuxième phase de cette démarche est le modèle neuronal Auto Régressif (AR). Deux réseaux de neurones ont été proposés pour estimer les paramètres du modèle au lieu de l'algorithme de Levinson, un réseau basé sur le gradient, l'autre un réseau récent proposé par Y. Zhang basé sur la minimisation d'un vecteur-erreur au lieu de la norme de l'erreur [44]. La figure ci-dessous donne une illustration de la première démarche de notre contribution dans la présente thèse.

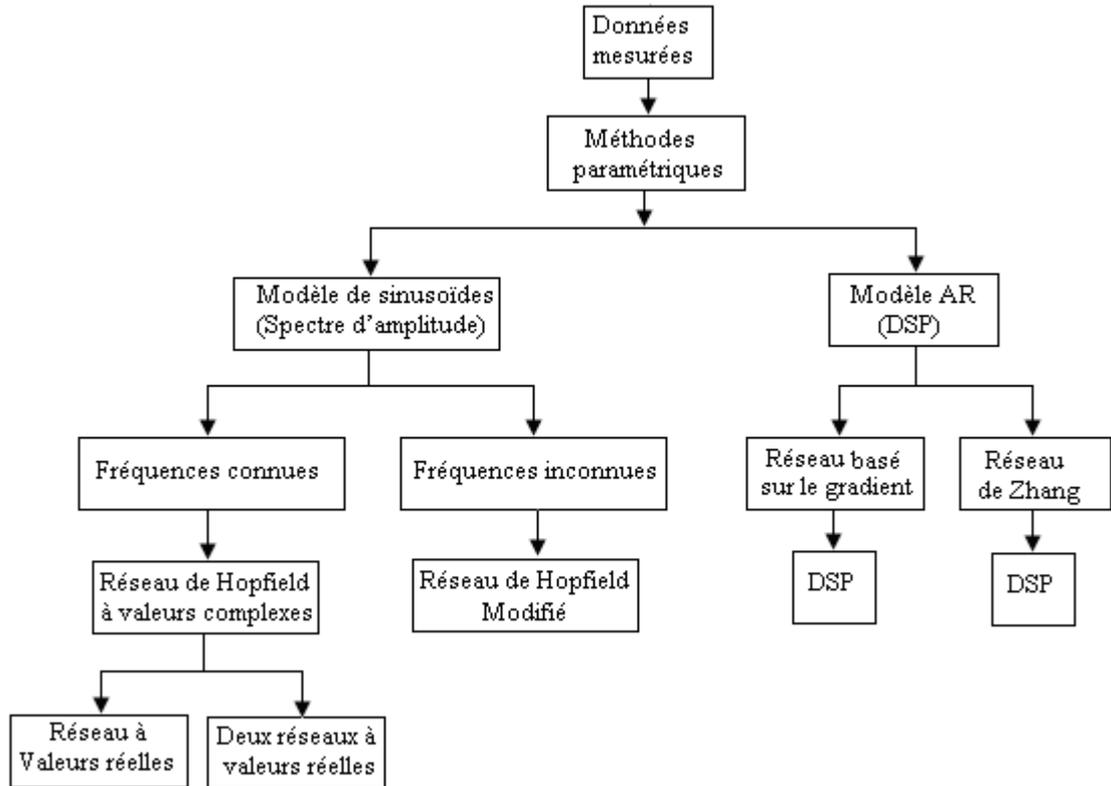


Figure 1. Organigramme de la contribution pour les méthodes paramétriques.

La deuxième démarche concerne les méthodes non paramétriques. Elle est initiée par l'introduction du réseau de Zhang dans la méthode du minimum de variance. L'approche proposée consiste à calculer les coefficients AR par le réseau de Zhang en résolvant les équations de Yule Walker, puis utiliser l'algorithme de Marple *et al.* [20] pour évaluer la densité spectrale de puissance. Suite à la structure Toeplitz du système à résoudre, une architecture réduite du réseau de Zhang a été proposée.

Concernant la méthode de Capon et la méthode APES pour l'estimation du spectre d'amplitude, le réseau de Zhang a été utilisé mais cette fois-ci pour inverser la matrice de corrélation dans l'expression du spectre d'amplitude [45]. Suivant la dynamique explicite ou implicite du réseau, deux modèles de réseau de Zhang ont été proposés. Parmi les applications les plus intéressantes de l'estimation du spectre d'amplitude est la reconstruction des images radar à synthèse d'ouverture et l'estimation des signaux à résonance magnétique nucléaire. Pour cela le réseau de Zhang a été étendu pour les signaux 2D et 3D suite à la taille importante de la matrice de corrélation dans ces cas [46-52]. La figure ci-dessous résume la deuxième démarche.

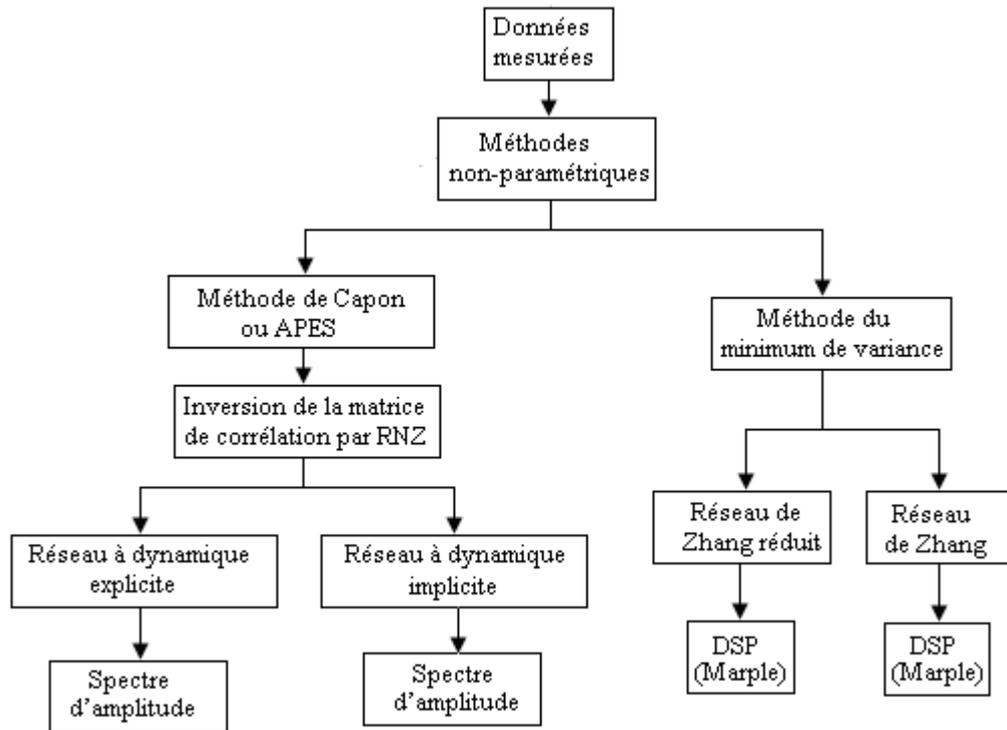


Figure 2. Organigramme de la contribution pour les méthodes non paramétriques

Organisation du manuscrit

Le travail présenté dans cette thèse est articulé autour de l'estimation spectrale utilisant les réseaux de neurones dynamiques. Il est organisé en cinq chapitres :

Dans le chapitre 1 on présente les différentes méthodes de l'estimation spectrales de l'estimation spectrale. Le chapitre 2 introduit le réseau de Hopfield à valeurs complexes pour estimer les paramètres sinusoidaux modélisant un vecteur d'observations. Dans le chapitre 3 on présente deux estimateurs neuronaux pour l'estimation de la densité spectrale de puissance d'un modèle autorégressif. Dans le chapitre 4 on présente un estimateur de la densité spectrale de puissance d'un vecteur d'observations utilisant le réseau de Zhang pour résoudre les équations de Yule-Walker et l'algorithme de Marple pour évaluer la densité spectrale. Puis on a une architecture réduite du réseau de Zhang utilisant la structure Toeplitz de la matrice de corrélation. Le dernier chapitre de ce manuscrit concerne l'utilisation du réseau de Zhang dans la méthode de Capon et la méthode APES pour l'estimation du spectre d'amplitude des signaux multidimensionnels.

Chapitre 1

Méthodes d'estimation Spectrale : Etat de l'art

1.1. Introduction

Les méthodes d'estimation spectrale se divisent en deux grandes classes de méthodes; les méthodes non paramétriques et les méthodes paramétriques. Les méthodes non paramétriques sont toutes basées sur la transformée de Fourier discrète [53]. De ce fait, ces méthodes sont efficaces du point de vue complexité lorsque l'algorithme de la transformée de Fourier rapide est utilisé [1]. L'inconvénient majeur de ces méthodes réside dans la faible résolution fréquentielle quand le nombre d'échantillons est petit. De l'autre côté, ces méthodes ne font aucune hypothèse a priori sur la forme spectrale et par conséquent elles sont très robustes [6,53].

La deuxième classe des méthodes inclue les méthodes paramétriques. Ces méthodes reposent sur l'idée que le processus est issu d'un modèle dont on cherche ses paramètres. Une fois les paramètres sont trouvés, il suffit d'introduire ces paramètres dans les expressions mathématiques de la densité spectrale. Ces méthodes sont caractérisées par une résolution fréquentielle plus meilleure que les méthodes non paramétriques [1-6]. Lorsque le modèle est parfaitement connu, la densité spectrale estimée par les méthodes paramétrique sera correcte, donc ces méthodes sont avantageuses. Mais dans le cas contraire, on a recours aux méthodes non paramétriques à condition d'utiliser un nombre d'observations suffisant [1].

Motivé par la robustesse des méthodes non paramétriques et la haute résolution fréquentielle des méthodes paramétriques, une nouvelle classe des méthodes est née [7-8]. Dans cette classe, le processus est supposé issu d'un filtre à réponse impulsionnelle finie dont les coefficients sont calculés en minimisant la variance du signal à la sortie du filtre, c'est pour cette raison que ces méthodes sont dites quelques fois les méthodes à variances minimale. Dans ce qui suit, on suppose la disponibilité d'une série de N échantillons d'un signal $x(n)$. L'objectif est de trouver une estimation de la densité spectrale à partir de ces données sous l'hypothèse de la stationnarité et l'ergodicité du processus [3].

1.2. Méthodes non-paramétriques

1.2.1. Périodogramme

La méthode du périodogramme fut introduite par Schuster en 1898. Il utilise les données disponibles pour estimer la densité spectrale du signal [1,6,53]:

$$\hat{P}_{per}(f) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x(n) e^{-j2\pi n f} \right|^2 = \frac{1}{N} |X(f)|^2 \quad (1.1)$$

où $X(f)$ est la transformée de Fourier discrète du signal $x(n)$.

Du fait de la troncature du signal, le périodogramme est en fait la convolution du spectre par une fonction sinus cardinale. Cette opération induit deux phénomènes principaux: élargissement du lobe principale ce qui provoque une perte de résolution fréquentielle, et l'apparition de lobes secondaires ce qui influe sur les hautes fréquences. Pour remédier, on utilise en général des fenêtres permettant d'atténuer les lobes secondaires au détriment d'un élargissement du lobe principal.

1.2.2. Corrélogramme

Une autre approche consiste à utiliser la définition du spectre à partir de la fonction de corrélation. Du fait qu'on ne dispose que d'un nombre fini d'observations, la vraie fonction d'autocorrélation ne peut pas être calculée et on est confronté au problème de leur estimation. On estime alors le spectre comme [53] :

$$\hat{P}_{cor}(f) = \sum_{k=-(N-1)}^{N-1} \hat{r}_{xx}(k) e^{-j2\pi k f} \quad (1.2)$$

où $\hat{r}_{xx}(m)$ est un estimateur de la fonction de l'autocorrélation qui est donné par :

$$\hat{r}_{xx}(k) = \frac{1}{N} \sum_{n=0}^{N-k-1} x^*(n) x(n+k) \quad k = 0, \dots, N-1 \quad (1.3)$$

Puisque $\hat{r}_{xx}(m)$ est un estimateur de la vraie fonction d'autocorrélation, on cherche à déterminer sa qualité par le calcul de sa moyenne qui est donnée par:

$$E[\hat{r}_{xx}(k)] = \frac{1}{N} \sum_{n=0}^{N-k-1} E[x^*(n) x(n+k)] = \frac{N-k}{N} r_{xx}(k) \quad (1.4)$$

Du fait que pour $k \geq N$, $\hat{r}_{xx}(k) = 0$, et $\hat{r}_{xx}(-k) = \hat{r}_{xx}^*(k)$, on obtient:

$$E[\hat{r}_{xx}(k)] = w_B(k) r_{xx}(k) \quad (1.5)$$

où :

$$w_B(k) = \begin{cases} \frac{N-|k|}{N}, & |k| \leq N \\ 0, & |k| > N \end{cases} \quad (1.6)$$

est une fenêtre triangulaire. Ainsi, $\hat{r}_{xx}(k)$ est un estimateur biaisé de la fonction de corrélation ; par conséquent, l'estimateur $\hat{P}_{cor}(f)$ est biaisé aussi.

Pour supprimer le biais, l'estimateur non biaisé de la fonction de l'autocorrélation est utilisé:

$$\hat{r}_{xx}(k) = \frac{1}{N-|k|} \sum_{n=0}^{N-k-1} x^*(n)x(n+k) \quad k = 0, \dots, N-1 \quad (1.7)$$

1.2.3. Périodogramme modifié

Dans le périodogramme modifié, la fenêtre rectangulaire utilisée pour estimer le périodogramme $\hat{P}_{per}(f)$ est remplacée par une fenêtre quelconque $w(n)$, le périodogramme résultant est de la forme [6] :

$$\hat{P}_{perm}(f) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x(n) w(n) e^{-j2\pi n f} \right|^2 \quad (1.8)$$

1.2.4. Méthode de Blackman-Tukey

Puisque les valeurs de la fonction d'autocorrélation sont petites pour les valeurs extrêmes du retard, il est recommandé de ne garder que les coefficients les plus importants. L'estimateur de Blackman-Tukey consiste à appliquer cette hypothèse [6]:

$$\hat{P}_{BT}(f) = \sum_{k=-(L-1)}^{L-1} \hat{r}_{xx}(k) w(k) e^{-j2\pi k f} \quad (1.9)$$

où $w(k)$ est une fenêtre dont les valeurs sont nulles pour $|k| > L-1$ et $L \ll N$.

1.2.5. Méthode de Bartlett

La méthode de Bartlett consiste à fragmenter le vecteur d'observations $x(n)$ de longueur N en $K = N/L$ sous-intervalles de même longueur L sans chevauchement et de calculer le périodogramme de chaque segment [6]. L'estimateur de Bartlett consiste en la moyenne de tous les K périodogrammes. Ceci est illustré par la figure (1.1).

$$\hat{P}_B(f) = \frac{1}{K} \sum_{k=0}^{K-1} \left| \frac{1}{L} \sum_{l=0}^{L-1} x(kL+l) e^{-j2\pi l f} \right|^2 \quad (1.10)$$

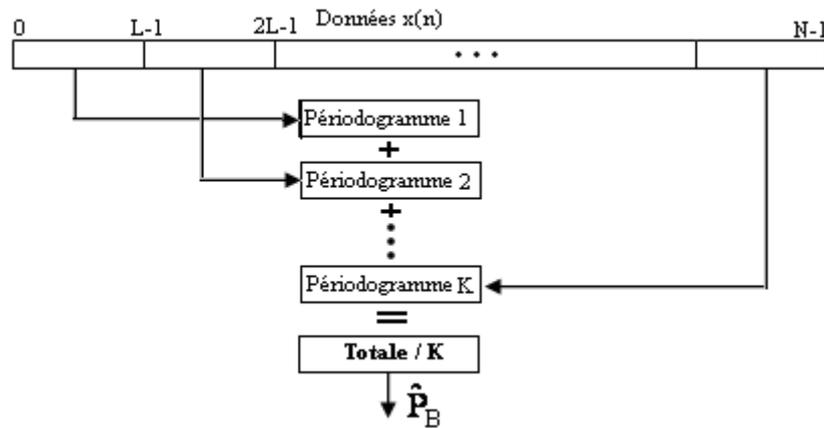


Figure 1.1. Méthode de Bartlett.

1.2.6. Méthode de Welch

En 1967, Welch proposa deux modifications à la méthode de Bartlett. La première est de permettre aux segments d'observations de se recouvrir et la seconde est de rajouter une fenêtre à chacune de ces segments produisant ainsi un ensemble de périodogrammes modifiés (voir figure (1.2) [54]). Donc la densité spectrale estimée par la méthode de Welch consiste à fragmenter le vecteur d'observations $x(n)$ de longueur N en $K = N/L$ sous-intervalles de même longueur L avec chevauchement, puis d'appliquer une fenêtre de pondération sur chacun de ces K sous-intervalles donnant une suite des segments pondérés $\{\tilde{x}(n)\}$. La moyenne des K périodogrammes constitue la densité spectrale de Welch.

$$\hat{P}_W(f) = \frac{1}{K} \sum_{k=0}^{K-1} \left| \frac{1}{L} \sum_{l=0}^{L-1} \tilde{x}(kL+l) e^{-j2\pi f l} \right|^2 \quad (1.11)$$

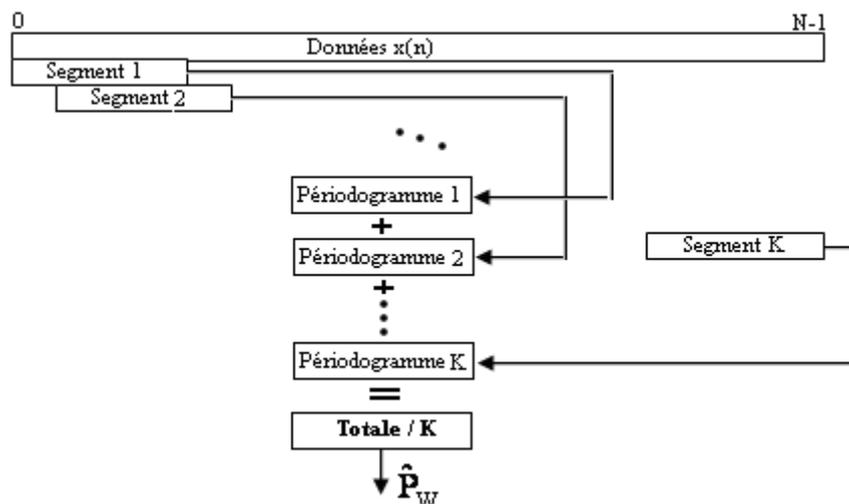


Figure 1.2. Méthode de Welch

1.2.7. Méthode de Welch modifiée

Dans la méthode de Welch, la longueur des segments n'est pas suffisamment grande, ce qui entraîne une faible résolution. Pour remédier, une modification consiste à augmenter progressivement la taille des segments pour avoir une résolution fréquentielle meilleure. On note ici que la pondération des segments par des fenêtres peut être incorporée. Deux types d'implémentation sont envisagés : la méthode symétrique représentée dans la figure (1.3) et la méthode anti-symétrique représentée dans la figure (1.4) [6].

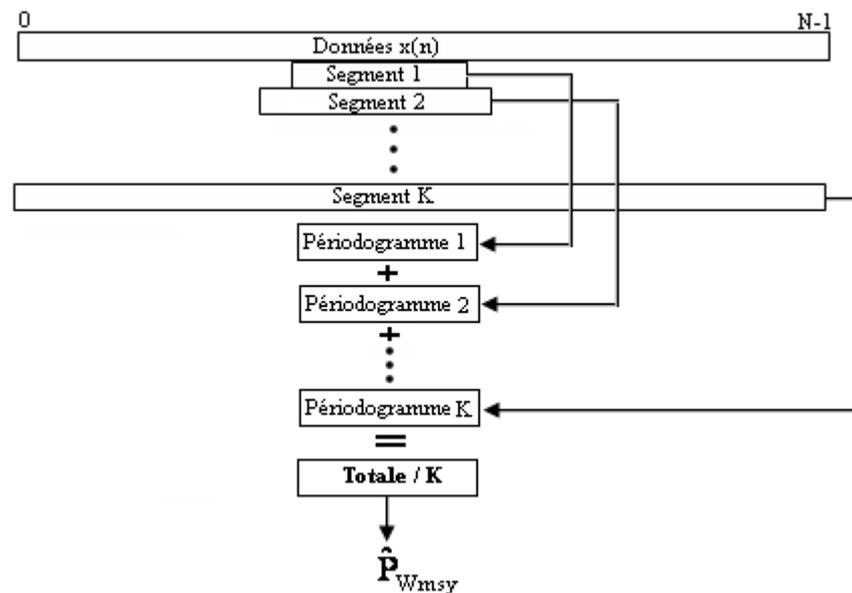


Figure 1.3. Méthode de Welch symétrique.

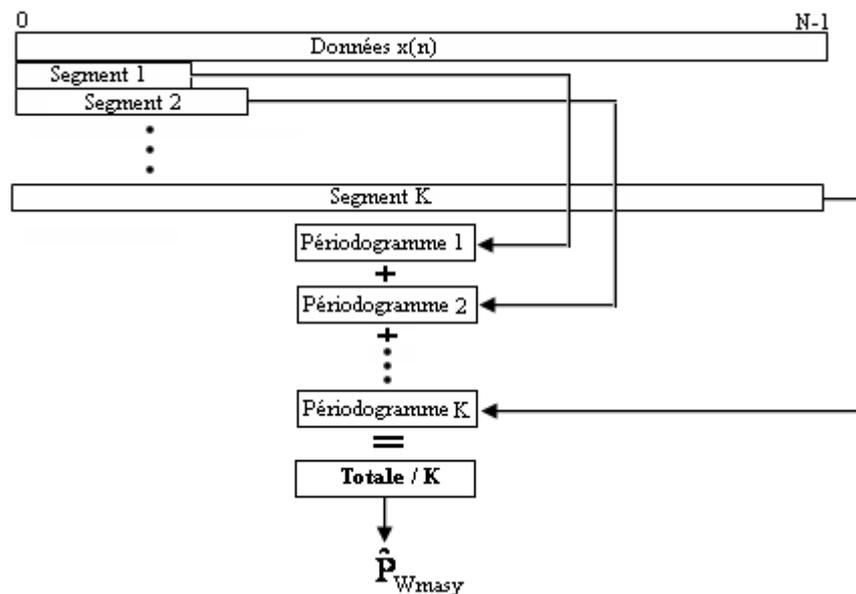


Figure 1.4. Méthode de Welch anti-symétrique.

1.3. Méthodes paramétriques

Ces méthodes sont basées principalement sur la modélisation paramétrique permettent l'obtention d'une meilleure estimation de la densité spectrale de puissance. Ainsi les méthodes paramétriques se ramènent à la sélection d'un modèle, l'estimation des paramètres de ce modèle en utilisant les échantillons disponibles et la substitution des valeurs des paramètres estimés dans l'expression théorique de la densité spectrale [4,6,56].

1.3.1. Modèles à fonction de transfert rationnelle

Dans la pratique, beaucoup de processus aléatoires sont approximés par un modèle de fonction de transfert rationnelle où l'entrée $e(n)$ et la sortie $x(n)$ de ce modèle sont liées par l'équation:

$$x(n) = -\sum_{k=1}^p a_k X(n-k) + \sum_{k=0}^q b_k e(n-k) \tag{1.12}$$

Ce modèle est le plus général, il est appelé le modèle ARMA (Autoregressive moving average). La figure (1.5) montre un processus ARMA d'ordre (p, q) .

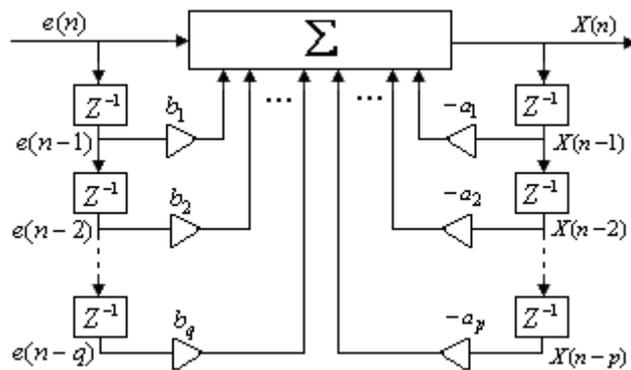


Figure.1.5. Structure d'un processus $ARMA(p, q)$.

Le processus $x(n)$ peut être vu comme la sortie d'un filtre linéaire, dont l'entrée est $e(n)$ et de fonction de transfert $H(z)$ telle que :

$$H(Z) = \frac{B(z)}{A(z)} = \frac{1 + \sum_{l=1}^q b_l Z^{-l}}{1 + \sum_{k=1}^p a_k Z^{-k}} \tag{1.13}$$

Pour assurer que le processus $x(n)$ soit stationnaire, il faut que $H(z)$ soit stable et causal; que $H(z)$ ait ses pôles à l'intérieur du cercle unité du plan z . Si on considère que l'entrée $e(n)$ du

filtre est un bruit blanc d'espérance nulle et de variance σ_e^2 , la densité spectrale de la sortie est évaluée sur le cercle unitaire ($z = e^{j2\pi f}$) :

$$P_{ARMA}(f) = P_{xx}(z) \Big|_{z=e^{j2\pi f}} = \sigma_e^2 \left| \frac{B(f)}{A(f)} \right|^2 \quad (1.14)$$

L'estimation de la densité spectrale revient à la détermination des paramètres \hat{a}_k , \hat{b}_k et $\hat{\sigma}_e^2$.

Un processus $AR(p)$ est un cas particulier du modèle $ARMA(p,q)$ pour $b_k = 0$, $k = 1, 2, \dots, q$. Il est donné par l'équation aux différences suivante [4,6] :

$$x(n) = -\sum_{k=1}^p a_k x(n-k) + e(n) \quad (1.15)$$

La fonction de transfert du filtre est :

$$H(Z) = \frac{1}{A(z)} = \frac{1}{1 + \sum_{k=1}^p a_k z^{-k}} \quad (1.16)$$

La structure du filtre est représentée sur la figure (1.6).

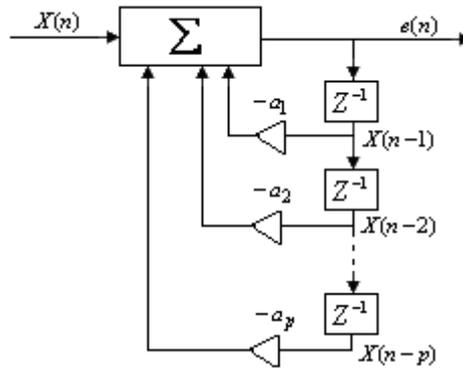


Figure 1.6. Structure d'un processus $AR(p)$.

La densité spectrale de puissance à la sortie du filtre est donnée par :

$$P_{AR}(f) = \frac{\sigma_e^2}{\left| 1 + \sum_{k=1}^p a_k e^{-j2\pi f k} \right|^2} \quad (1.17)$$

Le processus $ARMA(p,q)$ est dit un processus $MA(q)$ s'il est défini par [4,6] :

$$X(n) = \sum_{k=0}^q b_k e(n-k), \quad b_0 = 1. \quad (1.18)$$

En appliquant la transformée en Z, on aura :

$$H(z) = B(Z) = 1 + \sum_{k=1}^p b_k z^{-k} \quad (1.19)$$

La figure 1.7 montre la structure d'un processus MA d'ordre q

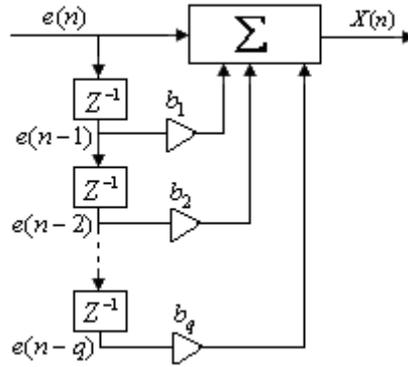


Figure 1.7. Structure d'un processus $MA(q)$

La densité spectrale du processus de sortie $MA(q)$ est alors de la forme:

$$P_{MA}(f) = \sigma_e^2 \left| 1 + \sum_{k=1}^q b_k e^{-j2\pi f k} \right|^2 \quad (1.20)$$

1.3.1.1. Densité spectrale estimée basée sur le modèle autorégressif

Pour trouver la densité spectrale $P_{AR}(f)$, nous devons estimer les paramètres \hat{a}_k et $\hat{\sigma}_e^2$ utilisant différentes méthodes [4,6,56].

Méthode d'autocorrélation

En multipliant l'équation (1.12) pour $l \geq 0$, par $x(n-l)$, et en prenant l'espérance mathématique sur les deux côtés, on obtient:

$$E[x(n)x(n-l)] = -\sum_{k=1}^p a_k E[x(n-k)x(n-l)] + E[e(n)x(n-l)] \quad (1.21)$$

Le terme $E[e(n)x(n-l)]$ vaut σ_e^2 pour $l = 0$. Cependant pour $l > 0$ ce terme est nul du fait que les termes $X(n-l)$ et $e(n)$ sont indépendants et $E[e(n)] = 0$. Le premier terme de l'équation (1.21) est la fonction d'autocorrélation $r_x(l)$.

Donc :

$$r_x(l) = \begin{cases} -\sum_{k=1}^p a_k r_x(l-k), & l > 0 \\ -\sum_{k=1}^p a_k r_x(l-k) + \sigma_e^2, & l = 0 \end{cases} \quad (1.22)$$

Pour estimer les p coefficients, il nous faut p équations en plus des estimations appropriées de la fonction d'autocorrélation.

En mettant l'expression (1.22) pour $l > 0$ sous la forme matricielle:

$$\begin{bmatrix} \hat{r}_x(1) \\ \hat{r}_x(2) \\ \vdots \\ \hat{r}_x(p) \end{bmatrix} = - \begin{bmatrix} \hat{r}_x(0) & \hat{r}_x(-1) & \cdots & \hat{r}_x(-p+1) \\ \hat{r}_x(1) & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \hat{r}_x(p-1) & \cdots & \cdots & \hat{r}_x(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} \quad (1.23)$$

$$\hat{\mathbf{R}}_x \mathbf{a} = -\hat{\mathbf{r}}_x \quad (1.24)$$

on obtient les équations de Yule-Walker. Les coefficients a_k sont estimés par :

$$\hat{\mathbf{a}} = -\hat{\mathbf{R}}_x^{-1} \hat{\mathbf{r}}_x \quad (1.25)$$

et la variance du bruit :

$$\hat{\sigma}_e^2 = \hat{r}_x(0) + \sum_{k=1}^p a_k \hat{r}_x^*(k) \quad (1.26)$$

où:

$$\hat{r}_x(k) = \begin{cases} \frac{1}{N} \sum_{n=0}^{N-1-k} x^*(n)x(n+k), & \text{pour } k = 0, 1, \dots, p \\ r_x^*(-k) & \text{pour } k = -(p-1), -(p-2), \dots, -1 \end{cases} \quad (1.27)$$

Algorithme de Levinson

Pour résoudre les équations de Yule-Walker, on peut utiliser n'importe quelle méthode de résolution d'un système d'équations linéaires comme la méthode d'élimination de Gauss, qui nécessite $O(p^3)$ opérations. Puisque les équations de Yule-Walker ont une structure spéciale, elles peuvent être résolues en $O(p^2)$ par l'algorithme de Levinson.

Cet algorithme calcule les coefficients de prédiction au rang m à partir de ceux obtenus au rang $m-1$, c'est-à-dire les ensembles de paramètres $\{a_1(1), \delta_1\}$, $\{a_2(1), a_2(2), \delta_2\}$, ..., $\{a_p(1), a_p(2), \dots, a_p(p), \delta_p\}$. L'ensemble final d'ordre p est la solution cherchée pour les équations de Yule-Walker. Si $x(n)$ est un processus $AR(p)$, alors, $a_i = a_p(i)$, $i = 1, \dots, p$ et $\sigma_n^2 = \delta_p$.

Mettant l'équation (1.22) au rang m sous la forme matricielle :

$$\begin{bmatrix} r_x(0) & r_x^*(1) & \cdots & r_x^*(m-1) \\ r_x(1) & r_x(0) & \ddots & \vdots \\ \vdots & \vdots & \ddots & r_x^*(1) \\ r_x(m-1) & \cdots & \cdots & r_x(0) \end{bmatrix} \cdot \begin{bmatrix} a_{m-1}(0) \\ a_{m-1}(1) \\ \vdots \\ a_{m-1}(m-1) \end{bmatrix} = \begin{bmatrix} \delta_{m-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (1.28)$$

ou dans une forme plus compacte :

$$\mathbf{R}_{m-1} \mathbf{a}_{m-1}^F = \begin{bmatrix} \delta_{m-1} \\ \mathbf{0}_{m-1 \times 1} \end{bmatrix}, \quad \mathbf{R}_{m-1} \mathbf{a}_{m-1}^{B*} = \begin{bmatrix} \mathbf{0}_{m-1 \times 1} \\ \delta_{m-1}^* \end{bmatrix} \quad (1.29)$$

avec:

$$\mathbf{a}_{m-1}^F = [1 \quad a_{m-1}(1) \quad \cdots \quad a_{m-1}(m-1)]^T, \quad \mathbf{a}_{m-1}^{B*} = [a_{m-1}(m-1) \quad \cdots \quad a_{m-1}(1) \quad 1]^T \quad (1.30)$$

En passant au rang m , l'équation (1.28) devient:

$$\begin{bmatrix} r_x(0) & r_x^*(1) & \cdots & r_x^*(m-1) & r_x^*(m) \\ r_x(1) & r_x(0) & \ddots & \vdots & r_x^*(m-1) \\ \vdots & \vdots & \ddots & r_x^*(1) & \vdots \\ r_x(m-1) & r_x(m-2) & \cdots & r_x(0) & r_x^*(1) \\ r_x(m) & r_x(m-1) & \cdots & r_x(1) & r_x(0) \end{bmatrix} \cdot \begin{bmatrix} a_m(0) \\ a_m(1) \\ a_m(2) \\ \vdots \\ a_m(m) \end{bmatrix} = \begin{bmatrix} \delta_m(0) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (1.31)$$

$$\mathbf{R}_m \mathbf{a}_m^F = \begin{bmatrix} \delta_m \\ \mathbf{0}_{m \times 1} \end{bmatrix} \quad (1.32)$$

$$\mathbf{R}_m = \begin{bmatrix} \mathbf{R}_{m-1} & \mathbf{r}_m^{B*} \\ \mathbf{r}_m^{BT} & r_x(0) \end{bmatrix} = \begin{bmatrix} r_x(0) & \mathbf{r}_m^{FH} \\ \mathbf{r}_m^F & \mathbf{R}_{m-1} \end{bmatrix} \quad (1.33)$$

avec:

$$\mathbf{r}_m^B = [r_x(m) \quad r_x(m-1) \quad \cdots \quad r_x(1)]^T, \quad \mathbf{r}_m^F = [r_x(1) \quad r_x(2) \quad \cdots \quad r_x(m)]^T \quad (1.34)$$

En utilisant l'expression (1.32)

$$\mathbf{R}_m \cdot \begin{bmatrix} \mathbf{a}_{m-1}^F \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{m-1} & \mathbf{r}_m^{B*} \\ \mathbf{r}_m^{BT} & r_x(0) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{a}_{m-1}^F \\ 0 \end{bmatrix} = \begin{bmatrix} \delta_{m-1} \\ \mathbf{0}_{m-1 \times 1} \\ \mathbf{r}_m^{BT} \mathbf{a}_{m-1}^F \end{bmatrix} \quad (1.35)$$

$$\mathbf{R}_m \cdot \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix} = \begin{bmatrix} r_x(0) & \mathbf{r}_m^{FH} \\ \mathbf{r}_m^F & \mathbf{R}_{m-1} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_m^{FH} \mathbf{a}_{m-1}^{B*} \\ \mathbf{0}_{m-1 \times 1} \\ \delta_{m-1}^* \end{bmatrix} \quad (1.36)$$

Par combinaison lineaire :

$$\mathbf{R}_m \left[\begin{bmatrix} \mathbf{a}_{m-1}^F \\ 0 \end{bmatrix} + k_m \cdot \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix} \right] = \begin{bmatrix} \delta_{m-1} + k_m \mathbf{r}_m^{FH} \mathbf{a}_{m-1}^{B*} \\ \mathbf{0}_{m-1 \times 1} \\ \mathbf{r}_m^{BT} \mathbf{a}_{m-1}^F + k_m \delta_{m-1}^* \end{bmatrix} \quad (1.37)$$

En comparant les expressions (1.32) et (1.37) on deduit:

$$\mathbf{r}_m^{BT} \mathbf{a}_{m-1}^F + k_m \delta_{m-1}^* = 0 \quad (1.38)$$

d'où :

$$k_m = -\frac{\mathbf{r}_m^{BT} \mathbf{a}_{m-1}^F}{\delta_{m-1}^*}. \quad (1.39)$$

$$\text{et: } \delta_m = \delta_{m-1} + k_m \mathbf{r}_m^{FH} \mathbf{a}_{m-1}^{B*} \quad (1.40)$$

$$\delta_m = \delta_{m-1} (1 - |k_m|^2) \quad (1.41)$$

$$\mathbf{a}_m^F = \begin{bmatrix} \mathbf{a}_{m-1}^F \\ 0 \end{bmatrix} + k_m \cdot \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix} \quad (1.42)$$

Les coefficients k_m sont connus par les coefficients de réflexion.

L'algorithme de Levinson s'écrit alors:

Valeurs initiales : $\delta_0 = r_x(0)$, $a_0(0) = 1$

pour $m = 1, 2, \dots, p$, répéter:

$$k_m = -\frac{r_x(m)a_{m-1}(0) + r_x(m-1)a_{m-1}(1) + \dots + r_x(1)a_{m-1}(m-1)}{\delta_{m-1}}$$

$$a_m(0) = 1, a_m(m) = k_m$$

$$a_m(i) = a_{m-1}(i) + k_m a_{m-1}^*(m-i) \quad i = 1, \dots, m-1$$

$$\delta_m = (1 - |k_m|^2) \delta_{m-1}$$

avec : $r_x(k) = \frac{1}{N} \sum_{i=1}^{N-k} x(i)x(i+k)$, N est le nombre d'échantillons du signal.

Méthode de Burg

L'idée de l'algorithme de Burg repose sur l'estimation des coefficients de réflexion à partir des échantillons du signal en minimisant deux erreurs de prédiction, l'erreur directe et l'erreur rétrograde, données par:

$$\varepsilon_m^F(n) = x(n) + a_m(1)x(n-1) + \dots + a_m(m)x(n-m) = \mathbf{X}^T(n) \mathbf{a}_m^F \quad (1.43)$$

$$\varepsilon_m^B(n) = a_m(m)x^*(n) + a_m(m-1)x^*(n-1) + \dots + x^*(n-m) = \mathbf{X}^H(n) \mathbf{a}_m^B \quad (1.44)$$

En introduisant l'équation (1.42) dans (1.43) et (1.44) on obtient :

$$\varepsilon_m^F(n) = \mathbf{X}^T(n) \left(\begin{bmatrix} \mathbf{a}_{m-1}^F \\ 0 \end{bmatrix} + k_m \cdot \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix} \right) = \varepsilon_{m-1}^F(n) + k_m \varepsilon_{m-1}^{B*}(n-1) \quad (1.45)$$

$$\varepsilon_m^B(n) = \mathbf{X}^H(n) \left(\begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^B \end{bmatrix} + k_m \cdot \begin{bmatrix} \mathbf{a}_{m-1}^{F*} \\ 0 \end{bmatrix} \right) = \varepsilon_{m-1}^B(n-1) + k_m \varepsilon_{m-1}^{F*}(n) \quad (1.46)$$

Pour trouver les coefficients de reflexion, Burg a proposer de minimiser la somme des deux erreurs :

$$\varepsilon_m(n) = E \left\{ \left| \varepsilon_m^F(n) \right|^2 + \left| \varepsilon_m^B(n) \right|^2 \right\} \quad (1.47)$$

d'où :

$$k_m = \frac{2E \left\{ \varepsilon_{m-1}^B(n-1) \varepsilon_{m-1}^F(n) \right\}}{E \left\{ \left| \varepsilon_{m-1}^F(n) \right|^2 + \left| \varepsilon_{m-1}^B(n-1) \right|^2 \right\}} \quad (1.48)$$

Dans la pratique on remplace les epérances par des moyennes temporelles :

$$E \left\{ \varepsilon_{m-1}^B(n-1) \varepsilon_{m-1}^F(n) \right\} \approx \frac{1}{N-m+1} \sum_{n=m}^N \varepsilon_{m-1}^B(n-1) \varepsilon_{m-1}^F(n) \quad (1.49)$$

$$E \left\{ \left| \varepsilon_{m-1}^F(n) \right|^2 + \left| \varepsilon_{m-1}^B(n-1) \right|^2 \right\} \approx \frac{1}{N-m+1} \sum_{n=m}^N \left| \varepsilon_{m-1}^F(n) \right|^2 + \left| \varepsilon_{m-1}^B(n-1) \right|^2 \quad (1.50)$$

Méthode des covariances modifiées

En reprenant l'expression (1.43) avec n allant de $p+1$ à $N-p$, on obtient:

$$\begin{bmatrix} x(p+1) \\ \vdots \\ x(N-p) \end{bmatrix} = \begin{bmatrix} x(p) & \dots & x(1) \\ \vdots & \ddots & \vdots \\ x(N-p-1) & \dots & x(N-2p) \end{bmatrix} \cdot \begin{bmatrix} -a_1 \\ \vdots \\ -a_p \end{bmatrix} + \begin{bmatrix} \varepsilon_F(p+1) \\ \vdots \\ \varepsilon_F(p+1) \end{bmatrix} \quad (1.51)$$

ou sous la forme matricielle :

$$X = -Da + \varepsilon_F \quad (1.52)$$

La solution recherchée en minimisant (1.52) par rapport à a est donnée par :

$$a = -(D^T D)^{-1} D^T X \quad (1.53)$$

De la même manière, à partir de l'expression (1.44), on forme le système d'équations:

$$\begin{bmatrix} x(p+1) \\ \vdots \\ x(N-p) \end{bmatrix} = \begin{bmatrix} x(p+2) & \dots & x(2p+1) \\ \vdots & \ddots & \vdots \\ x(N-p+1) & \dots & x(N) \end{bmatrix} \cdot \begin{bmatrix} -a_1 \\ \vdots \\ -a_p \end{bmatrix} + \begin{bmatrix} \varepsilon_B(2p+1) \\ \vdots \\ \varepsilon_B(N) \end{bmatrix} \quad (1.54)$$

$$X = -Ba + \varepsilon_B \quad (1.55)$$

En sommant maintenant les deux expressions (1.52),(1.55), on obtient:

$$X = -\frac{1}{2}(D+B)a + \frac{1}{2}(\varepsilon_F + \varepsilon_B) \quad (1.56)$$

D'où la solution recherchée :

$$a = -(H^T H)^{-1} H^T X \quad (1.57)$$

1.3.1.2. Densité spectrale estimée basée sur le modèle à moyenne mobile

Pour un modèle $MA(q)$, les valeurs de $r_x(k)$ pour $|k| > q$ sont nulles. D'où l'expression (1.20) peut être réécrite sous la forme :

$$P_{MA}(f) = \sum_{k=-q}^q r_x(k) e^{-j2\pi f k} \quad (1.58)$$

Donc pour trouver $P_{MA}(f)$, il suffit d'estimer les coefficients $r_x(k)$ pour $|k| \leq q$ et les remplacer dans l'expression (1.58). Une autre approche pour déterminer $P_{MA}(f)$ consiste à estimer les coefficients MA et la variance σ_e^2 , puis les remplacer dans l'expression (1.20). Durbin a proposé une méthode approximative qui consiste à approximer le modèle $MA(q)$ par un modèle AR d'ordres très supérieur devant q [1]. Tout d'abord les données sont modélisées par un modèle AR d'ordre $L \gg q$. Les coefficients $\hat{a}_i, i = 1, 2, \dots, L$ et $\hat{\sigma}_e^2$ sont estimés par l'algorithme de Levinson. Dans la deuxième étape, on utilise l'ensemble des coefficients $\{1, a_i\}$ comme des données pour estimer les coefficients $\hat{b}_i, i = 1, 2, \dots, q$ par la méthode d'autocorrélation.

1.3.1.3. Densité spectrale estimée basée sur le modèle autorégressif à moyenne mobile

Reprenant l'équation (1.22) pour $M \geq l \geq q + 1$:

$$\begin{bmatrix} r_x(q+1) \\ r_x(q+2) \\ \vdots \\ r_x(M) \end{bmatrix} = - \begin{bmatrix} r_x(q) & r_x(q-1) & \cdots & r_x(q-p+1) \\ r_x(1) & \vdots & & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ r_x(M-1) & \cdots & \cdots & r_x(M-p) \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} \quad (1.59)$$

On a $M - q$ équations avec p inconnus. Supposons que $M - q > p$, donc on aura plus d'équations que d'inconnus. Quand la fonction d'autocorrélation théorique est remplacée par son estimation, les équations ne sont plus satisfaites ; en prenant compte aux erreurs d'estimation de la FAC, le système d'équation devient :

$$\hat{\mathbf{r}}_x = -\hat{\mathbf{R}}\mathbf{a} + \mathbf{e} \quad (1.60)$$

Le vecteur a des coefficients AR peut être obtenu par l'estimateur des moindres carrés :

$$\hat{\mathbf{a}} = -(\hat{\mathbf{R}}^H \hat{\mathbf{R}})^{-1} \hat{\mathbf{R}}^H \hat{\mathbf{r}}_x \quad (1.61)$$

Pour les coefficients \hat{b}_k et $\hat{\sigma}_e^2$, on peut utiliser la procédure de Durbin pour le processus MA .

1.3.2. Modèle à décomposition harmonique

Supposons que les données consistent en m sinusoides complexes noyées dans un bruit blanc gaussien complexe $e(n)$:

$$x(n) = \sum_{i=1}^m A_i e^{j2\pi f_i n} + e(n) \quad (1.62)$$

où f_i , A_i sont la fréquence et l'amplitude complexe de la sinusoïde i . L'amplitude complexe A_i peut être mis sous la forme :

$$A_i = |A_i| e^{j\phi_i} \quad (1.63)$$

où ϕ_i est une phase aléatoire.

La densité spectrale de puissance du processus est une somme d'un spectre continu du bruit $P_e(f)$ et un ensemble d'impulsions d'une surface $|A_i|^2$ à la fréquence f_i .

$$P(f) = \sum_{i=1}^m |A_i|^2 \delta(f - f_i) + P_e(f) \quad (1.64)$$

Pisarenko [57] a montré que les fréquences des sinusoides peuvent être obtenues à partir du vecteur propre correspondant à la plus petite valeur propre de la matrice d'autocorrélation.

Sous l'hypothèse que le bruit soit centré et de variance σ_e^2 , l'autocorrélation de la séquence $\{x(n)\}_{n=0}^{N-1}$ peut être écrite sous la forme :

$$r_x(k) = \sum_{i=1}^m |A_i|^2 e^{j2\pi f_i k} + \sigma_e^2 \delta(k) \quad (1.65)$$

La matrice d'autocorrélation peut être également écrite sous la forme :

$$\mathbf{R} = \sum_{i=1}^m |A_i|^2 \mathbf{e}_i \mathbf{e}_i^H + \sigma_e^2 \mathbf{I} \quad (1.66)$$

avec:
$$\mathbf{e}_i = [1 e^{j2\pi f_i k} e^{j2\pi f_i k} \dots e^{j2\pi(N-1)f_i k}]^T \quad (1.67)$$

et \mathbf{I} la matrice identité

Il est clair que la matrice \mathbf{R} est la somme de la matrice d'autocorrélation du signal et celle du bruit.

$$\mathbf{R} = \mathbf{R}_s + \sigma_e^2 \mathbf{I} \quad (1.68)$$

où:
$$\mathbf{R}_s = \mathbf{E} \mathbf{P} \mathbf{E}^H \quad (1.69)$$

$$\text{avec: } \mathbf{E} = [\mathbf{e}_1 \ \mathbf{e}_2 \ \dots \ \mathbf{e}_m] \quad (1.70)$$

$$\mathbf{P} = \text{diag} \{ |A_1|^2, |A_2|^2, \dots, |A_m|^2 \} \quad (1.71)$$

Une autre représentation importante de la matrice \mathbf{R} par ses valeurs et vecteurs propres :

$$\mathbf{R} = \sum_{i=1}^m (\lambda_i + \sigma_e^2) \mathbf{v}_i \mathbf{v}_i^H + \sum_{i=1}^m \sigma_e^2 \mathbf{v}_i \mathbf{v}_i^H \quad (1.72)$$

où $\lambda_i, i=1,2,\dots,m$ sont les valeurs propres non nulles de la matrice \mathbf{R}_s

En arrangeant les valeurs propres dans un ordre décroissant $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_M$ et soient \mathbf{v}_i les vecteurs propres correspondant à λ_i . L'espace engendré par les vecteurs propres $\mathbf{v}_i, i=1,2,\dots,m$ est dit 'espace signal', et l'espace engendré par $\mathbf{v}_i, i=m+1, m+2, \dots, M$ est dit espace bruit. Puisque l'ensemble des vecteurs propres sont orthonormaux, les deux espaces sont orthogonaux.

En supposant que la matrice \mathbf{R} est de dimension $(m+1) \times (m+1)$, Pisarenko a observé que la variance du bruit correspond à la plus petite valeur propre de \mathbf{R} , en plus les fréquences sont estimées via l'orthogonalité des deux espaces :

$$\mathbf{e}_i^H \mathbf{v}_{m+1} = 0, \quad i=1,2,\dots,m \quad (1.73)$$

Ceci conduit à la formulation du pseudo densité spectral de puissance:

$$\hat{P}(f) = \frac{1}{|\mathbf{e}^H(f) \mathbf{v}_{m+1}|^2} \quad (1.74)$$

1.4. Méthodes non-paramétriques basé sur le filtrage adaptatif

Comme indiqué dans l'introduction de ce chapitre, le problème de l'estimation spectrale consiste à estimer la façon dont la puissance est répartie dans la bande fréquentielle à partir d'un nombre fini d'observations [58]. Pour un signal composé de sinusoïdes noyées dans un bruit stationnaire :

$$x(n) = \sum_{k=1}^K \alpha_k e^{j2\pi f_k n} + v(n) \quad (1.77)$$

où α_k, f_k , l'amplitude et la fréquence de la $k^{\text{ème}}$ sinusoïde à estimer; $v(n)$ est le bruit d'observation complexe, il est plus naturel de définir le spectre d'amplitudes complexes :

$$\alpha_k(f) = \begin{cases} \alpha_k & f = f_k \\ 0 & f \neq f_k \end{cases} \quad (1.78)$$

Pour estimer la densité spectrale de la suite $\{x(n)\}$, on suppose que le signal $x(n)$ à la fréquence f est composé d'une sinusoïde d'amplitude $\alpha(f)$ et d'un terme résiduel $v_f(n)$ [5].

$$x(n) = \alpha(f)e^{j2\pi fn} + v_f(n) \quad (1.79)$$

Dans le contexte de l'équation (1.79), $v_f(n)$ contient tous les sinusoïdes à l'exception de celle présente dans f et le bruit d'observation $v(n)$.

$$v_f(n) = \sum_{k=1, f_k \neq f}^K \alpha_k e^{j2\pi f_k n} + v(n) \quad (1.80)$$

Par conséquent le problème de l'estimation du spectre d'amplitude de $x(n)$ consiste à l'estimation de $\alpha(f)$ à partir des observations $x(n)$

Par la subdivision du vecteur d'observations en L segments de longueur M comme montré dans la figure (1.2). Chaque segment est défini par:

$$\mathbf{y}_l = [x(l) \quad x(l+1) \quad \dots \quad x(l+M-1)]^T \quad (1.81)$$

$$l = 0, 1, \dots, L-1$$

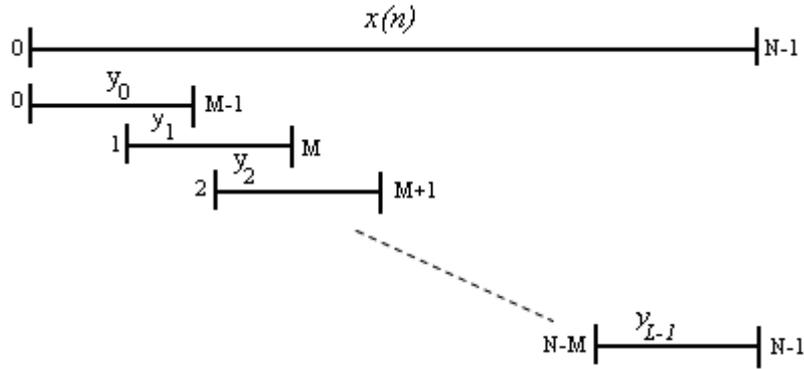


Figure 1.8. Segmentation du vecteur d'observations.

Cette dernière définition nous permet d'arranger les segments \mathbf{y}_l sous la forme suivante :

$$\mathbf{Y} = [\mathbf{y}_0, \dots, \mathbf{y}_{L-1}] = \begin{bmatrix} x(0) & x(1) & \dots & x(L-1) \\ x(1) & x(2) & \dots & x(L) \\ \vdots & \vdots & \ddots & \vdots \\ x(M-1) & x(M) & \dots & x(N-1) \end{bmatrix} \quad (1.82)$$

De la même manière on définit :

$$\mathbf{V}(f) = [\mathbf{v}_0(f), \dots, \mathbf{v}_{L-1}(f)] = \begin{bmatrix} v_f(0) & v_f(1) & \dots & v_f(L-1) \\ v_f(1) & v_f(2) & \dots & v_f(L) \\ \vdots & \vdots & \ddots & \vdots \\ v_f(M-1) & v_f(M) & \dots & v_f(N-1) \end{bmatrix} \quad (1.83)$$

et on définit aussi le vecteur $\mathbf{a}_M(f)$ appelé le vecteur de Fourier d'ordre M comme :

$$\mathbf{a}_M(f) = [1 \quad e^{j2\pi f} \quad \dots \quad e^{j2\pi(M-1)f}]^T \quad (1.84)$$

Si on pose :

$$\mathbf{A}(f) = \mathbf{a}_M(f)\mathbf{a}_L^T(f) \quad (1.85)$$

L'équation (1.77) pour l'ensemble des échantillons disponibles peut être réécrite sous la forme :

$$\mathbf{Y} = \alpha(f)\mathbf{A}(f) + \mathbf{V}(f) \quad (1.86)$$

L'équation (1.86) exprime une régression linéaire entre la quantité à estimer $\alpha(f)$ et la matrice des données \mathbf{Y}

1.4.1. Méthode de Capon

La méthode d'analyse spectrale de Capon [9], est considérée comme une des techniques importantes parmi les méthodes dites à haute résolution, dans le sens qu'elles atteignent de meilleures performances pour l'estimation de la fréquence d'une sinusoïde par rapport aux techniques basées sur les méthodes classiques de Fourier [59]. La densité spectrale du signal sous analyse est estimée en utilisant un banc de filtres de façon à couvrir toute la bande analysée [60-63].

Soit $\mathbf{h}(f)$ un vecteur de longueur M qui contient les coefficients du filtre, l'argument f désigne la fréquence d'accord du filtre.

$$\mathbf{h}(f) = [h_0(f) \quad h_1(f) \quad \dots, \quad h_{M-1}(f)]^T \quad (1.87)$$

Si on applique ce filtre au vecteur de données disponibles, la sortie du filtre peut être écrite sous la forme :

$$z(l) = \sum_{m=0}^{M-1} x(l+m)h_m(f) = \mathbf{h}^H(f)\mathbf{y}_l, \quad l = 0, \dots, L-1 \quad (1.88)$$

Les coefficients du filtre sont choisis de façon à minimiser la puissance du signal en sortie du filtre avec la contrainte qu'une exponentielle, de fréquence f , à l'entrée du filtre passe sans distorsion [2,5] :

$$\sum_{m=0}^{M-1} h_m(f)e^{-j2\pi mf} = 1 \quad (1.89)$$

On cherche donc à conserver la puissance à la fréquence f , tout en minimisant les puissances aux autres fréquences. Si donc le signal contient une composante à cette fréquence, cette composante sera rehaussée par rapport au reste qu'on cherche à minimiser. En observant la puissance en sortie du filtre en fonction de f , on observe donc des pics aux fréquences

présentes dans le signal $x(n)$. Les coefficients de ce filtre sont déterminés par la minimisation de la puissance à la sortie du filtre. Cette puissance est donnée par :

$$E\{|z(l)|^2\} = E\{|\mathbf{h}^H(f)\mathbf{y}_l|^2\} = \mathbf{h}^H(f)E\{\mathbf{y}_l\mathbf{y}_l^H\}\mathbf{h}(f) = \mathbf{h}^H(f)\mathbf{R}\mathbf{h}(f) \quad (1.90)$$

\mathbf{R} est la matrice d'autocorrélation du segment \mathbf{y}_l .

Le critère pour concevoir le filtre passe-bande est de minimiser $E\{|z(l)|^2\}$ avec la contrainte :

$$\mathbf{h}^H(f)\mathbf{a}_M(f) = 1 \quad (1.91)$$

$$\text{où : } \mathbf{a}(f) = [1 \quad e^{j2\pi f} \quad e^{j2\pi f^2} \quad \dots \quad e^{j\pi f(M-1)}]. \quad (1.92)$$

Pour résoudre ce problème, on utilise la méthode de Lagrange :

$$J = \mathbf{h}^H(f)\mathbf{R}\mathbf{h}(f) + \lambda (1 - \mathbf{h}^H(f)\mathbf{a}_M(f)) \quad (1.93)$$

Le problème revient à trouver le filtre $\mathbf{h}(f)$ qui minimise la fonction J . En dérivant l'expression (1.93) par rapport à $\mathbf{h}(f)$, on trouve :

$$\frac{\partial J}{\partial \mathbf{h}(f)} = 2\mathbf{R}\mathbf{h}(f) - \lambda\mathbf{a}_M(f) \quad (1.94)$$

$\mathbf{h}(f)$ est obtenu par la mise à zéro de l'expression (1.94)

$$\frac{\partial J}{\partial \mathbf{h}(f)} = 0 \Rightarrow \mathbf{h}(f) = \frac{\lambda}{2}\mathbf{R}^{-1}\mathbf{a}_M(f) \quad (1.95)$$

En remplaçant l'équation (1.95) dans la contrainte $\mathbf{h}^H(f)\mathbf{a}_M(f) = 1$, on obtient :

$$\frac{\lambda}{2} = \frac{1}{\mathbf{a}_M^H(f)\mathbf{R}^{-1}\mathbf{a}_M(f)} \quad (1.96)$$

En remplaçant l'expression (1.96) dans 1.95), l'expression du filtre devient :

$$\mathbf{h}_{\text{Capon}}(f) = \frac{\mathbf{R}^{-1}\mathbf{a}(f)}{\mathbf{a}_M^H(f)\mathbf{R}^{-1}\mathbf{a}_M(f)} \quad (1.97)$$

La puissance en sortie du filtre est déduite de (1.97) et (1.90):

$$\hat{P}_{\text{Capon}}(f) = \frac{1}{\mathbf{a}_M^H(f)\mathbf{R}^{-1}\mathbf{a}_M(f)} \quad (1.98)$$

La densité spectrale estimée par la méthode de Capon $\hat{\alpha}(f)$ est donnée par :

$$\hat{\alpha}_{\text{Capon}}(f) = \mathbf{h}^H(f)g(f) = \frac{\mathbf{a}_M^H(f)\mathbf{R}^{-1}\mathbf{g}(f)}{\mathbf{a}_M^H(f)\mathbf{R}^{-1}\mathbf{a}_M(f)} \quad (1.99)$$

$\mathbf{g}(f)$ est la transformée de Fourier de \mathbf{y}_l à la fréquence f . Elle est donnée par :

$$\mathbf{g}(f) = \frac{1}{L} \sum_{l=0}^{L-1} \mathbf{y}_l e^{-j2\pi fl} \quad (1.100)$$

Puisque le calcul de la matrice de corrélation \mathbf{R} n'est pas possible suite au nombre fini d'échantillons, cette dernière peut être approximée par [58,64] :

$$\hat{\mathbf{R}} = \frac{1}{L} \sum_{l=0}^{L-1} \mathbf{y}_l \mathbf{y}_l^H \quad (1.101)$$

La méthode de Capon ne nécessite aucune hypothèse spéciale sur le signal. De plus, elle permet en général d'obtenir une meilleure résolution que le périodogramme, notamment pour de faibles valeurs de N .

1.4.2. Méthode APES

Le principe de la méthode APES est similaire à celui de Capon, pour une fréquence f , on cherche à concevoir un filtre $\mathbf{h}(f)$ qui minimise l'erreur entre la sortie du filtre et la composante du signal à la fréquence désirée [9,10].

En partant de l'équation (1.79), on considère la minimisation de la fonction objective :

$$\sum_{l=0}^{L-1} |\mathbf{h}^H(f) \mathbf{y}_l - \alpha(f) e^{j2\pi fl}|^2 \quad (1.102)$$

sous la même contrainte vue au paragraphe précédent. La fonction objective (1.102) peut être mise sous la forme:

$$\begin{aligned} & \sum_{l=0}^{L-1} |\mathbf{h}^H(f) \mathbf{y}_l - \alpha(f) e^{j2\pi fl}|^2 \\ &= \mathbf{h}^H(f) \hat{\mathbf{R}} \mathbf{h}(f) - \alpha^*(f) \mathbf{h}^H(f) \mathbf{g}(f) - \alpha(f) \mathbf{g}^H(f) \mathbf{h}(f) + |\alpha(f)|^2 \\ &= |\alpha(f) - \mathbf{h}^H(f) \mathbf{g}(f)|^2 + \mathbf{h}^H(f) \hat{\mathbf{R}} \mathbf{h}(f) - |\mathbf{h}^H(f) \mathbf{g}(f)|^2 \end{aligned} \quad (1.103)$$

où $(\cdot)^*$ désigne le complexe conjugué.

La minimisation de (1.103) par rapport à $\alpha(f)$ donne :

$$\alpha(f) = \mathbf{h}^H(f) \mathbf{g}(f) \quad (1.104)$$

Par l'insertion de (1.104) dans (1.103) le problème de minimisation est reformulé comme :

$$\begin{aligned} & \min_{\mathbf{h}(f)} \mathbf{h}^H(f) \hat{\mathbf{Q}} \mathbf{h}(f) \\ & \text{sous la contrainte : } \mathbf{h}^H(f) \mathbf{a}_M(f) = 1 \end{aligned} \quad (1.105)$$

$$\text{avec: } \hat{\mathbf{Q}} = \hat{\mathbf{R}} - \mathbf{g}(f) \mathbf{g}^H(f) \quad (1.106)$$

La solution de ce problème est similaire à celle de Capon, pour déterminer le filtre, la puissance à la sortie du filtre et la densité spectrale du signal $x(n)$ il suffit de remplacer $\hat{\mathbf{R}}$ par $\hat{\mathbf{Q}}$ dans les expressions (1.97), (1.98) et (1.99).

L'expression du filtre devient :

$$\mathbf{h}_{APES}(f) = \frac{\hat{\mathbf{Q}}^{-1} \mathbf{a}_M(f)}{\mathbf{a}_M^H(f) \hat{\mathbf{Q}}^{-1} \mathbf{a}_M(f)} \quad (1.107)$$

La puissance à la sortie du filtre est donnée par :

$$\hat{P}_{APES}(f) = \frac{1}{\mathbf{a}_M^H(f) \hat{\mathbf{Q}}^{-1} \mathbf{a}_M(f)} \quad (1.108)$$

La densité spectrale estimée par la méthode d'APES est donnée par :

$$\hat{\alpha}_{APES}(f) = \frac{\mathbf{a}_M^H \hat{\mathbf{Q}}^{-1} \mathbf{g}(f)}{\mathbf{a}_M^H(f) \hat{\mathbf{Q}}^{-1} \mathbf{a}_M(f)} \quad (1.109)$$

Pour éviter l'inversion de la matrice $\hat{\mathbf{Q}}$ pour chaque fréquence, on utilise [59] :

$$\hat{\mathbf{Q}}^{-1}(f) = \hat{\mathbf{R}}^{-1} + \frac{\hat{\mathbf{R}}^{-1} \mathbf{g}(f) \mathbf{g}^H(f) \hat{\mathbf{R}}^{-1}}{1 - \mathbf{g}^H(f) \hat{\mathbf{R}}^{-1} \mathbf{g}(f)} \quad (1.110)$$

1.5. Conclusion

Le problème de l'estimation spectrale revient à définir comment la puissance d'un signal est répartie dans sa bande fréquentielle. Dès la naissance de l'analyse de Fourier jusqu'à nos jours, une grande diversité de méthodes ont été proposées. Ces méthodes peuvent être regroupées en trois familles, les méthodes non paramétriques, les méthodes paramétriques et les méthodes non paramétriques basées sur le filtrage adaptatif. Bien que les méthodes non paramétriques soient plus robustes, elles souffrent de la faible résolution fréquentielle conditionnée par la limite de Fourier.

Pour les méthodes paramétriques, elles se divisent elles aussi en deux catégories selon le modèle; les méthodes basées sur le modèle de la fonction de transfert, et les méthodes basées sur le modèle des sinusoides. Quand le modèle est parfaitement connu, les méthodes paramétriques sont très efficaces. De même elles offrent une résolution fréquentielle beaucoup plus meilleure que celle des méthodes non paramétriques. L'inconvénient de cette classe de méthodes réside dans la connaissance du modèle.

Dans la troisième classe des méthodes, le signal est supposé une somme de sinusoides noyé dans un bruit stationnaire et le problème de l'estimation spectrale consiste à estimer la densité spectrale d'amplitudes de ce signal. Ces méthodes consistent à concevoir un filtre à réponse impulsionnelle fini, centré à la fréquence désirée de telle sorte qu'il ne laisse passer que la fréquence désirée sans distorsion. Cette classe est un compromis entre les deux classes précédentes ; elle n'exige aucune information sur le signal et offre une résolution fréquentielle située entre celles des deux autres classes de méthodes.

Chapitre 2

Estimation du spectre de raies par le réseau de neurones de Hopfield

2.1. Introduction

La manière la plus naturelle d'appliquer les réseaux de neurones récurrents dans l'estimation spectrale consiste à rendre le problème de l'estimation spectrale à un problème d'optimisation dont le réseau de neurones de Hopfield est très approprié. Donc l'idée derrière l'utilisation du réseau de Hopfield réside dans la reformulation du problème de l'estimation spectrale sous la forme d'une fonction de coût à minimiser équivalente à l'énergie du réseau de Hopfield [68-69]. Dans le présent chapitre, on s'intéresse à l'estimation de l'amplitude des raies spectrales d'un mélange de sinusoides réelles ou complexes noyées dans un bruit blanc gaussien. Deux cas sont distingués et seront étudiés, les fréquences des raies sont connues ou bien non connues. Dans le premier cas, le réseau est construit de deux façons, l'une consiste à séparer les données en partie réelle et partie imaginaire, et de concevoir un réseau de Hopfield dont l'état contient la partie réelle et la partie imaginaire des amplitudes estimées. L'autre consiste à concevoir deux réseaux à valeurs réelles, l'un pour estimer la partie réelle, l'autre pour estimer la partie imaginaire du vecteur d'amplitudes recherchées. Dans le deuxième cas on s'intéresse à l'estimation simultanée de l'amplitude, la phase et la fréquence utilisant un réseau de Hopfield modifié.

2.2. Réseaux de neurones

2.2.1. Définitions des réseaux de neurones

Inspirés du fonctionnement du cerveau humain, les réseaux de neurones artificiels occupent aujourd'hui une place prépondérante dans plusieurs domaines des sciences de l'ingénieur. Le neurone formel souvent nommé perceptron mis au point par McCulloch et Pitts est l'élément constitutif de base d'un réseau de neurones. C'est un processeur

élémentaire qui reçoit un nombre d'entrées dont à chacune est associé un poids représentant la force de la connexion et délivre une sortie [23.] Le fonctionnement d'un neurone se fait en deux étapes ; il calcule son état d'excitation interne en fonction des entrées pondérées, puis il émet un résultat en sortie à travers une fonction appelée fonction d'activation.

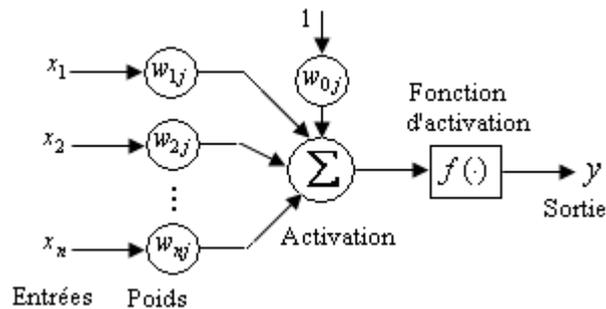


Figure 2.1. Le modèle de neurone formel.

La figure (2.1) donne le schéma de principe d'un perceptron, on distingue un vecteur d'entrée. $\mathbf{x} = [x_1, x_2, \dots, x_n]$, un vecteur de poids de connexions $\mathbf{w} = [w_{1j}, w_{2j}, \dots, w_{nj}]$ tel que w_{ij} est le poids entre l'entrée i et le neurone j et un sommateur qui calcule l'état d'excitation du neurone donnée par :

$$a_j = \sum_{i=1}^N w_{ij} x_i \quad (2.1)$$

Le neurone est dit excité si $a_j > 0$, et non excité dans le cas contraire. Généralement on ajoute au vecteur d'entrée une entrée dont la valeur est fixée à 1, dans ce cas l'activation du neurone devient:

$$a_j = \sum_{i=1}^N w_{ij} x_i + w_{0j} x_0 \quad (2.2)$$

Cela revient à fixer un seuil $-w_{0j}$ au-delà duquel le neurone sera considéré comme excité. En sortie, le perceptron délivre une valeur $y = f(a_j)$. La fonction f est appelée fonction d'activation.

2.2.2. Types de réseaux de neurones

Deux grandes familles de réseaux de neurones peuvent être distinguées: les réseaux non-récurrents et les réseaux récurrents. Dans les réseaux non récurrents appelés aussi non bouclés ou réseaux statiques, les neurones sont organisés en couches successives et dont l'information se propage en avant de couche en couche. Les sorties des neurones d'une couche servent d'entrées aux neurones de la couche suivante. Au sein d'une même couche, les

neurones ne sont pas connectés entre elles. La première couche est appelée la couche d'entrée, la dernière, la couche de sortie et les couches intermédiaires sont appelées les couches cachées. Les réseaux de neurones bouclés appelés aussi réseaux récurrents ou réseaux dynamiques, sont des réseaux dans lesquels il y a un retour de l'information en arrière. Cela suppose que la sortie d'un neurone peut être fonction d'elle-même. Or cela n'est possible que si la notion de temps est prise en compte [65]. La figure (2.2) montre les deux structures des réseaux de neurones.

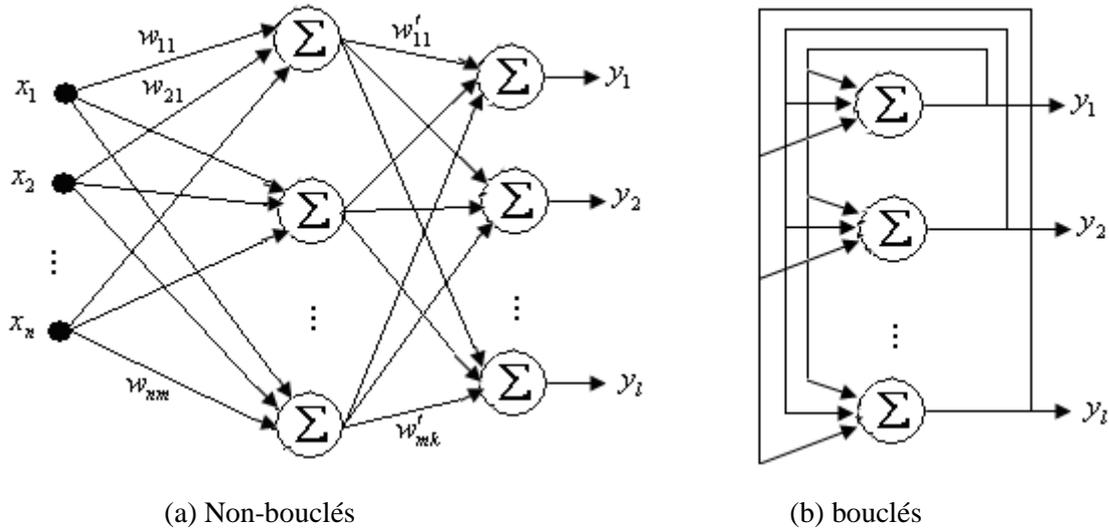


Figure 2.2. Structure des réseaux de neurones.

2.2.3. Réseaux de neurones de Hopfield pour l'optimisation

Le réseau de Hopfield [66] est un réseau dynamique constitué d'une seule couche de neurones complètement connectés. La dynamique du réseau de Hopfield à temps continu est régit par l'équation différentielle [65] :

$$\tau_i \dot{y}_i(t) = -y_i(t) + \sum w_{ij} u_j(t) + v_i \quad (2.5)$$

$$u_i(t) = f(y_i(t))$$

où $y_i(t)$ est l'état interne du neurone i , $u_i(t)$ est la sortie du neurone i , w_{ij} est le poids de connexion entre le neurone i et le neurone j et $v_i(t)$ est l'entrée du neurone i . La figure (2.3) montre la structure du réseau de Hopfield selon l'équation (2.5). En partant d'un état initial aléatoire, le réseau converge vers un état stable correspond au minima d'une fonction énergie définie par Hopfield comme:

$$E(t) = -\frac{1}{2} \mathbf{u}(t)^T \mathbf{W} \mathbf{u}(t) - \mathbf{u}(t)^T \mathbf{v} \quad (2.6)$$

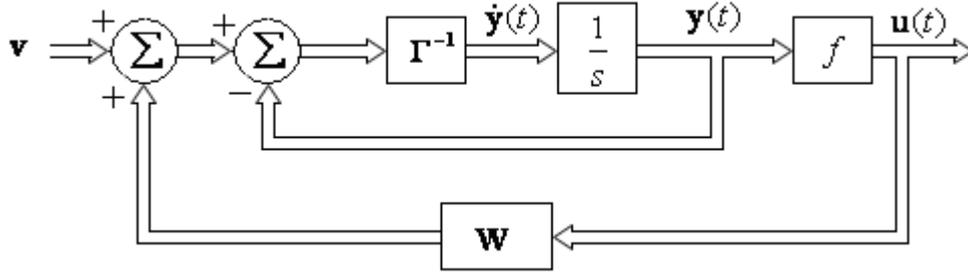


Figure 2.3. Structure du réseau de Hopfield.

Les réseaux de neurones sont caractérisés par le principe du parallélisme intrinsèque. Ils permettent d'exécuter un grand nombre de calculs dans un temps limité, d'où ils offrent une alternative intéressante pour résoudre les problèmes complexes d'optimisation, dont le réseau de neurones de Hopfield est très approprié.

Les réseaux de neurones récurrents sont largement exploités dans le domaine de l'estimation spectrale. Dans [40], S. K. Park a proposé d'introduire le réseau de Hopfield dans l'estimation de la densité spectrale de puissance utilisant la modélisation autorégressive. Pour estimer les paramètres d'un signal sinusoïdal à savoir la fréquence, la phase et l'amplitude, L. Han [38] a proposé une approche basée sur le réseau de Hopfield. Puis une extension de cette approche pour les signaux composés d'exponentielles complexes a été faite dans [39].

2.3. Formulation du problème de l'estimation du spectre de raies

On considère N observations d'un signal $x(n)$ composé de p sinusoïdes complexes corrompues par un bruit additif $v(n)$

$$x(n) = \sum_{m=1}^p a_m e^{j2\pi f_m n} + v(n), \quad n = 0, 1, \dots, N-1 \quad (2.1)$$

a_m , f_m sont l'amplitude et la fréquence de la sinusoïde m , $v(n)$ est un bruit complexe, avec une moyenne nulle et une variance σ^2 .

En réécrivant l'équation (2.1) sous la forme matricielle :

$$\begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix} = \begin{bmatrix} 1 & \dots & 1 \\ e^{j2\pi f_1} & \dots & e^{j2\pi f_p} \\ \vdots & \vdots & \vdots \\ e^{j(N-1)2\pi f_1} & \dots & e^{j(N-1)2\pi f_p} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} + \begin{bmatrix} v(0) \\ v(1) \\ \vdots \\ v(N-1) \end{bmatrix} \quad (2.2)$$

Si on pose :

$$\mathbf{x} = \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}, \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix}, \mathbf{v} = \begin{bmatrix} v(0) \\ v(1) \\ \vdots \\ v(N-1) \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 1 & \dots & 1 \\ e^{j2\pi f_1} & \dots & e^{j2\pi f_p} \\ \vdots & \vdots & \vdots \\ e^{j(N-1)2\pi f_1} & \dots & e^{j(N-1)2\pi f_p} \end{bmatrix}$$

on obtient l'expression suivante:

$$\mathbf{x} = \mathbf{A}\mathbf{a} + \mathbf{v} \quad (2.3)$$

Cette expression est un système d'équations à valeurs complexes dont la solution nous fournit les amplitudes des raies cherchées.

Le problème de l'estimation peut être reformulé d'une autre manière, comme le signal est complexe, $x(n)$ peut être réécrit sous la forme :

$$\begin{aligned} x(n) &= x_{\text{Re}}(n) + jx_{\text{Im}}(n) \\ &= \sum_{m=1}^p (a_m^{\text{Re}} + ja_m^{\text{Im}}) (\cos(n2\pi f_m) + j \sin(n2\pi f_m)) + v(n) \end{aligned} \quad (2.4)$$

d'où on extrait la partie réelle et la partie imaginaire du signal $x(n)$:

$$\begin{aligned} x_{\text{Re}}(n) &= \sum_{m=1}^p (a_m^{\text{Re}} \cos(n2\pi f_m) - a_m^{\text{Im}} \sin(n2\pi f_m)) + v^{\text{Re}}(n) \\ x_{\text{Im}}(n) &= \sum_{m=1}^p (a_m^{\text{Re}} \sin(n2\pi f_m) + a_m^{\text{Im}} \cos(n2\pi f_m)) + v^{\text{Im}}(n) \end{aligned} \quad (2.5)$$

Ces deux expressions nous permettent de montrer que pour $m = 1, \dots, p$, on aura:

$$\begin{aligned} x_{\text{Re}}(n) &= [\cos(n2\pi f_m) - \sin(n2\pi f_m)] \begin{bmatrix} a_m^{\text{Re}} \\ a_m^{\text{Im}} \end{bmatrix} + v^{\text{Re}}(n) \\ x_{\text{Im}}(n) &= [\sin(n2\pi f_m) \cos(n2\pi f_m)] \begin{bmatrix} a_m^{\text{Re}} \\ a_m^{\text{Im}} \end{bmatrix} + v^{\text{Im}}(n) \end{aligned} \quad (2.6)$$

On suppose maintenant que le vecteur d'observations, le vecteur du bruit et le vecteur des paramètres sont réorganisés de la forme :

$$\begin{aligned} \bar{\mathbf{x}} &= [x^{\text{Re}}(0), x^{\text{Re}}(1), \dots, x^{\text{Re}}(N-1), x^{\text{Im}}(0), x^{\text{Im}}(1), \dots, x^{\text{Im}}(N-1)]^T \\ \bar{\mathbf{v}} &= [v^{\text{Re}}(0), v^{\text{Re}}(1), \dots, v^{\text{Re}}(N-1), v^{\text{Im}}(0), v^{\text{Im}}(1), \dots, v^{\text{Im}}(N-1)]^T \\ \bar{\mathbf{v}} &= [a_1^{\text{Re}}, a_2^{\text{Re}}, \dots, a_p^{\text{Re}}, a_1^{\text{Im}}, a_2^{\text{Im}}, \dots, a_p^{\text{Im}}]^T \end{aligned}$$

Ceci nous donne immédiatement [43] :

$$\bar{\mathbf{x}} = \bar{\mathbf{A}}\bar{\mathbf{a}} + \bar{\mathbf{V}} \quad (2.8)$$

$$\text{où : } \bar{\mathbf{A}} = \begin{bmatrix} \text{Re}\{\mathbf{A}\} & -\text{Im}\{\mathbf{A}\} \\ \text{Im}\{\mathbf{A}\} & \text{Re}\{\mathbf{A}\} \end{bmatrix} \quad (2.7)$$

Cette fois-ci, l'expression (2.8) est un système d'équations à valeurs réelles dont la solution nous fournit la partie réelle et la partie imaginaire des amplitudes des raies. Les expressions (2.3) et (2.8) sont des régressions linéaire dont la méthode des moindres carrés pour estimer le vecteur \mathbf{a} ou $\bar{\mathbf{a}}$ consiste à minimiser les fonctions de coût :

$$j(\hat{\mathbf{a}}) = \|\mathbf{x} - \mathbf{A}\hat{\mathbf{a}}\|^2 \quad (2.9)$$

$$j(\hat{\bar{\mathbf{a}}}) = \|\bar{\mathbf{x}} - \bar{\mathbf{A}}\hat{\bar{\mathbf{a}}}\|^2 \quad (2.10)$$

Les solutions des expressions (2.9) et (2.10) utilisant la méthode des moindres carrés sont donnée par :

$$\hat{\mathbf{a}} = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \mathbf{x} \quad (2.11)$$

$$\hat{\bar{\mathbf{a}}} = (\bar{\mathbf{A}}^T \bar{\mathbf{A}})^{-1} \bar{\mathbf{A}}^T \bar{\mathbf{x}} \quad (2.12)$$

2.4. Estimateur neuronal

L'idée d'introduire le réseau de Hopfield dans le problème d'optimisation revient à trouver un lien entre les fonctions de coût à minimiser $j(\hat{\mathbf{a}})$, $j(\hat{\bar{\mathbf{a}}})$ et la fonction d'énergie du réseau de Hopfield à valeurs réelles [41] et celui à valeurs complexes [41] données par :

$$E_R = -\frac{1}{2} \mathbf{v}^T \mathbf{W} \mathbf{v} - \mathbf{b}^T \mathbf{v} \quad (2.13)$$

$$E_C = -\frac{1}{2} (\mathbf{v}^H \mathbf{W} \mathbf{v} + \mathbf{b}^H \mathbf{v} + \mathbf{v}^H \mathbf{b}) \quad (2.14)$$

où $(\cdot)^T$, $(\cdot)^H$ dénotent le transpose et le transpose conjugué respectivement .

2.4.1. Réseau de Hopfield à valeurs réelles

L'architecture du réseau de Hopfield à valeurs réelles est déduite à partir de la fonction à minimiser donnée dans l'expression (2.10). Cette dernière est équivalente à :

$$j(\hat{\bar{\mathbf{a}}}) = \frac{1}{2} \hat{\bar{\mathbf{a}}}^T \bar{\mathbf{A}}^T \bar{\mathbf{A}} \hat{\bar{\mathbf{a}}} - \bar{\mathbf{x}}^T \bar{\mathbf{A}} \hat{\bar{\mathbf{a}}} \quad (2.15)$$

On note ici qu'un terme $\bar{\mathbf{x}}^T \bar{\mathbf{x}}$ a été ignoré du fait qu'il ne dépend pas de $\hat{\bar{\mathbf{a}}}$.

En comparant l'équation (2.13) et l'équation (2.15) on peut obtenir la matrice des poids des connections \mathbf{W} et le vecteur des biais \mathbf{I} du réseau de Hopfield. Le vecteur des états du réseau de Hopfield \mathbf{v} correspond aux paramètres à estimer $\hat{\bar{\mathbf{a}}}$:

$$\begin{cases} \mathbf{W} = -\overline{\mathbf{A}^T \mathbf{A}} \\ \mathbf{b} = \overline{\mathbf{A}^T \mathbf{x}} \\ \mathbf{v} = \overline{\mathbf{a}} \end{cases} \quad (2.16)$$

Pour le $i^{\text{ème}}$ neurone, le signal d'entrée (ou bien l'activation du neurone) est donné par :

$$u_i = \sum_{m=1}^{2p} w_{im} v_m + b_i \quad (2.17)$$

où w_{im} est le poids de connexion à valeurs réelles entre les deux neurones i et m , b_i est la valeur du biais du neurone i et v_m est le signal de sortie du neurone m donné par $v_m = f(u_m)$.

La dynamique du réseau à valeurs réelles est régie par l'équation différentielle [40]:

$$\frac{du_i}{dt} = -\frac{u_i}{\tau_i} + \sum_{m=1}^{2p} w_{im} v_m + b_i, \quad v_m(t) = f(u_m(t)), \quad i = 1, 2, \dots, 2p \quad (2.18)$$

Le modèle discret équivalent est obtenu par l'utilisation de la formule d'Euler dans l'expression (2.18).

$$\frac{u_i(k+1) - u_i(k)}{h} = -\frac{u_i(k)}{\tau_i} + \sum_{m=1}^{2p} w_{im} v_m + b_i \quad (2.19)$$

$$v_m(k+1) = f(u_m(k+1)), \quad i = 1, 2, \dots, 2p$$

où h est le pas d'intégration.

L'implémentation du réseau dans ces deux versions est déduite directement des équations (2.18), (2.19). Elle est représentée sur la figure (2.4).

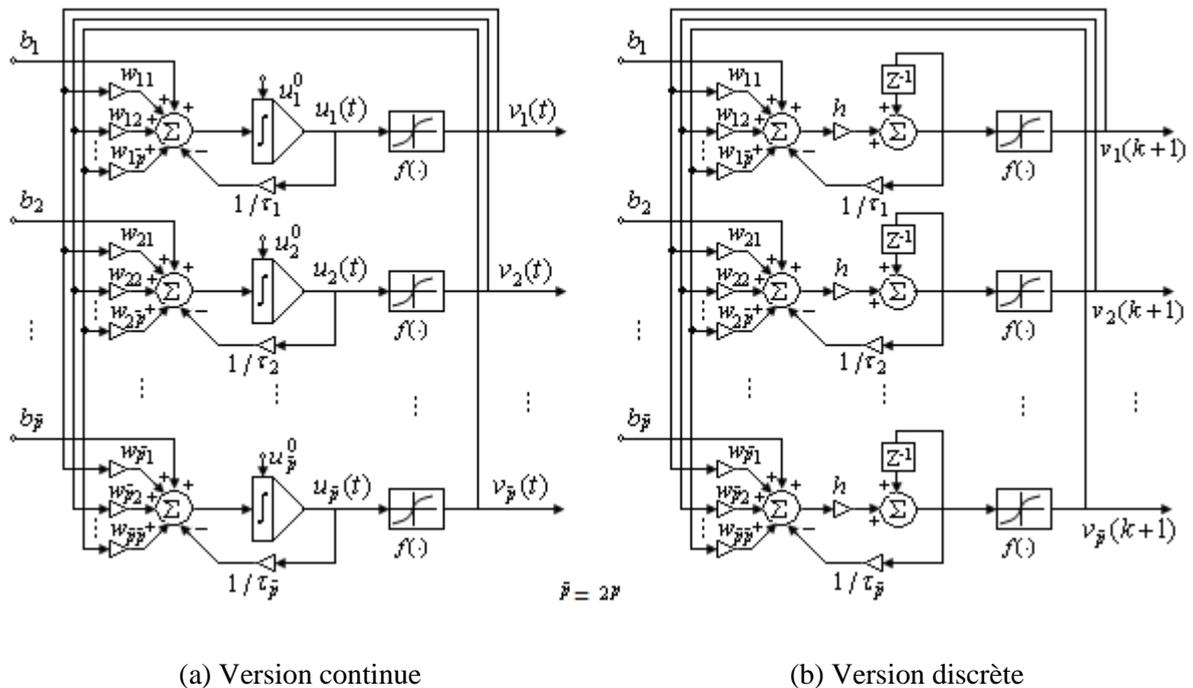


Figure 2.4. Architecture du réseau de Hopfield.

2.4.2. Réseau de Hopfield à valeurs complexes

De la même manière qu'on a fait pour le réseau à valeurs réelles, L'architecture du réseau à valeurs complexes est déduite de l'expression (2.11) qui est équivalente à :

$$j(\hat{\mathbf{a}}) = \frac{1}{2}(\mathbf{a}^H \mathbf{A}^H \mathbf{A} \mathbf{a} - \mathbf{x}^H \mathbf{A} \mathbf{a} - \mathbf{a}^H \mathbf{A}^H \mathbf{x}) \quad (2.20)$$

En liant l'équation (2.20) et l'équation (2.14), on déduit facilement les paramètres du réseau.

$$\begin{cases} \mathbf{W} = -\mathbf{A}^H \mathbf{A} \\ \mathbf{b} = \mathbf{A}^H \mathbf{x} \\ \mathbf{v} = \mathbf{a} \end{cases} \quad (2.21)$$

La structure du réseau de Hopfield à valeurs complexes est similaire à celle du réseau à valeurs réelles. La seule différence réside dans le fait que les signaux d'entrée, les poids, les biais et les signaux de sortie sont des nombres complexes [70-74].

Pour le $i^{\text{ème}}$ neurone, le signal d'entrée (ou bien l'activation du neurone) est donné par :

$$u_i = \sum_{m=1}^p w_{im} v_m + b_i \quad (2.22)$$

ou sous la forme matricielle:

$$\mathbf{U} = \mathbf{WV} + \mathbf{B} \quad (2.23)$$

w_{im} est le poids de connexion à valeur complexe entre les deux neurones i et m et b_i la valeur du biais du neurone i . v_m est le signal de sortie du neurone m donné par :

$$v_m = f_c(u_m) = f_c(u_m^{\text{Re}} + i u_m^{\text{Im}}) \quad (3.24)$$

où u_m^{Re} et u_m^{Im} sont la partie réelle et la partie imaginaire de l'entrée du neurone m . $f_c(\cdot)$ est la fonction d'activation complexe du réseau. En séparant en partie réelle et partie complexe, la fonction $f_c(\cdot)$ peut être exprimée sous la forme:

$$f_c(u_m) = f_R(u_m^{\text{Re}}) + i f_R(u_m^{\text{Im}}) \quad (2.25)$$

où $f_R(\cdot)$ est une n'importe qu'elle fonction d'activation pour le réseau à valeurs réelles.

$$v_m = v_m^{\text{Re}} + i v_m^{\text{Im}} = f_R\left(\left[\mathbf{W}^{\text{Re}} \quad -\mathbf{W}^{\text{Im}}\right] \cdot \begin{bmatrix} \mathbf{U}^{\text{Re}} \\ \mathbf{U}^{\text{Im}} \end{bmatrix} + \mathbf{B}^{\text{Re}}\right) + i f_R\left(\left[\mathbf{W}^{\text{Im}} \quad \mathbf{W}^{\text{Re}}\right] \cdot \begin{bmatrix} \mathbf{U}^{\text{Re}} \\ \mathbf{U}^{\text{Im}} \end{bmatrix} + \mathbf{B}^{\text{Im}}\right) \quad (2.26)$$

La dynamique du réseau à valeurs complexes est régie par une équation différentielle similaire à celle du réseau à valeurs réelles à l'exception que les termes u_i, v_m, b_i soient des valeurs complexes:

$$\frac{du_i}{dt} = -\frac{u_i}{\tau_i} + \sum_{m=1}^p w_{im} v_m + b_i, \quad v_i(t) = f_c(u_i(t)), \quad i = 1, 2, \dots, p \quad (2.27)$$

Pour implémenter le réseau dans un hardware analogique, réécrivons l'équation (2.27) sous la forme :

$$\begin{aligned} \frac{d}{dt}(u_i^{\text{Re}} + iu_i^{\text{Im}}) &= -\frac{1}{\tau_i}(u_i^{\text{Re}} + iu_i^{\text{Im}}) + \sum_{m=1}^p w_{im}(v_m^{\text{Re}} + iv_m^{\text{Im}}) + b_i^{\text{Re}} + ib_i^{\text{Im}} \\ &= -\frac{1}{\tau_i}(u_i^{\text{Re}} + iu_i^{\text{Im}}) + \sum_{m=1}^p (w_{im}^{\text{Re}} + iw_{im}^{\text{Im}})_{ij}(v_m^{\text{Re}} + iv_m^{\text{Im}}) + b_i^{\text{Re}} + ib_i^{\text{Im}} \\ &= -\frac{1}{\tau_i}(u_i^{\text{Re}} + iu_i^{\text{Im}}) + \sum_{m=1}^p (w_{im}^{\text{Re}} v_m^{\text{Re}} - w_{im}^{\text{Im}} v_m^{\text{Im}} + i(w_{im}^{\text{Im}} v_m^{\text{Re}} - w_{im}^{\text{Re}} v_m^{\text{Im}})) + b_i^{\text{Re}} + ib_i^{\text{Im}} \end{aligned}$$

On déduit :

$$\frac{d}{dt}(u_i^{\text{Re}}) = -\frac{1}{\tau_i}(u_i^{\text{Re}}) + \sum_{m=1}^p (w_{ip}^{\text{Re}} v_m^{\text{Re}} - w_{ip}^{\text{Im}} v_m^{\text{Im}}) + b_i^{\text{Re}} \quad (2.28)$$

$$\frac{d}{dt}(u_i^{\text{Im}}) = -\frac{1}{\tau_i}(u_i^{\text{Im}}) + \sum_{m=1}^p (w_{ip}^{\text{Im}} v_m^{\text{Re}} - w_{ip}^{\text{Re}} v_m^{\text{Im}}) + b_i^{\text{Im}} \quad (2.29)$$

Le circuit électrique du réseau à valeurs complexes est constitué de deux réseaux similaires à celui à valeurs réelles représenté dans la figure (2.4).

2.5. Résultats des simulations

Dans toutes les simulations qu'on a fait, le signal qu'on va estimer ses paramètres est composé d'un signal utile $s(n)$ composé de deux sinusoides pures complexes de fréquences $f = [0.1 \ 0.3]$, d'amplitudes complexes $a = [1.e^{j\pi/8} \ 1.5.e^{j\pi/4}]$ noyées dans un bruit blanc $v(n)$, centré et de variance σ^2 .

$$x(n) = s(n) + v(n), \quad n = 0, 1, \dots, N-1 \quad (2.30)$$

Un ensemble de 200 blocs du signal dont chacun est constitué de 64 échantillons ont été utilisé. La qualité du signal est déterminée par le rapport signal sur bruit (SNR : signal to noise ratio) donné par :

$$SNR = 10 \log_{10} \frac{P_s}{P_v} \quad (2.31)$$

où P_s, P_v désignent la puissance du signal et celle du bruit respectivement.

L'erreur quadratique moyenne (EQM) utilisée pour tester les performances des estimateurs est donné par :

$$EQM\{\mathbf{a}\} = \frac{1}{M} \sum_{i=1}^M \|\hat{\mathbf{a}}(i) - \mathbf{a}\|^2 \quad (2.32)$$

où $\hat{\mathbf{a}}(i)$ est le vecteur des amplitudes estimées utilisant le $i^{\text{ème}}$ bloc et M étant le nombre de réalisations. Une condition nécessaire pour la convergence du réseau de Hopfield et que la fonction d'activation doit être strictement croissante. Deux types de fonction ont été utilisés ; la fonction linéaire et la fonction sigmoïde. Or pour notre cas, on a introduit une fonction sigmoïde modifiée de la forme :

$$v_i = \frac{2\alpha}{1 + e^{-\beta \cdot u_i}} - \alpha \quad (2.33)$$

La constante α est choisie arbitraire à condition que toutes les valeurs des amplitudes soient comprises entre $-\alpha$ et $+\alpha$, le paramètre β représente la pente de la fonction d'activation. Le réseau est initialisé par un vecteur d'état aléatoire.

2.5.1. Fonction d'activation linéaire

La figure (2.5) montre la moyenne et l'écart-type de la partie réelle et de la partie imaginaire des amplitudes estimées. On constate que les amplitudes estimées par le réseau de Hopfield tendent vers les vraies valeurs en augmentant le SNR. La figure (2.6) donne une comparaison du réseau de neurones avec la méthode des moindres carrés et celle de la transformée de Fourier discrète, on remarque que le réseau de Hopfield et la méthode des moindres carrés se coïncident. Par contre l'estimateur par la transformée discrète présente une erreur suite à son biais. On note aussi que l'erreur quadratique moyenne de la partie réelle et de la partie imaginaire sont pareilles.

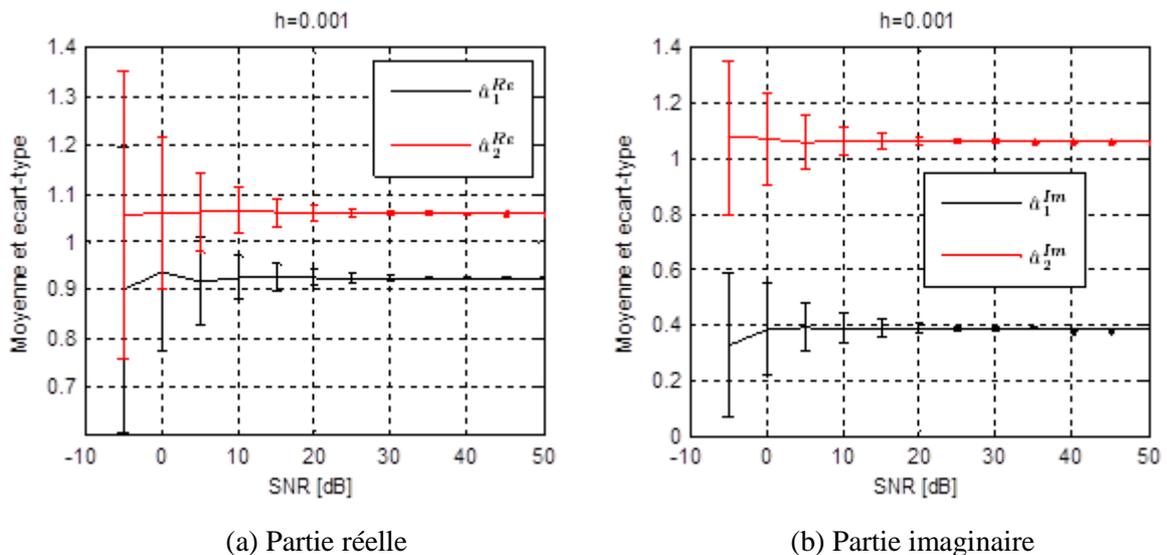


Figure 2.5. Moyenne et écart-type des amplitudes estimées.

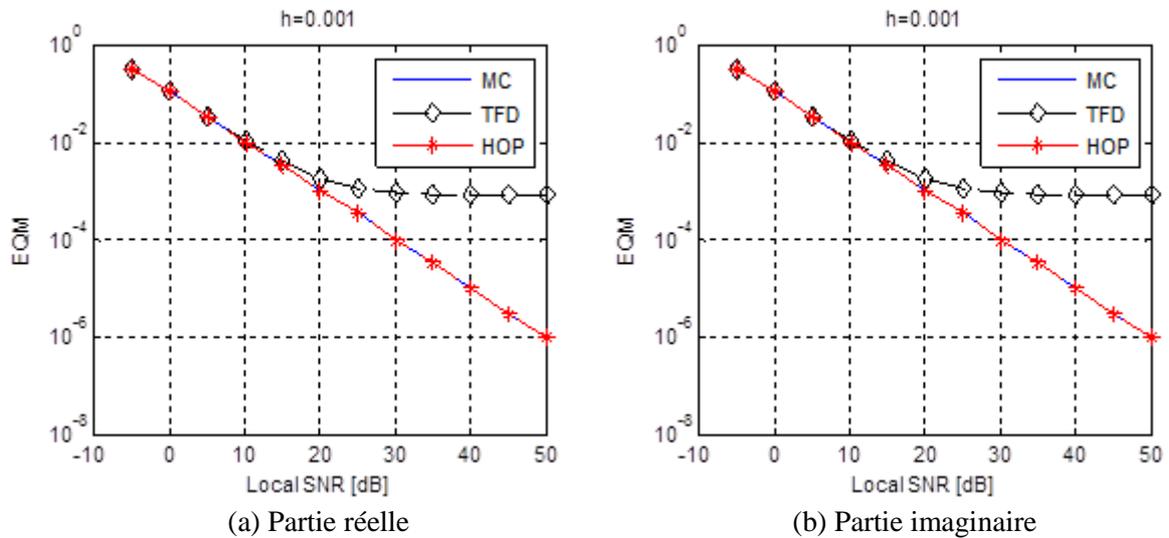


Figure 2.6. Erreur quadratique moyenne des amplitudes estimées.

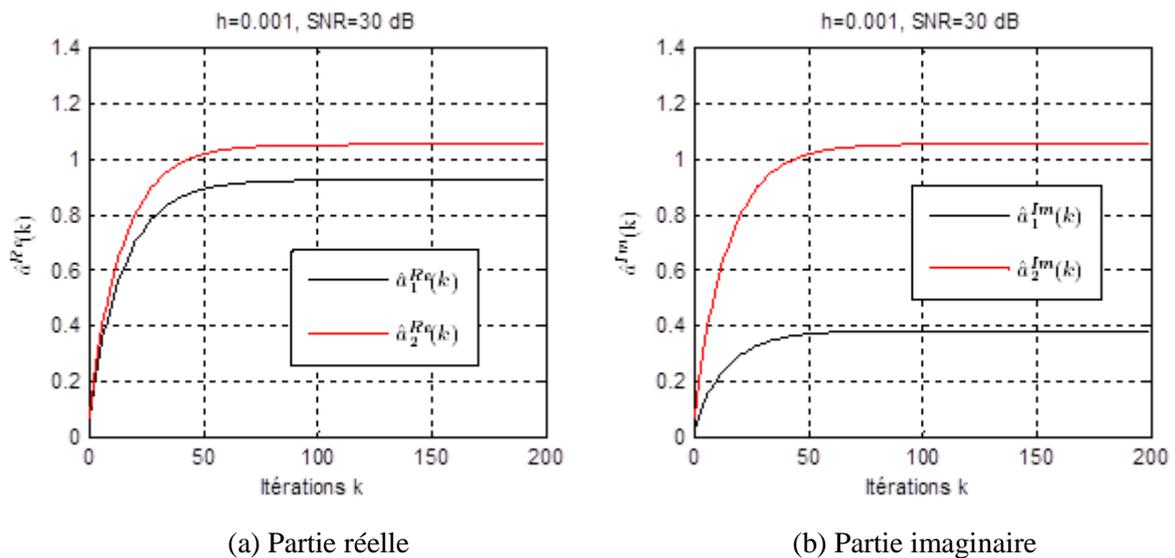


Figure 2.7. Trajectoire des amplitudes estimées.

La figure (2.7) montre l'évolution de la partie réelle et celle imaginaire des amplitudes estimées pour un $SNR = 30dB$. On remarque que quelques dizaines d'itérations sont suffisantes pour la convergence du réseau.

2.5.2. Fonction d'activation sigmoïde

Le choix des paramètres du réseau de Hopfield est un compromis entre la rapidité de la convergence et la précision des résultats. Il est clair que la valeur de α doit être choisie supérieure à la valeur maximale de la partie réelle et de la partie imaginaire de l'amplitude a_i . Pour montrer l'effet de la valeur du paramètre β , on a simulé l'exemple précédent pour différentes valeurs de β . La figure (2.8) montre l'effet de β sur l'amplitude estimée \hat{a} , on

constate que lorsque β augmente, le réseau s'accélère. Mais lorsque β devient suffisamment grande, le réseau oscille et peut se diverger.

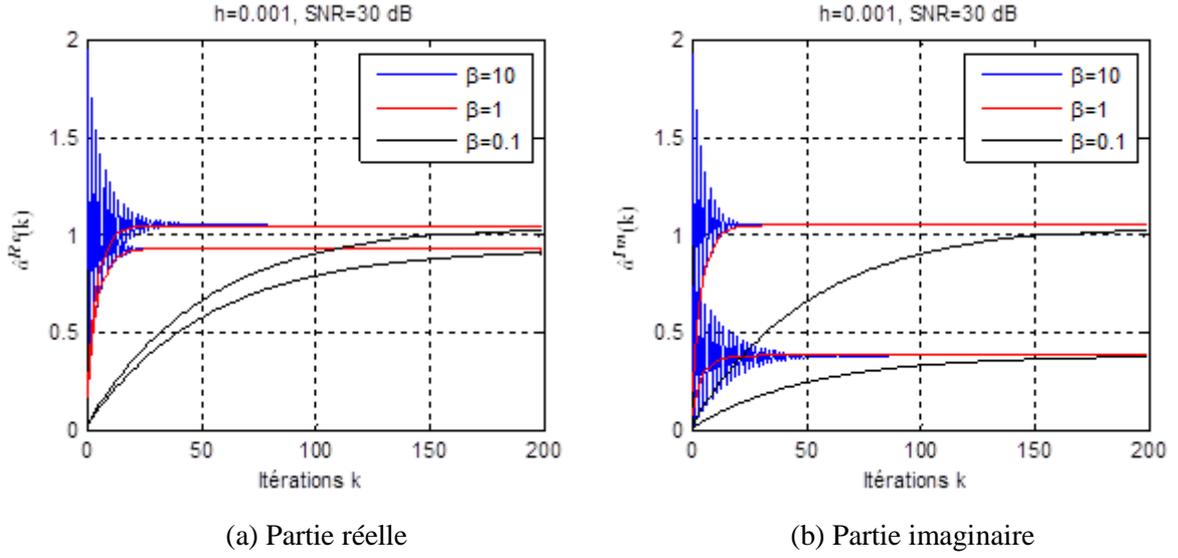


Figure 2.8. Trajectoire des amplitudes estimées en utilisant la fonction sigmoïde.

2.6. Réseau de Hopfield modifié

Dans [38], L. Han a introduit le réseau de Hopfield pour l'estimation des paramètres d'un signal sinusoïdal réel, dans cette section on propose une extension de ce réseau pour les signaux à valeurs complexes. On suppose que le signal à estimer ces paramètres est modélisé par :

$$\hat{x}(n) = \sum_{m=1}^p a_m e^{j2\pi f_m n + \varphi_m}, \quad n = 0, 1, \dots, N-1 \quad (2.34)$$

où p est le nombre de sinusoïdes présentes dans le signal, f_m , a_m et φ_m sont les fréquences, les amplitudes et les phases respectivement. Ces paramètres sont constants mais inconnus. L'objectif est de concevoir un réseau de neurones capable d'estimer ces paramètres à partir des échantillons observés.

De la même manière qu'aux sections précédentes, on ramène le problème d'estimation à un problème de minimisation. On définit une fonction d'erreur:

$$E = \frac{1}{2} \sum_{n=0}^{N-1} \left[x(n) - \sum_{m=1}^p a_m e^{j2\pi f_m n + \varphi_m} \right]^2 \quad (2.35)$$

Si on définit :

$$d(n) = x(n) - \sum_{m=1}^p a_m e^{j2\pi f_m n + \varphi_m} \quad (2.36)$$

On aura :

$$E = \frac{1}{2} \sum_{n=0}^{N-1} [d(n)]^2 = \frac{1}{2} \sum_{n=0}^{N-1} d(n)d^*(n) \quad (2.37)$$

En utilisant l'approche générale du gradient pour la minimisation d'une fonction [38-39] :

$$\frac{\partial E}{\partial a_m} = -\frac{da_m}{dt} \quad (2.38)$$

$$\frac{\partial E}{\partial \varphi_m} = -\frac{d\varphi_m}{dt} \quad (2.39)$$

$$\frac{\partial E}{\partial f_m} = -\frac{df_m}{dt} \quad (2.40)$$

on déduit la variation de l'énergie dans le temps comme:

$$\frac{dE}{dt} = \sum_{m=1}^p \frac{\partial E}{\partial a_m} \frac{da_m}{dt} + \frac{\partial E}{\partial f_m} \frac{df_m}{dt} + \frac{\partial E}{\partial \varphi_m} \frac{d\varphi_m}{dt} = -\sum_{m=1}^p \left| \frac{\partial E}{\partial a_m} \right|^2 + \left| \frac{\partial E}{\partial f_m} \right|^2 + \left| \frac{\partial E}{\partial \varphi_m} \right|^2 \leq 0 \quad (2.41)$$

où:

$$\frac{da_m}{dt} = -\frac{\partial E}{\partial a_m} = \sum_{n=0}^{N-1} e^{j(2\pi f_m n + \varphi_m)} d(n) \quad (2.42)$$

$$\frac{df_m}{dt} = -\frac{\partial E}{\partial f_m} = \sum_{n=0}^{N-1} a_m j 2\pi n e^{j(2\pi f_m n + \varphi_m)} d(n) \quad (2.43)$$

$$\frac{d\varphi_m}{dt} = -\frac{\partial E}{\partial \varphi_m} = \sum_{n=0}^{N-1} a_m e^{j(2\pi f_m n + \varphi_m)} d(n) \quad (2.44)$$

Pour faciliter l'écriture, on définit le vecteur des paramètres :

$$p_m = [a_m \ f_m \ \varphi_m] \quad (2.45)$$

La décomposition de l'expression (2.36) en partie réelle et partie imaginaire nous donne:

$$d(n) = d_{\text{Re}}(n) + j d_{\text{Im}}(n) \quad (2.46)$$

avec :

$$d_{\text{Re}}(n) = x_{\text{Re}}(n) - \sum_{m=1}^p a_m \cos(n 2\pi f_m + \varphi_m) \quad (2.47)$$

$$d_{\text{Im}}(n) = x_{\text{Im}}(n) - \sum_{m=1}^p a_m \sin(n 2\pi f_m + \varphi_m) \quad (2.48)$$

L'expression de la fonction d'erreur devient :

$$E = \frac{1}{2} \sum_{n=0}^{N-1} (d_{\text{Re}}(n) + j d_{\text{Im}}(n))(d_{\text{Re}}(n) - j d_{\text{Im}}(n)) = E_{\text{Re}} + E_{\text{Im}} \quad (2.49)$$

$$\begin{aligned}
 E_{\text{Re}} &= \frac{1}{2} \sum_{n=0}^{N-1} (d_{\text{Re}}(n))^2 \\
 E_{\text{Im}} &= \frac{1}{2} \sum_{n=0}^{N-1} (d_{\text{Im}}(n))^2
 \end{aligned}
 \tag{2.50}$$

La variation de l'énergie en fonction du temps :

$$\frac{dE}{dt} = \sum_{m=1}^p \sum_{p_m} \frac{\partial E}{\partial p_m} \frac{dp_m}{dt} = \sum_{m=1}^p \sum_{p_m} \left(\frac{\partial E_{\text{Re}}}{\partial p_m} + \frac{\partial E_{\text{Im}}}{\partial p_m} \right) \frac{dp_m}{dt}
 \tag{2.51}$$

Mais comme :

$$\frac{dp_m}{dt} = - \left(\frac{\partial E_{\text{Re}}}{\partial p_m} + \frac{\partial E_{\text{Im}}}{\partial p_m} \right)
 \tag{2.52}$$

On obtient :

$$\frac{dE}{dt} = - \sum_{m=1}^p \sum_{p_m} \left| \frac{\partial E_{\text{Re}}}{\partial p_m} + \frac{\partial E_{\text{Im}}}{\partial p_m} \right|^2 \leq 0
 \tag{2.53}$$

L'expression (2.53) garantit la convergence du réseau.

2.6.1. Architecture du réseau de Hopfield modifié

Pour la conception du réseau, on doit identifier ces entrées, ces sorties et ces poids de connexion. En remplaçant les expressions de la partie réelle et de la partie imaginaire de l'énergie dans l'expression (2.52), on obtient:

$$\begin{aligned}
 \frac{dp_m}{dt} &= - \frac{1}{2} \left(\frac{\partial}{\partial p_m} \sum_{n=0}^{N-1} (d_{\text{Re}}(n))^2 + \frac{\partial}{\partial p_m} \sum_{n=0}^{N-1} (d_{\text{Im}}(n))^2 \right) \\
 \frac{dp_m}{dt} &= - \sum_{n=0}^{N-1} \frac{\partial d_{\text{Re}}(n)}{\partial p_m} d_{\text{Re}}(n) - \sum_{n=0}^{N-1} \frac{\partial d_{\text{Im}}(n)}{\partial p_m} d_{\text{Im}}(n)
 \end{aligned}
 \tag{2.54}$$

L'expression (2.54) montre l'équation différentielle qui régit la dynamique du réseau de neurones, $d_{\text{Re}}(n), d_{\text{Im}}(n)$ constituent les entrées, p_m les sorties. Les connexions sont calculées

à partir de $\frac{\partial d_{\text{Re}}(n)}{\partial p_m}$ et $\frac{\partial d_{\text{Im}}(n)}{\partial p_m}$. On note ici qu'il y a une symétrie dans le réseau, constituée

par les partie paires et impaires de $d(n)$ et $\frac{\partial d(n)}{\partial p_m}$. Les figures (2.9) et (2.10) illustrent la

structure du réseau.

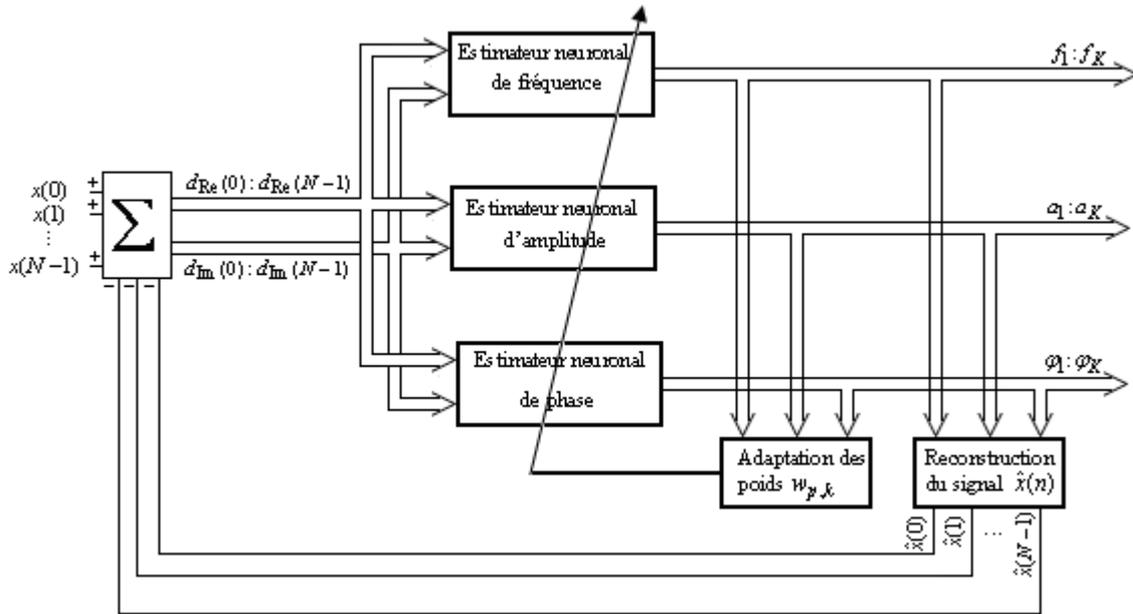


Figure 2.9. Schéma synoptique du réseau de neurones.

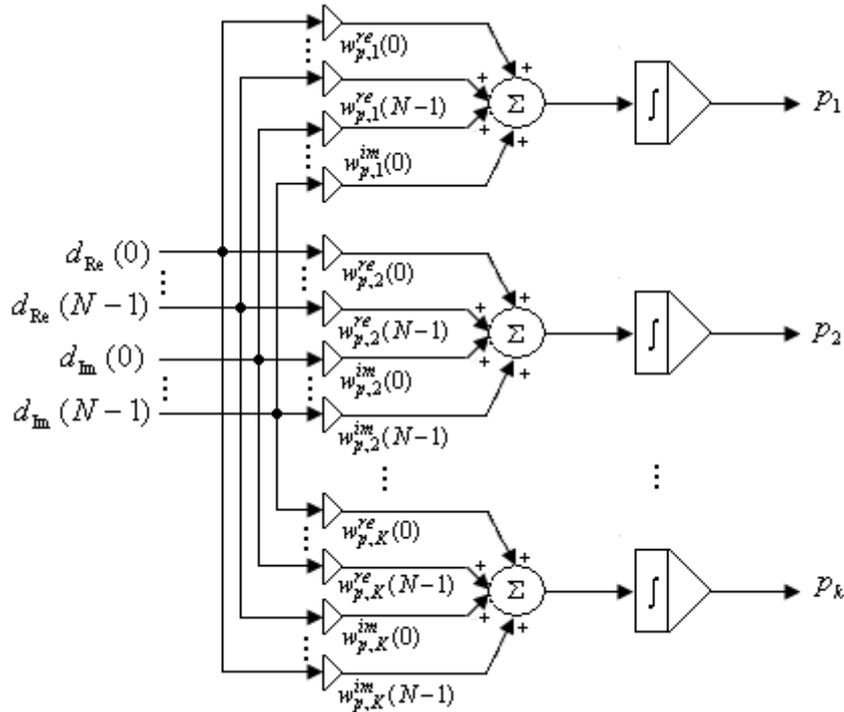


Figure 2.10. Architecture de chaque estimateur neuronal de la figure (2.9).

2.6.2. Résultats de Simulation

On considère un signal composé de deux sinusoïdes complexes de fréquences $f = [0.1 \ 0.3]$, d'amplitudes $a = [1 \ 1.5]$ et de phases $\varphi = [\pi/8 \ \pi/4]$. Le signal est corrompu par un bruit gaussien centré. Un ensemble de 200 blocs du signal dont chacun est constitué de 64 échantillons ont été utilisé. La même expérience a été répétée pour différentes valeurs du SNR. La dynamique du réseau de neurones dans cette exemple est régie par :

$$\begin{cases} \frac{da_i}{dt} = \sum_{n=0}^{N-1} (\cos(nw_i + \varphi_i))d_{\text{Re}}(n) + \sum_{n=0}^{N-1} (\sin(nw_i + \varphi_i))d_{\text{Im}}(n) \\ \frac{d\varphi_i}{dt} = -\sum_{n=0}^{N-1} a_i \sin(nw_i + \varphi_i)d_{\text{Re}}(n) + \sum_{n=0}^{N-1} a_i \cos(nw_i + \varphi_i)d_{\text{Im}}(n) \\ \frac{df_i}{dt} = -\sum_{n=0}^{N-1} n2\pi a_i \sin(nw_i + \varphi_i)d_{\text{Re}}(n) + \sum_{n=0}^{N-1} n2\pi a_i \cos(nw_i + \varphi_i)d_{\text{Im}}(n) \end{cases} \quad i = 1;2$$

avec : $d_{\text{Re}}(n) = x_{\text{Re}}(n) - a_1 \cos(nw_1 + \varphi_1) - a_2 \cos(nw_2 + \varphi_2)$

$d_{\text{Im}}(n) = x_{\text{Im}}(n) - a_1 \sin(nw_1 + \varphi_1) - a_2 \sin(nw_2 + \varphi_2)$

Vu le nombre des paramètres à estimer (6), la représentation de la fonction de coût n'est pas évidente, pour cela on représente cette fonction en variant deux paramètres et en fixant les autres à chaque fois. La figure (2.11) illustre la fonction de coût pour chaque cas. On constate dans les deux premiers cas qu'il n'existe qu'un seul minimum global, alors pour le cas où les fréquences varient, on constate la présence d'un nombre important de minimums locaux en plus du minimum global d'où la probabilité pour que le réseau converge vers un minimum local est grande.

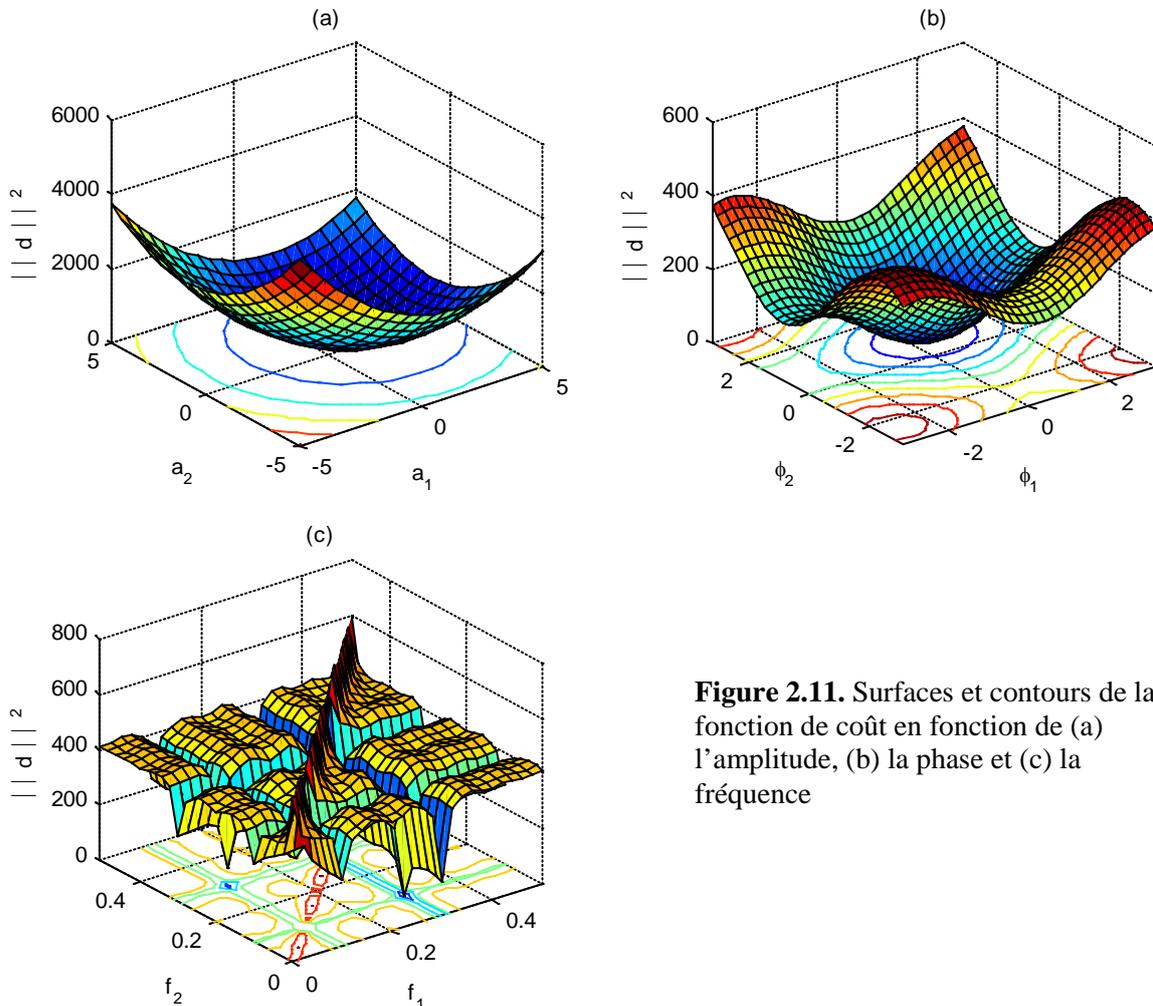
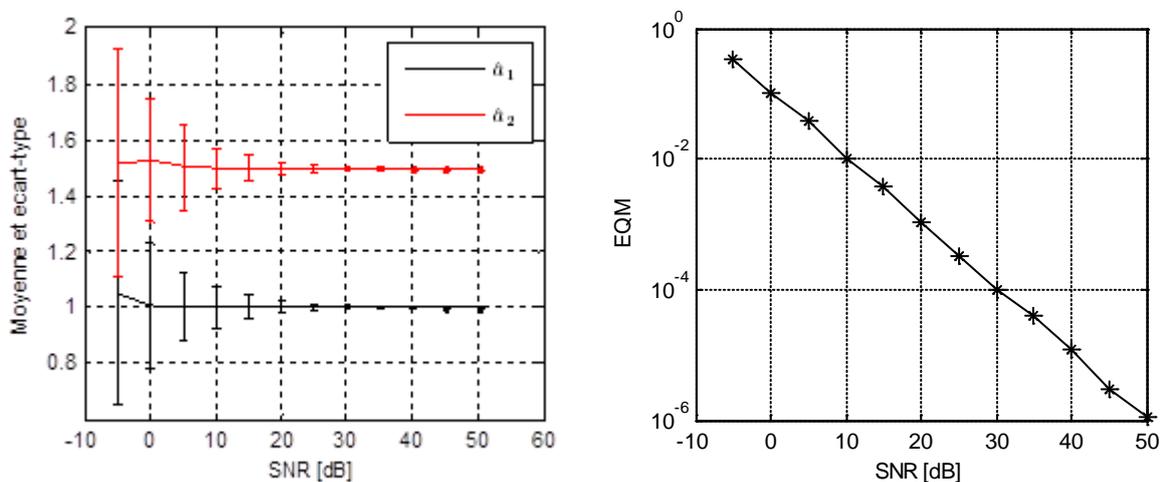


Figure 2.11. Surfaces et contours de la fonction de coût en fonction de (a) l'amplitude, (b) la phase et (c) la fréquence

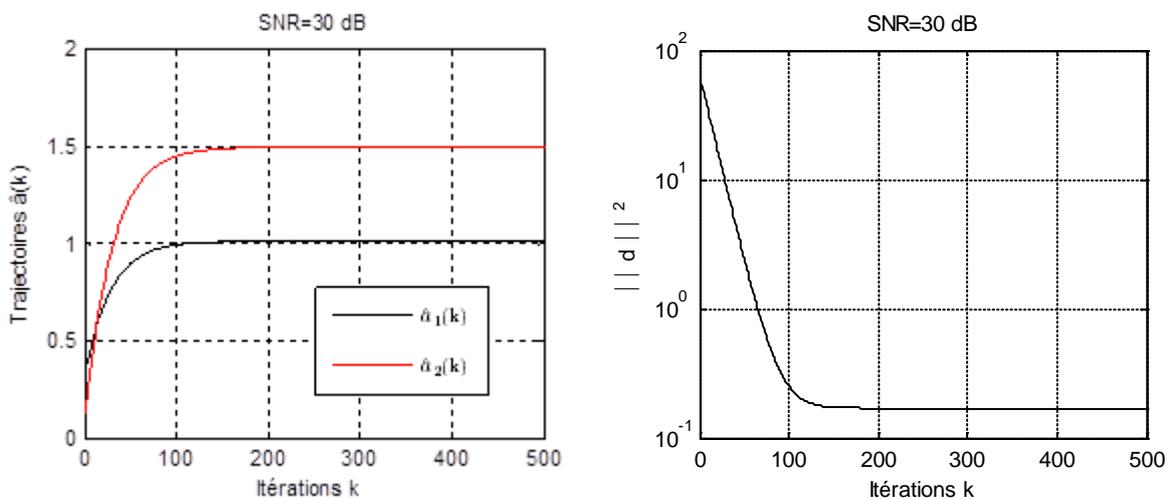
2.6.2.1 Amplitudes inconnues

Dans le premier exemple, on suppose que les fréquences et les phases sont connues, seules les amplitudes sont à estimer. Le réseau est initialisé par des valeurs aléatoires uniformément repartis entre 0 et 1. Dans la figure (2.12.a) on montre la moyenne et l'écart-type des paramètres a_1, a_2 . On constate que la moyenne des paramètres tend vers les vraies valeurs pour un $SNR \geq 5dB$, mais avec un écart-type grand. Lorsque le SNR augmente, l'écart-type tend vers zéro. Quant à l'erreur quadratique moyenne, elle tend asymptotiquement vers zéro. Ceci est montré dans la figure (2.12.b). La figure (2.13) montre les trajectoires des paramètres a_1, a_2 et l'évolution de la fonction du coût pour un $SNR = 30dB$.



(a) Moyenne et écart-type (b) Erreur quadratique

Figure 2.12. Performances du réseau vis-à-vis le SNR pour l'estimation des amplitudes.



(a) Trajectoires (b) Erreur résiduelle

Figure 2.13. Trajectoires des amplitudes estimées et erreur résiduelle pour un $SNR = 30dB$.

2.6.2.2. Phases inconnues

Dans ce deuxième cas on suppose que les fréquences et les amplitudes sont connues, seules les phases sont à estimer. Le réseau est initialisé par des valeurs aléatoires uniformément repartis entre 0 et 2π . La figure (2.14) montre les propriétés statistiques des phases estimées. Quant à la figure (2.15), elle montre l'évolution des phases estimées ainsi que la fonction-erreur pour un $SNR = 30dB$. La même discussion qu'on a faite dans le cas précédent reste valable dans ce deuxième cas.

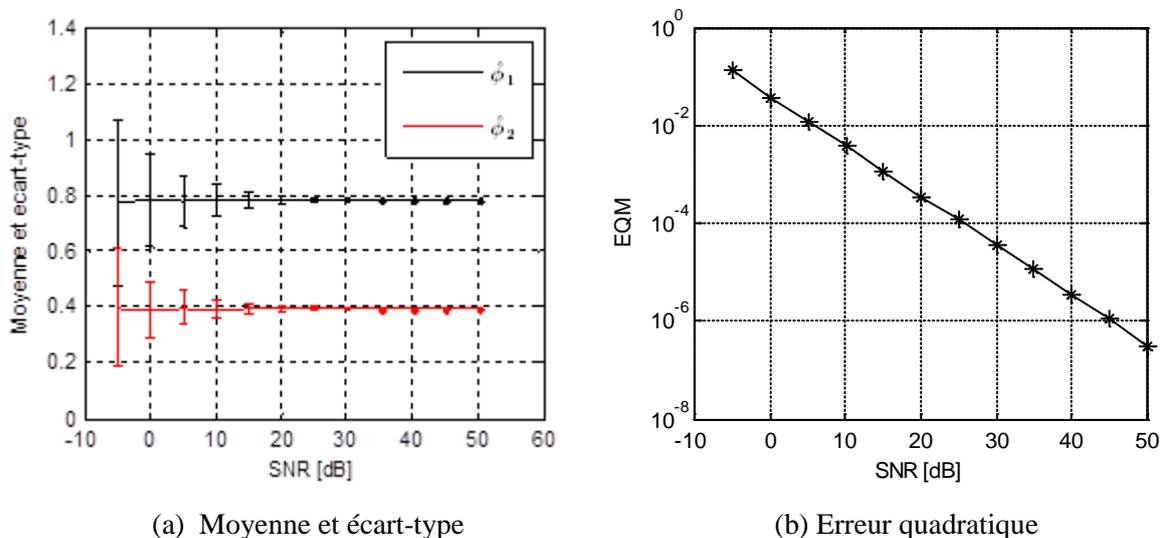


Figure 2.14. Performances du réseau vis-à-vis le SNR pour l'estimation des phases.

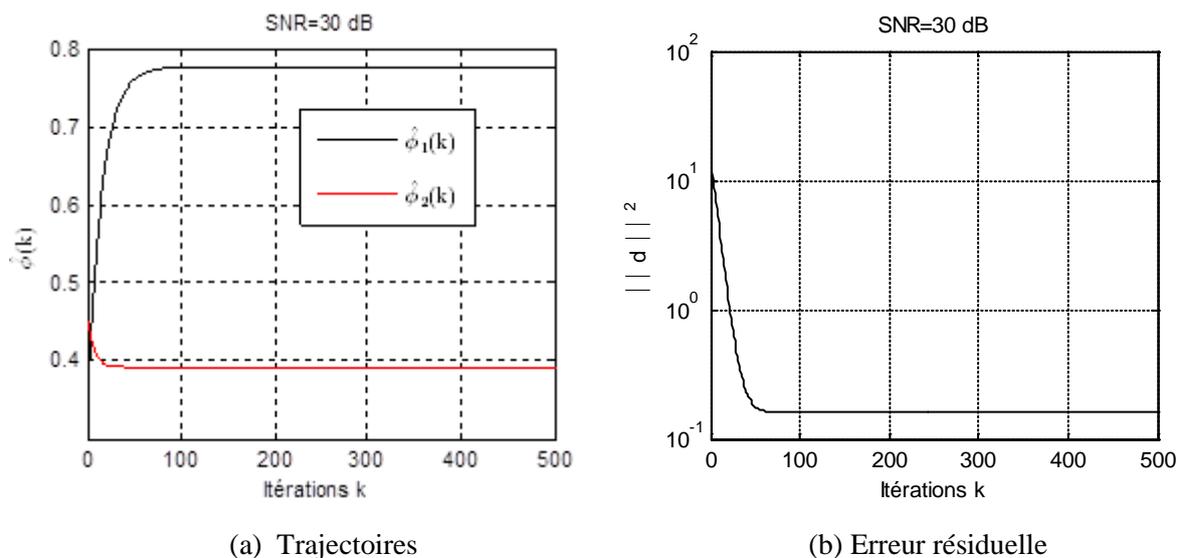


Figure 2.15. Trajectoires des phases estimées et erreur résiduelle pour un $SNR = 30dB$.

2.6.2.3. Fréquences inconnues

Le troisième cas est plus délicat, la présence d'un grand nombre de minimums locaux rend le réseau très sensible aux conditions initiales $f_0 = [f_{10} \ f_{20}]$. En effet le réseau converge vers le minimum local le plus près de l'état initial du réseau. Une solution consiste à initialiser le réseau par des fréquences initiales voisines des fréquences réelles. Ceci est possible si on précède le processus itératif par une estimation primaire des fréquences en utilisant l'une des méthodes conventionnelles comme le périodogrammes à titre d'exemple.

On note ici que le pas de convergence doit être choisi très petit pour éviter l'instabilité du réseau. Dans notre cas, on a fixé $h = 10^{-8}$.

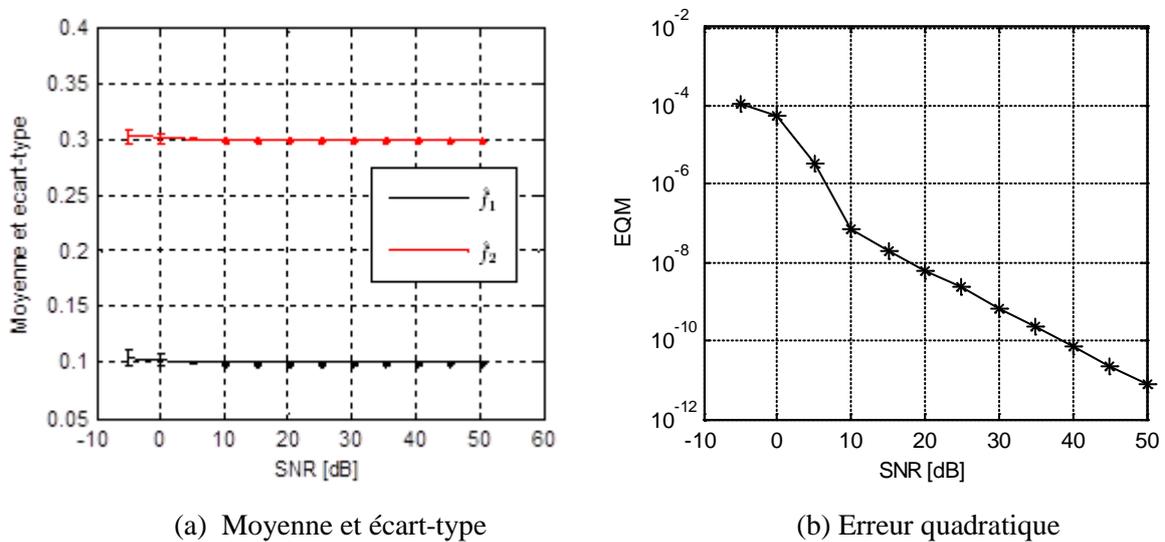


Figure 2.16. Performances du réseau vis-à-vis le SNR pour l'estimation des fréquences.

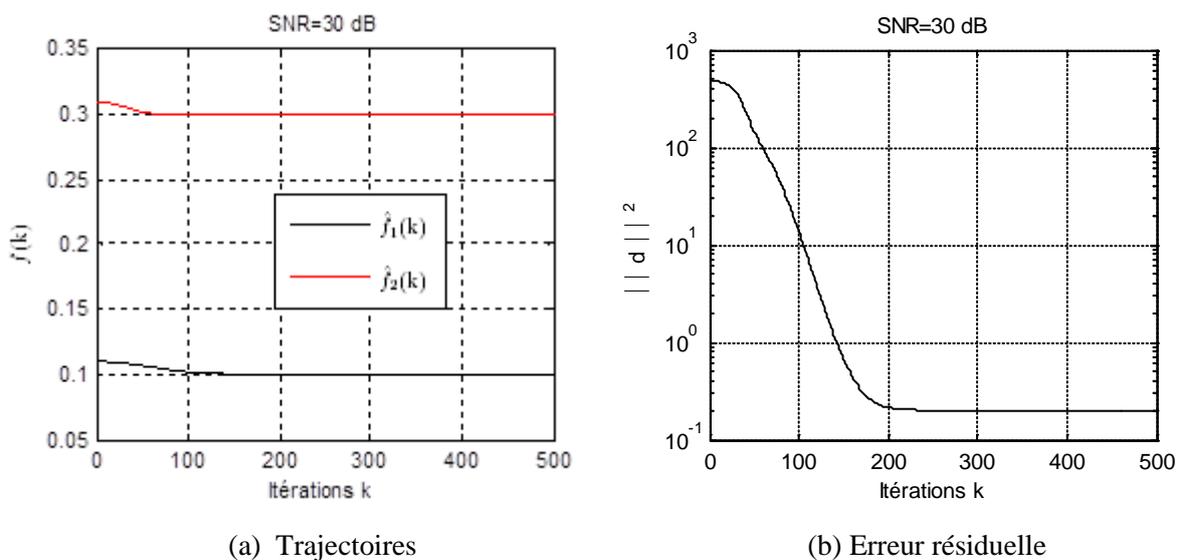


Figure 2.17. Trajectoires des phases estimées et erreur résiduelle pour un $SNR = 30dB$.

La figure (2.16) montre les propriétés statistiques des fréquences estimées pour un état initial $f_0 = [0.1 + 0.6/N, 0.3 + 0.6/N]$. On constate que le réseau atteint facilement les vraies fréquences même si le signal est fortement bruité. Ceci est dû au choix judicieux des fréquences initiales. Quant à la figure (2.17), elle montre l'évolution des deux fréquences ainsi que la norme de l'erreur entre le signal observé et le signal estimé. On remarque que le réseau converge vers les vraies fréquences après quelques dizaines d'itérations.

2.6.2.4. Amplitudes, phases et fréquences inconnues

Dans ce dernier cas on considère que tous les paramètres du signal sont inconnus. Le réseau est initialisé par les valeurs utilisées dans les trois cas précédents. On constate d'après la figure (2.18) que les amplitudes ainsi que les phases estimées tendent vers les vraies valeurs en augmentant le SNR. On note ici que les fréquences estimées sont plus proches des valeurs réelles suite aux fréquences initiales prises très proches des fréquences vraies.

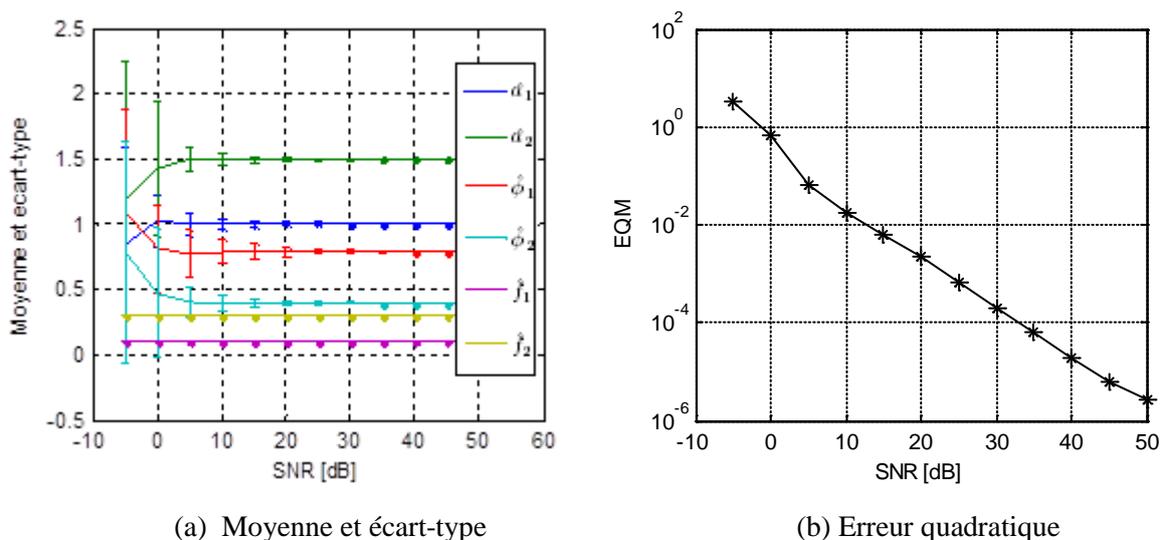


Figure 2.18. Performances du réseau vis-à-vis le SNR pour l'estimation des trois paramètres.

2.7. Conclusion

Dans ce chapitre, nous avons présenté deux types de réseaux récurrents pour l'estimation des paramètres d'un signal composé de sinusoides complexes. Le premier consiste à estimer les amplitudes des sinusoides en supposant que les fréquences sont connues. Ce réseau repose sur le modèle de Hopfield original. Deux versions du réseau ont été discutées, le premier consiste à diviser le signal d'entrée en partie réelle et partie imaginaire puis on construit une nouvelle régression linéaire purement réelle dont la solution est directement adaptable au réseau de Hopfield original. La deuxième version consiste à utiliser

la régression linéaire complexe comme telle quelle est. Le réseau construit est composé de deux parties similaires, l'un pour la partie réelle, l'autre pour la partie imaginaire.

Le deuxième réseau étudié sert à estimer l'amplitude, la phase et la fréquence des sinusoides simultanément. Les résultats des simulations montrent le réseau basé sur le modèle de Hopfield donne des valeurs optimales des amplitudes estimées. Par contre le deuxième réseau, il fournit des bonnes estimations des amplitudes et des phases dans le cas où les fréquences sont connues. Mais pour estimer les trois paramètres simultanément, les fréquences estimées doivent être initialisées par des valeurs proches des vraies fréquences pour éviter les minimums locaux.

Chapitre 3

Estimateur neuronal d'un modèle auto-régressif

3.1. Introduction

L'estimation spectrale auto-régressive (AR) consiste à estimer les paramètres AR puis de les restituer dans l'expression théorique de la densité spectrale de puissance. Quand l'ordre du modèle est modéré, le célèbre algorithme de Levinson est très efficace pour calculer ces coefficients. Mais quand l'ordre du système devient large, la solution du système devient une tâche difficile [1]. Motivé par la nature parallèle des réseaux de neurones et leur aptitude à faire des calculs complexes en temps réel, on va envisager la possibilité d'introduire les réseaux de neurones pour estimer les paramètres AR. On commence par l'utilisation d'un réseau de neurone basé sur le gradient où on minimise la norme d'un vecteur d'erreurs [75-77]. Puis, on va introduire un nouveau réseau proposé par Y. Zhang et qui est sur la minimisation d'un vecteur-erreurs élément par élément [78-81].

3.2. Positionnement du problème

On considère le problème de l'estimation des paramètres d'un processus auto-régressif stationnaire [83]:

$$x(n) = \sum_{i=1}^p a_i x(n-i) + e(n) \quad (3.1)$$

où $\mathbf{a} = [a_1, \dots, a_p]^T$ est le vecteur des paramètres AR inconnus, $\mathbf{x}(n) = [x(n-1), \dots, x(n-p)]^T$ est une régression du processus $x(n)$ d'un ordre connu p ; $e(n)$ est un processus blanc, Gaussien, centré et de variance σ^2 . Notre objectif est de chercher une estimation optimale des paramètres AR en utilisant les observations $\{x(n)\}_{n=0}^{N-1}$, où N est le nombre des échantillons disponibles. Dans les cas pratiques, $e(n)$ est plus petit par rapport à $x(n)$, donc on peut estimer $x(n)$ par [1]:

$$\hat{x}(n) = x(n) - e(n) = \sum_{i=1}^p a_i x(n-i) \quad (3.2)$$

$$x(n) \approx \hat{x}(n) = \sum_{i=1}^p a_i x(n-i) \quad (3.3)$$

En réécrivant (1) sous la forme matricielle:

$$\begin{bmatrix} x(p+1) \\ x(p+2) \\ \vdots \\ x(N) \end{bmatrix} = \begin{bmatrix} x(p) & x(p-1) & \cdots & x(1) \\ x(p+1) & x(p) & \cdots & x(2) \\ \vdots & \vdots & \ddots & \vdots \\ x(N-1) & x(N-2) & \cdots & x(N-p) \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} \quad (3.4)$$

et en posant : $\mathbf{x} = [x(p+1), x(p+2), \dots, x(N)]^T$

$$\mathbf{a} = [a_1, a_2, \dots, a_p]^T$$

$$\mathbf{S} = \begin{bmatrix} x(p) & x(p-1) & \cdots & x(1) \\ x(p+1) & x(p) & \cdots & x(2) \\ \vdots & \vdots & \ddots & \vdots \\ x(N-1) & x(N-2) & \cdots & x(N-p) \end{bmatrix}$$

L'équation (3.1) sera réécrite sous la forme:

$$\mathbf{S} \cdot \mathbf{a} = \mathbf{x} \quad (3.5)$$

Les paramètres a_i peuvent être estimés par différentes méthodes, cependant la résolution directe du système d'équation est plus avantageuse que le calcul de la matrice inverse \mathbf{S}^{-1} et sa multiplication par le vecteur \mathbf{x} . Une fois les coefficients sont calculés, la densité spectrale estimée est obtenue par l'introduction de ces paramètres dans son expression théorique [4]:

$$P(f) = \frac{\sigma_e^2}{\left| 1 + \sum_{k=1}^p a_k e^{-j2\pi k f} \right|^2} \quad (3.6)$$

3.3. Estimateur neuronal AR par gradient

Les réseaux de neurones basés sur le gradient sont largement étudiés pour la résolution des systèmes d'équations [75,77]. L'idée consiste à définir une norme d'erreur $E(t)$ puis exploiter le négatif de son gradient dans la direction descendante pour minimiser $E(t)$.

On considère la fonction à minimiser [44] :

$$E = \frac{1}{2} \|\mathbf{S} \cdot \mathbf{a} - \mathbf{x}\|^2 \quad (3.7)$$

où $\|\cdot\|$ dénote la norme-2 d'un vecteur. Le minimum de cette fonction de coût, est la solution

du système d'équation $\mathbf{x} = \mathbf{S} \cdot \mathbf{a}$.

Pour que la fonction de coût atteigne un minimum, on prend le négatif du gradient de la fonction de coût :

$$-\frac{\partial E}{\partial \mathbf{a}} = -\mathbf{S}^T (\mathbf{S}\mathbf{a} - \mathbf{x}) \quad (3.8)$$

et par l'utilisation de :

$$\frac{d\mathbf{a}}{dt} = -\frac{\partial E}{\partial \mathbf{a}} \quad (3.9)$$

on aura :

$$\dot{\mathbf{a}}(t) = \frac{d\mathbf{a}}{dt} = -\frac{\partial E}{\partial \mathbf{a}} = -\gamma \mathbf{S}^T (\mathbf{S}\mathbf{a}(t) - \mathbf{x}) \quad (3.10)$$

où $\gamma > 0$ est un paramètre pour modifier le taux de convergence.

Le réseau de neurones généralisé peut être obtenu par l'utilisation d'une fonction d'activation non linéaire $f(\cdot)$ quelconque. Dans ce cas l'équation dynamique du réseau devient [44]:

$$\dot{\mathbf{a}}(t) = -\gamma f(\mathbf{S}^T \mathbf{S}\mathbf{a}(t) - \mathbf{S}^T \mathbf{x}) \quad (3.11)$$

Si on pose $\mathbf{W} = \mathbf{S}^T \mathbf{S}$, $\mathbf{b} = \mathbf{S}^T \mathbf{x}$, l'équation (3.11) devient:

$$\dot{\mathbf{a}}(t) = -\gamma f(\mathbf{W}\mathbf{a}(t) - \mathbf{b}) \quad (3.12)$$

A partir de l'équation (3.12), on déduit la dynamique du $i^{\text{ème}}$ neurone de la forme:

$$\dot{a}_i = -\gamma f\left(\sum_{j=1}^p w_{ij} a_j - b_i\right) \quad (3.13)$$

Pour l'implémentation du réseau sur des calculateurs numériques, on applique la formule d'Euler $\dot{\mathbf{a}}(t) = (\mathbf{a}(k+1)h - \mathbf{a}(kh))/h$ pour obtenir la version discrète du GNN. k et h sont l'indice de l'itération et le pas de discrétisation respectivement.

Dans la suite de ce chapitre, on ignore la lettre h dans l'expression précédente et on note $\mathbf{a}(k+1)$ au lieu de $\mathbf{a}((k+1)h)$ pour la simplification de la lecture.

Les versions discrètes de l'équation (3.12) et (3.13) deviennent [44] :

$$\mathbf{a}(k+1) = \mathbf{a}(k) - \tau f(\mathbf{W}\mathbf{a}(k) - \mathbf{b}) \quad (3.14)$$

$$a_i(k+1) = a_i(k) - \tau f\left(\sum_{j=1}^p w_{ij} a_j(k) - b_i\right) \quad (3.15)$$

avec $\tau = \gamma h$.

La figure (3.1) illustre l'architecture du réseau de neurones basé sur le gradient dans ces deux versions, continue et discrète.

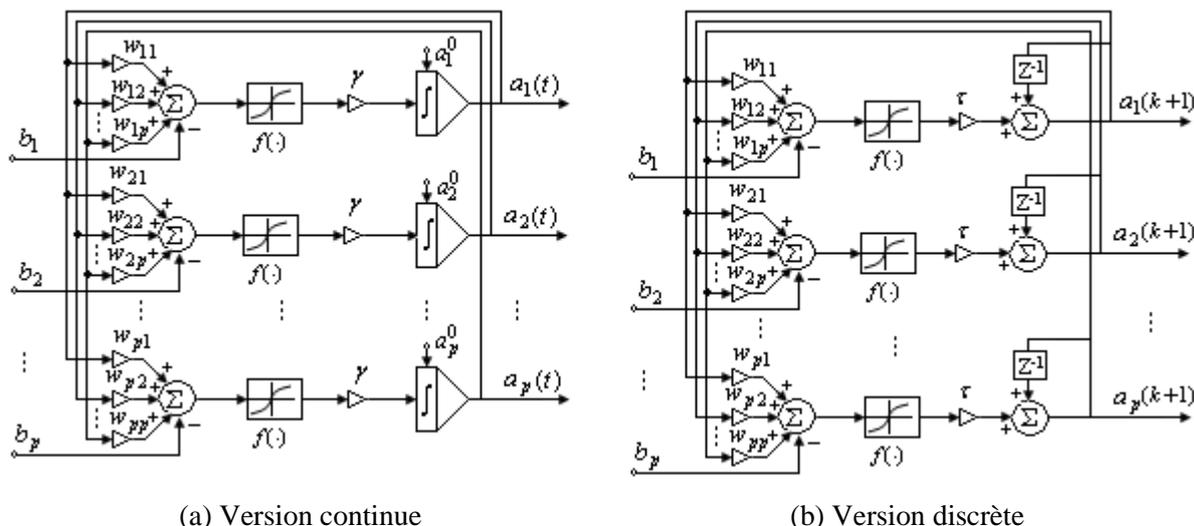


Figure 3.1. Architecture du réseau RNG.

Arrivant à ce stade, les paramètres AR sont estimés, il reste à évaluer l'expression (3.6). On constate que le dénominateur est le carré de la transformée de Fourier discrète, donc il suffit d'utiliser la transformée de Fourier rapide (FFT) et des multiplicateurs. On note ici que le nombre de points ($p+1$ points) est assez faible pour avoir une résolution fréquentielle supérieure. Une méthode pour augmenter la résolution est d'utiliser le zéro padding [53].

La figure (3.2) illustre la structure du réseau neuronal autorégressif. Dans la partie à gauche on a montré l'architecture du réseau qui estime les paramètres AR suivant l'équation (3.15) et dans la partie à droite, on a montré le circuit permettant l'évaluation de l'expression (3.6).

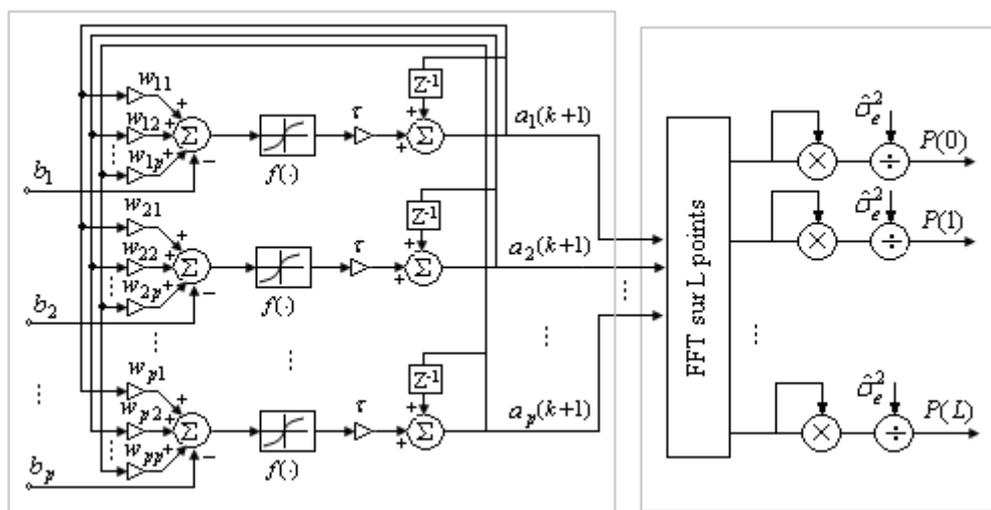


Figure 3.2. Architecture du réseau RNG pour l'estimation spectrale.

Pour la complexité du réseau, la figure (3.2) montre que le réseau nécessite $p^2 + p + 2L$ multiplications et p additions en plus de $O(L \log_2 L)$ opérations du block FFT par itération.

3.4. Estimateur AR par le réseau de Zhang

Le réseau de Zhang est basé sur la résolution du système d'équations $\mathbf{S}\mathbf{a}(t) - \mathbf{x} = 0$.

Puisque le nombre de données est généralement supérieur à l'ordre du modèle, le système d'équations peut être réécrit sous la forme [44]:

$$\mathbf{S}^T [\mathbf{S}\mathbf{a}(t) - \mathbf{x}] = 0 \quad (3.16)$$

où \mathbf{S}^T dénotes la matrice transposée de \mathbf{S}

Pour résoudre le système (3.16), on définit un vecteur-erreur $\boldsymbol{\varepsilon}(t) \in \mathbb{R}^p$ de telle sorte [78]:

$$\boldsymbol{\varepsilon}(t) = \mathbf{S}^T [\mathbf{S}\mathbf{a}(t) - \mathbf{x}] \quad (3.17)$$

puis on détermine le négatif du gradient dans la direction descendante:

$$\frac{d\boldsymbol{\varepsilon}(t)}{dt} = -\boldsymbol{\Gamma} f(\boldsymbol{\varepsilon}(t)) \quad (3.18)$$

où la matrice $\boldsymbol{\Gamma} \in \mathbb{R}^{p \times p}$ définie positive est utilisée pour modifier la cadence de convergence et $f(\cdot): \mathbb{R}^p \rightarrow \mathbb{R}^p$ dénote le vecteur-fonction d'activation.

Le développement des équations (3.17) et (3.18) nous permet d'écrire :

$$\mathbf{S}^T \dot{\mathbf{S}}\mathbf{a}(t) = -\boldsymbol{\Gamma} f(\mathbf{S}^T [\mathbf{S}\mathbf{a}(t) - \mathbf{x}]) \quad (3.19)$$

Pour assurer que tous les éléments du vecteur-erreur converge avec la même cadence et dans le même temps, il suffit de prendre $\boldsymbol{\Gamma} = \gamma \mathbf{I}$ avec $\gamma > 0$, ceci induit:

$$\mathbf{S}^T \dot{\mathbf{S}}\mathbf{a}(t) = -\gamma f(\mathbf{S}^T [\mathbf{S}\mathbf{a}(t) - \mathbf{x}]) \quad (3.20)$$

Si on pose $\mathbf{W} = \mathbf{S}^T \mathbf{S}$, $\mathbf{b} = \mathbf{S}^T \mathbf{x}$, l'équation (3.20) devient:

$$\mathbf{W}\dot{\mathbf{a}}(t) = -\gamma f(\mathbf{W}\mathbf{a}(t) - \mathbf{b}) \quad (3.21)$$

À partir de cette dernière équation, on déduit la dynamique du $i^{\text{ème}}$ neurone [44]:

$$\dot{a}_i = -\gamma f\left(\sum_{j=1}^p w_{ij} a_j - b_i\right) - \left(\sum_{j=1, j \neq i}^p w_{ij} \dot{a}_j\right) + (1 - w_{ii}) \dot{a}_i \quad (3.22)$$

La version discrète du réseau de Zhang (RNZD) est obtenue en remplaçant $\dot{\mathbf{a}}(t)$ de l'équation (3.21) par $(\mathbf{a}(k+1)h) - \mathbf{a}(kh) / h$. L'architecture du réseau de Zhang continu et sa version discrète sont illustrées dans la figure (3.3). La figure (3.4) montre l'architecture du réseau suivant les équations (3.6) et (3.22). Le réseau nécessite $2p^2 + p + 2L$ multiplications, $2p$ additions et $O(L \log_2 L)$ opérations du block FFT

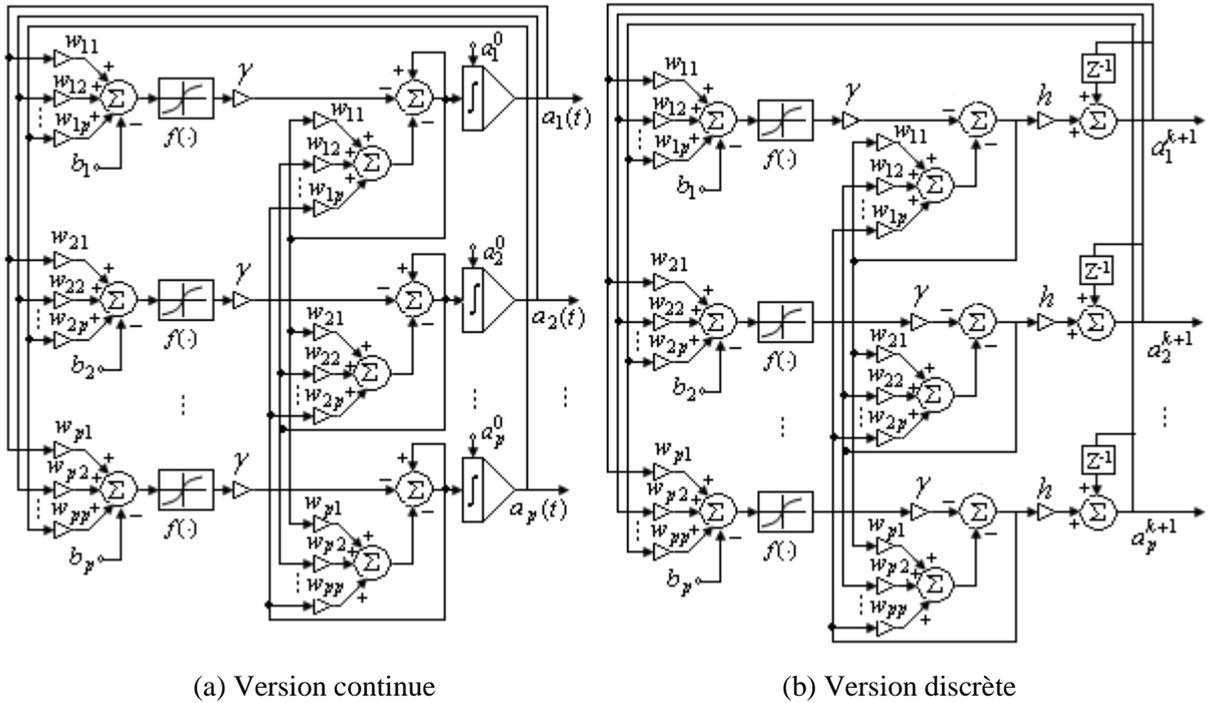


Figure 3.3. Architecture du réseau RNZ.

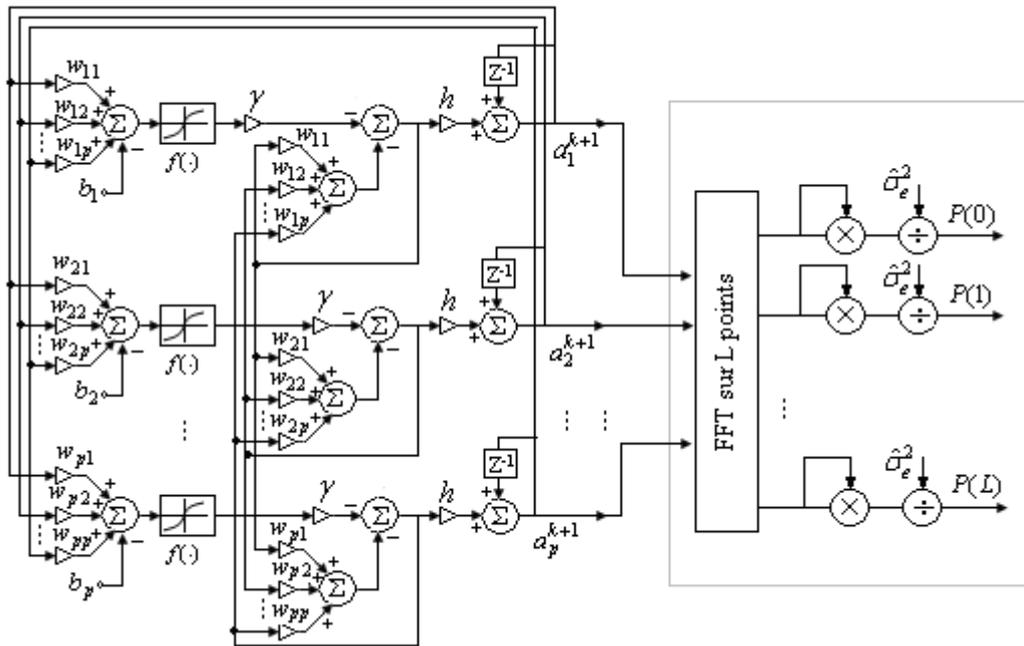


Figure 3.4. Architecture du réseau RNZ pour l'estimation spectrale

3.4.1. Fonctions d'activation

On rencontre dans littérature, une diversité de fonction d'activation, dans notre travail, on se limite aux trois fonctions les plus utilisées [79]. Soient ε_i désigne le i me élément du vecteur-erreur ε et $f_c(\cdot)$ désigne chaque élément du vecteur-fonction d'activation $\mathbf{F}_c(\cdot)$.

$$\boldsymbol{\varepsilon} = [\varepsilon_1, \varepsilon_2, \dots, \varepsilon_p]^T, \mathbf{F}_c(\cdot) = [f_c, f_c, \dots, f_c]^T \quad (3.23)$$

Fonction d'activation linéaire

$$f(\varepsilon_i) = \varepsilon_i \quad (3.24)$$

Fonction d'activation sigmoïde

$$f(\varepsilon_i) = \frac{1 - \exp(-\xi \varepsilon_i)}{1 + \exp(-\xi \varepsilon_i)} \quad (3.25)$$

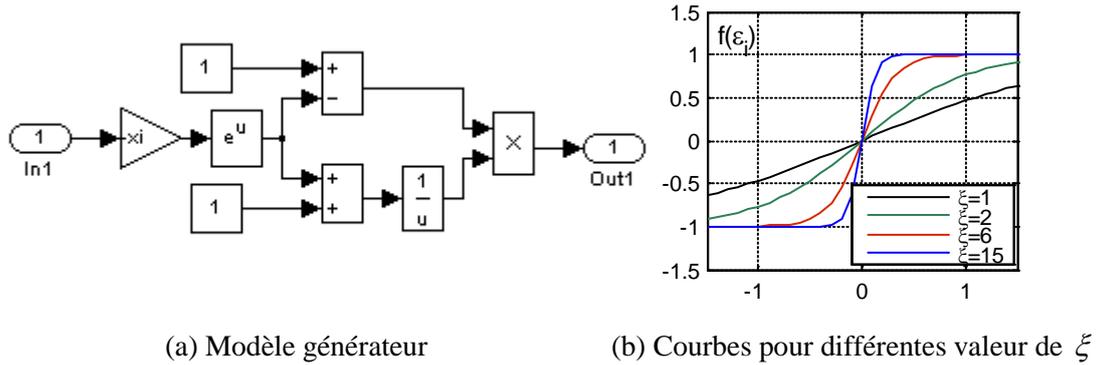


Figure 3.5. Fonction sigmoïde.

Fonction signe - puissance

$$f(\varepsilon_i) = \frac{1}{2} \left(\text{sig}^r(\varepsilon_i) + \text{sig}^{\frac{1}{r}}(\varepsilon_i) \right) \quad 0 < r < 1 \quad (3.26)$$

avec : $\text{sig}^r(\varepsilon_i) = \begin{cases} |\varepsilon_i|^r & \text{si } \varepsilon_i > 0 \\ 0 & \text{si } \varepsilon_i = 0 \\ -|\varepsilon_i|^r & \text{si } \varepsilon_i < 0 \end{cases}$

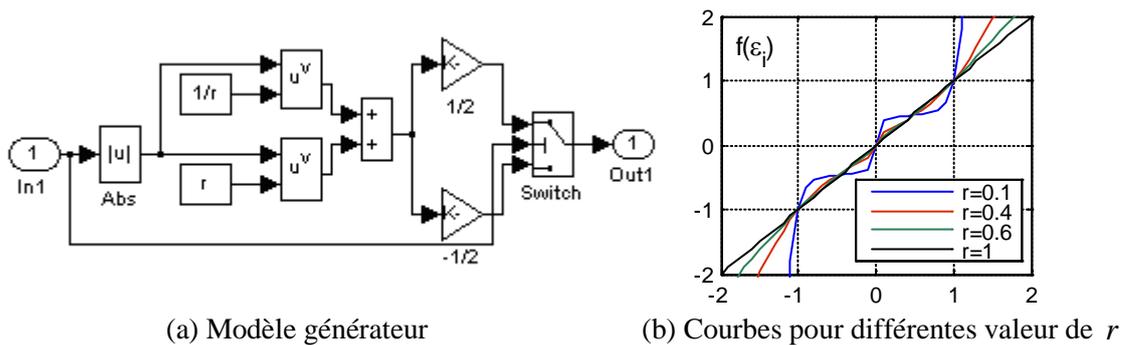


Figure 3.6. Fonction signe-puissance.

3.5. Résultats de simulation pour l'estimation des coefficients AR

Dans cette section on présente quelques exemples pour montrer les performances des réseaux de neurones proposés. Dans le premier exemple, on suppose la disponibilité $N = 64$ échantillons d'un processus autorégressif dont on connaît le modèle générateur :

$$x(n) = 2.0371x(n-1) - 2.4332x(n-2) + 1.7832x(n-3) - 0.7019x(n-4) + e(n) \quad (3.27)$$

où $e(n)$ est un processus aléatoire Gaussien, blanc de moyenne nulle et de variance σ_e^2 .

Les figures (3.7) et (3.8) montrent le schéma d'implémentation des deux réseaux sous l'environnement Simulink de MATLAB pour l'estimation des paramètres du modèle.

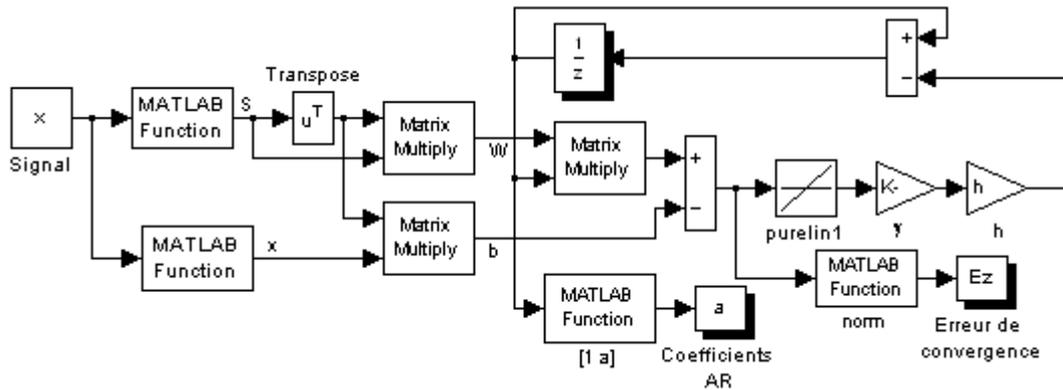


Figure 3.7. Schéma d'implémentation du modèle basé sur le gradient.

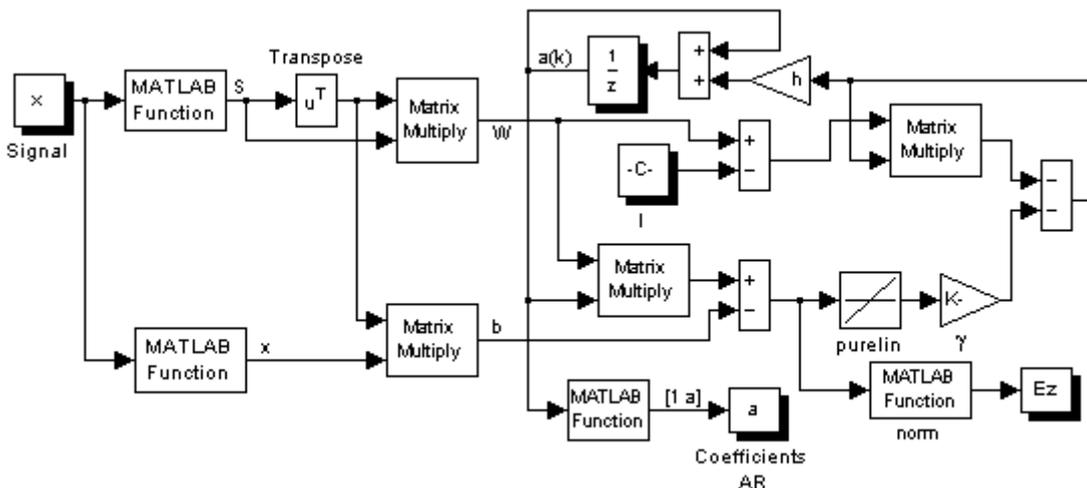


Figure 3.8. Schéma d'implémentation du modèle de Zhang.

3.5.1. Choix de la fonction d'activation

Plusieurs fonctions d'activation peuvent être utilisées, mais on se limite dans cette section à la fonction linéaire (3.24) et la fonction sigmoïde (3.25). Dans l'exemple suivant, on fixe $h = 10^{-4}$ et $\gamma = 200$. La figure (3.9) montre les trajectoires des coefficients estimés ainsi que l'erreur de convergence pour une séquence d'entrée ayant une excitation de variance $\sigma^2 = 10^{-4}$. On constate que la convergence via la fonction sigmoïde dépend du paramètre ξ ; pour des faibles valeurs, la convergence est assez lente que celle avec la fonction linéaire, mais quand ξ devient grand (la pente de la fonction d'activation devient

raide), la convergence devient rapide. Cependant, pour des valeurs critiques de ξ , les trajectoires peuvent osciller.

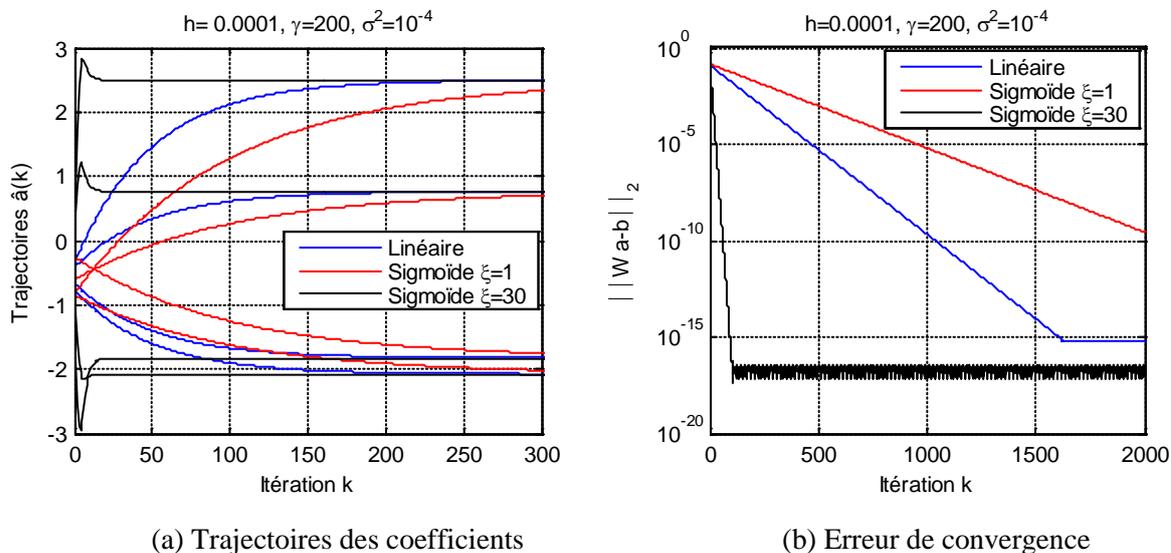


Figure 3.9. Performances du réseau RNZ pour $\sigma^2 = 10^{-4}$.

La figure (3.10) illustre le cas d'une séquence d'entrée ayant une excitation de variance $\sigma^2 = 10^{-2}$. On remarque le réseau avec une fonction linéaire est plus robuste vis-à-vis la variance de l'excitation qu'avec une fonction sigmoïde.

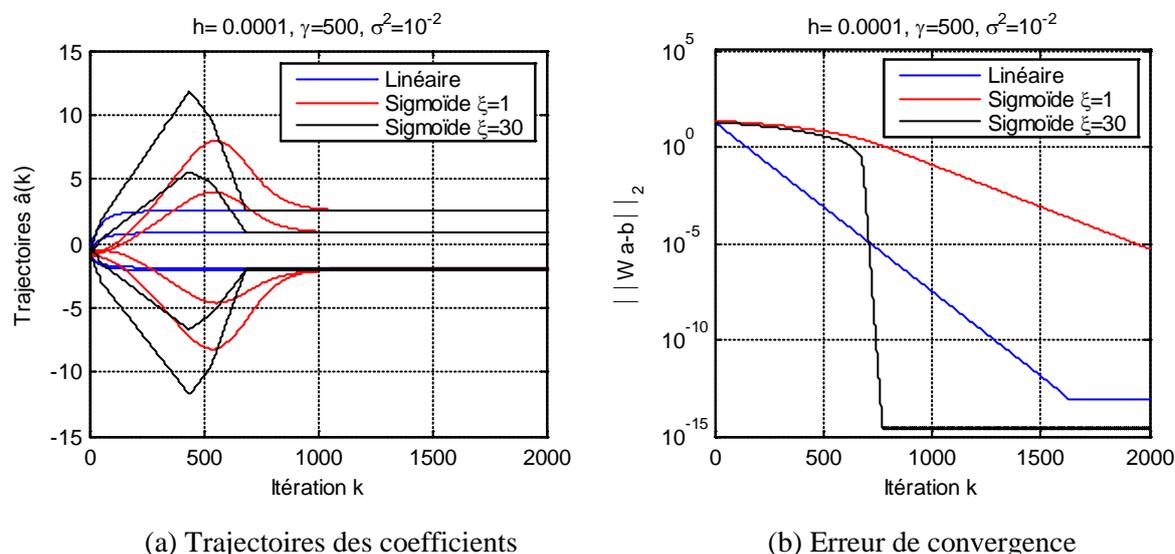


Figure 3.10. Performances du réseau RNZ pour $\sigma^2 = 10^{-2}$.

3.5.2. Choix du coefficient de convergence γ

3.5.2.1. Le réseau de Zhang

La convergence du réseau de neurones dépend fortement du paramètre γ , dans cette sous-section, on va examiner l'effet du paramètre γ sur le comportement du réseau dans les

deux cas de la fonction d'activation, linéaire et sigmoïde. Pour ce fait, on considère l'exemple ci-dessus en fixant le pas de discrétisation $h = 10^{-4}$ et $\sigma_e^2 = 10^{-1}$. Pour la fonction linéaire, la première remarque qu'on a faite, est que la convergence du réseau n'est plus garantie pour des valeurs de γ extrêmement larges.

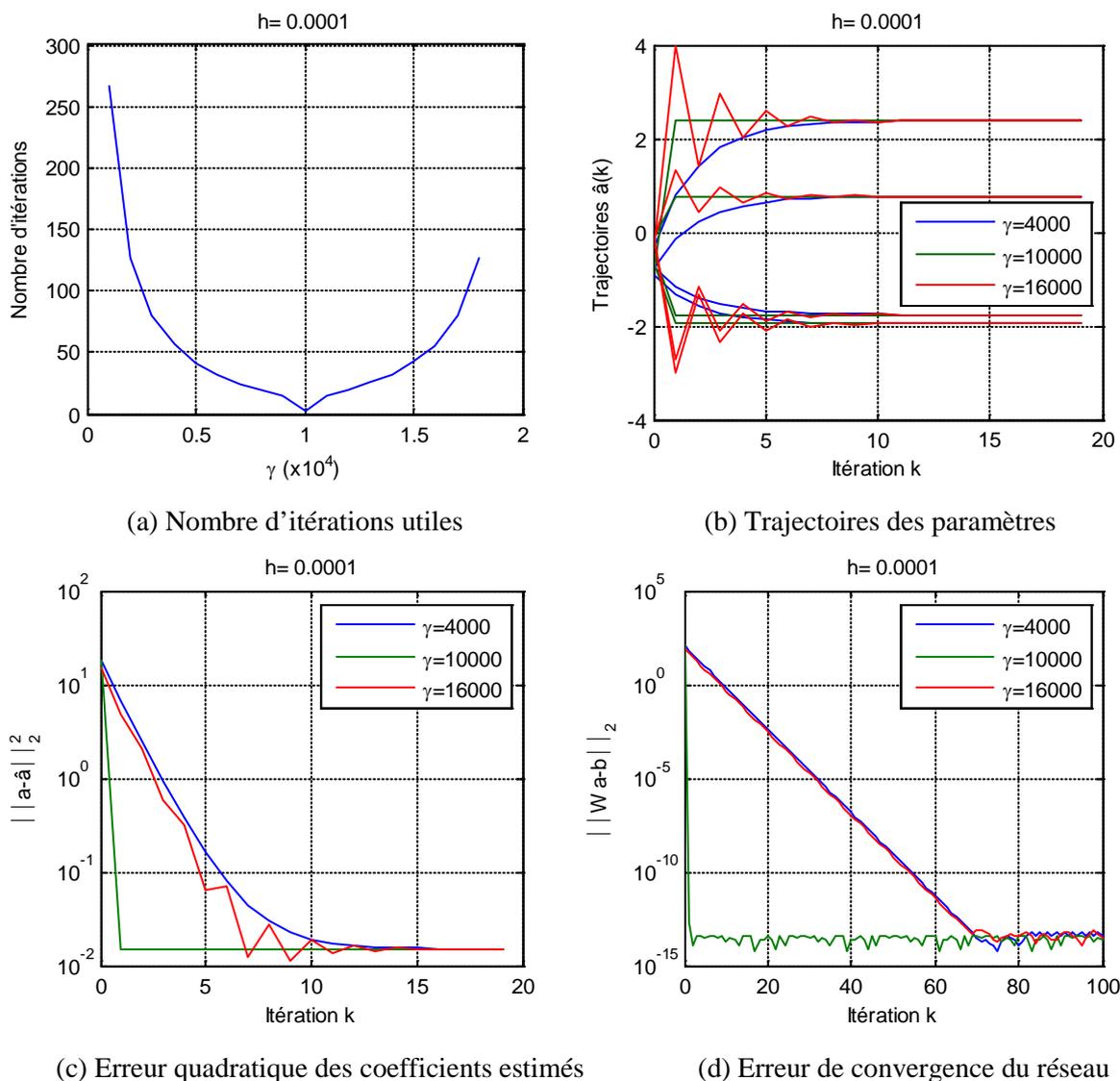


Figure 3.11. Performances du réseau de Zhang avec la fonction linéaire pour différentes valeurs de γ .

La figure (3.11.a) montre le nombre d'itérations nécessaires pour que le réseau atteigne une erreur de convergence égale à 10^{-10} , on remarque que le nombre minimal des itérations est obtenu pour une valeur $\gamma = 10^4$. Pour des valeurs supérieures ou inférieures à cette valeur optimale, le nombre d'itérations augmente. D'après toujours cette figure, on constate qu'il y a une symétrie, autour de la valeur optimale. Ceci est renforcé par la figure (3.11.d) où les courbes de convergence pour $\gamma = 4 \cdot 10^3$ et $\gamma = 16 \cdot 10^3$ sont confondues. Mais en réalité le

comportement du réseau pour ces deux valeurs est différent. Les figures (3.11.b,c) montrent les trajectoires des paramètres ainsi que leurs erreurs quadratiques pour les deux valeurs γ citées en amont. On remarque bien que la convergence est simultanée, le réseau présente des oscillations amorties pour des valeurs supérieures à la valeur optimale.

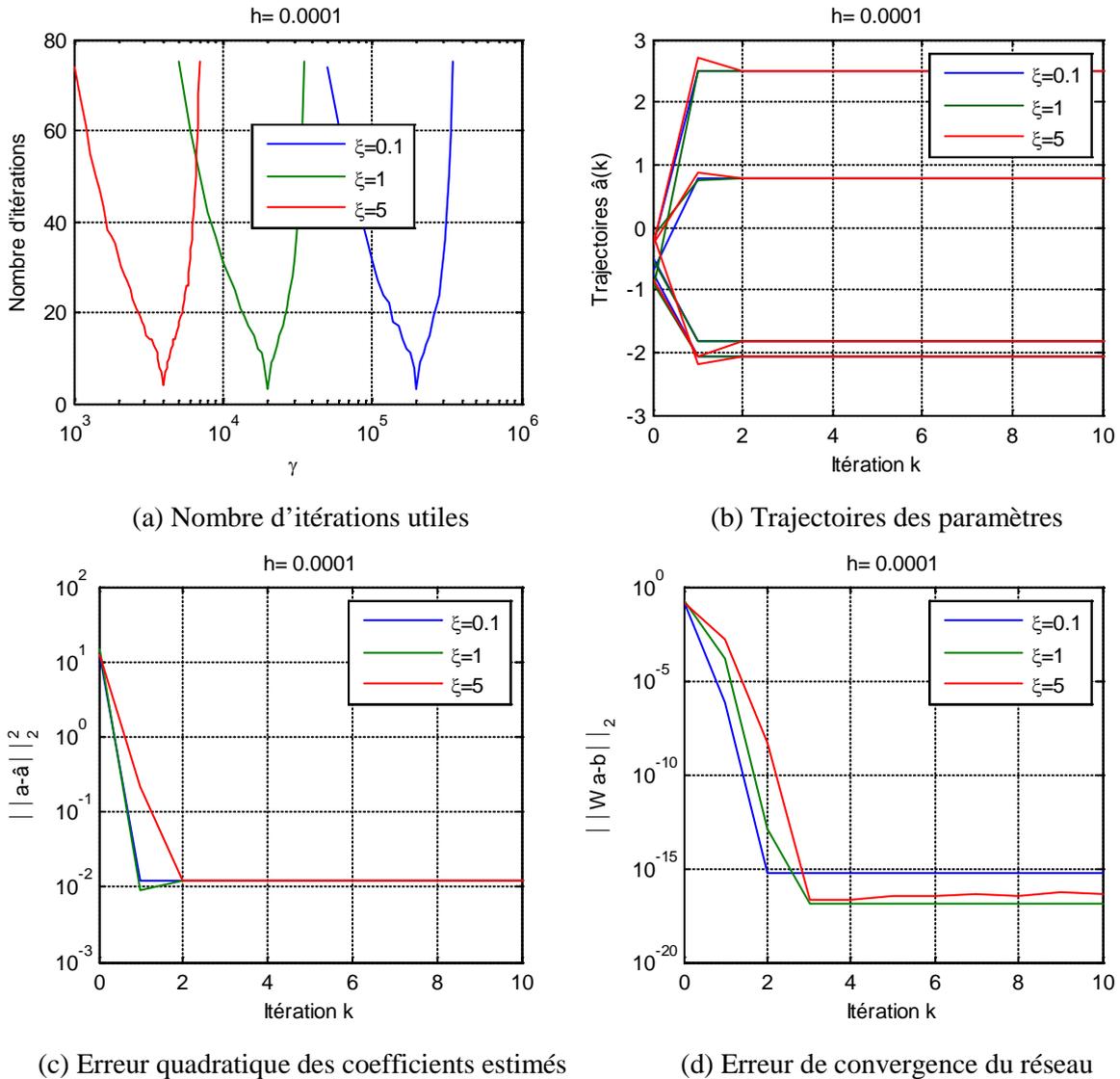


Figure 3.12. Performances du RNZ avec la fonction sigmoïde pour différentes valeurs de ξ .

Pour la fonction sigmoïde, la situation est différente. La convergence ne dépend pas seulement du paramètre γ , mais aussi du paramètre ξ . Dans la figure (3.12.a) on remarque que le nombre d'itérations nécessaires pour que le réseau atteigne une erreur de convergence égale à 10^{-10} est lié à la valeur de ξ , à titre d'exemple, pour les valeurs 0.1, 1 et 5 de ξ , le nombre minimal d'itérations est obtenu pour $\gamma = 2 \cdot 10^5$, $2 \cdot 10^4$ et $4 \cdot 10^3$. Ceci nous amène à conclure que la valeur optimale de γ est donnée empiriquement par :

$$\gamma_{opt} \approx \frac{2}{h\xi} \tag{3.28}$$

Les figures (3.12.b,c,d) montrent les trajectoires des paramètres, l'erreur quadratique et l'erreur de convergence pour les valeurs de ξ et les valeurs de γ_{opt} citées précédemment.

3.5.2.2. Le réseau basé sur le gradient

Avec la même séquence d'entrée et le même pas de discrétisation, on répète la simulation avec le réseau basé sur le gradient. Après plusieurs tentatives, on a pu trouver une plage de variation de la valeur de γ où la convergence du réseau est garantie. La valeur optimale donnant un une erreur de convergence de 10^{-10} en un minimum d'itérations (836) est égale à 160. Ceci est montré sur la figure (3.13.a).

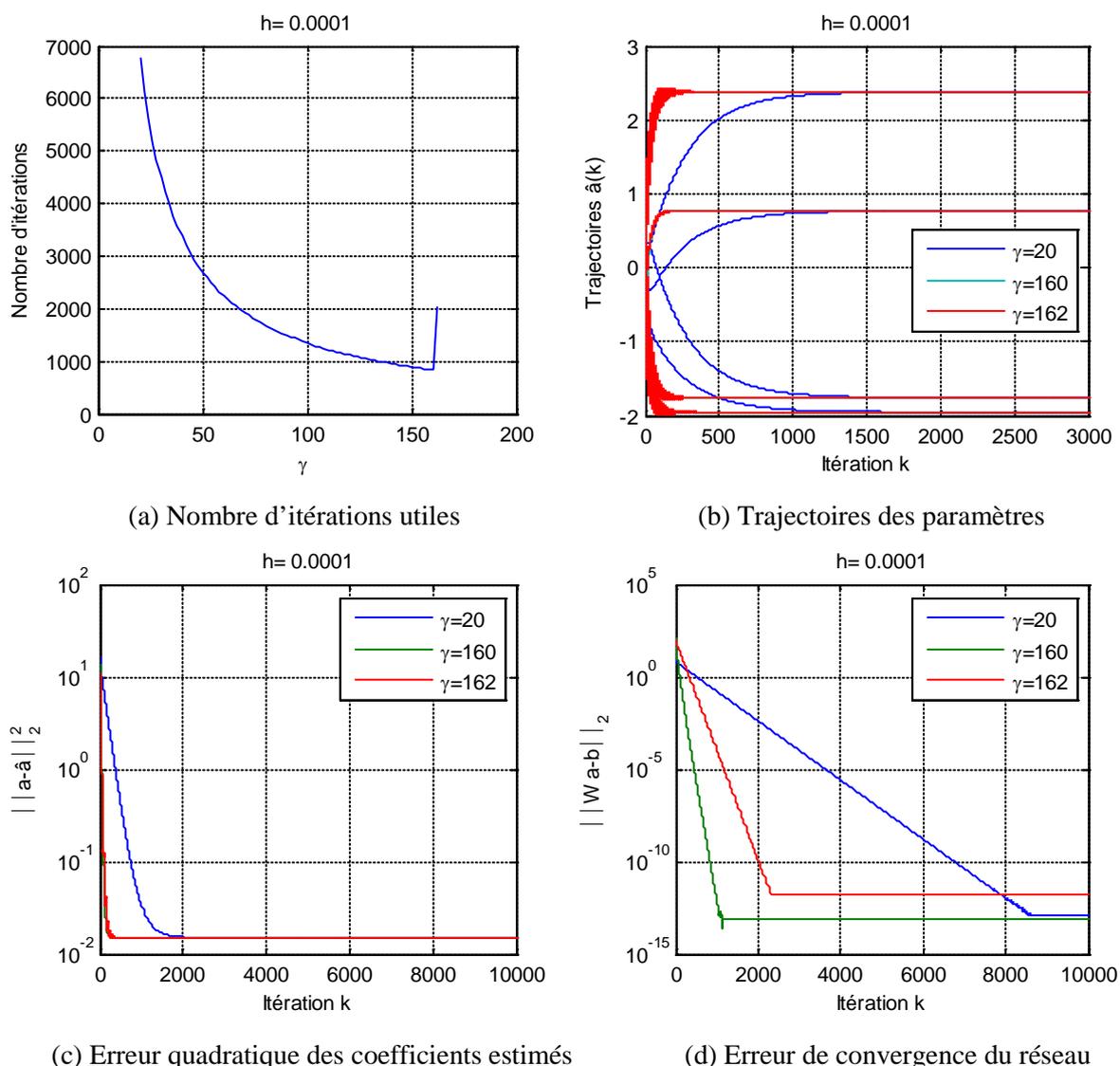


Figure 3.13. Performances du réseau basé sur le gradient suivant la valeur de γ .

Pour des valeurs plus petites, le temps de convergence est plus important. Cependant pour des valeurs justes supérieures à la valeur optimale, le réseau présente des oscillations amorties. Si on augmente encore la valeur de γ , le réseau diverge. La figure (3.13.b) montre les trajectoires des coefficients estimés pour les trois valeurs de γ indiquées. Les trajectoires pour $\gamma = 160$ sont invisibles dans la figure, elles sont juste au-dessous de celles pour $\gamma = 162$ et ne présentent aucune oscillation. Les figures (3.13.c,d) montrent l'erreur quadratique des coefficients estimés ainsi que la norme de l'erreur de convergence du réseau. Elles confirment que lorsqu'on augmente γ , le réseau s'accélère à condition de ne pas dépasser la valeur critique.

3.5.3. Choix du pas de discrétisation h

On répète le même exemple précédent mais cette fois ci on fixe le coefficient de convergence γ et on fait varier le pas de discrétisation. Les mêmes remarques discutées pour les valeurs de γ sont valables pour le pas h . Dans la figure (3.14), on montre le nombre d'itérations nécessaires pour avoir une erreur de convergence inférieure ou égale à 10^{-10} pour les deux réseaux. On constate que pour le réseau de Zhang, deux itérations sont suffisantes et sont obtenues pour $h = 25 \cdot 10^{-4}$ et $\gamma = 400$. Quant au réseau basé sur le gradient, le nombre minimal d'itérations est égal à 838. Cette performance est obtenue pour un pas $h = 1.6 \times 10^{-3}$ et $\gamma = 10$. Par une comparaison avec les résultats précédents, on conclut que le nombre minimal d'itérations est obtenu pour $\tau_{RNZ} = \gamma \cdot h \approx 1$ et $\tau_{RNG} = \gamma \cdot h \approx 0.0160$.

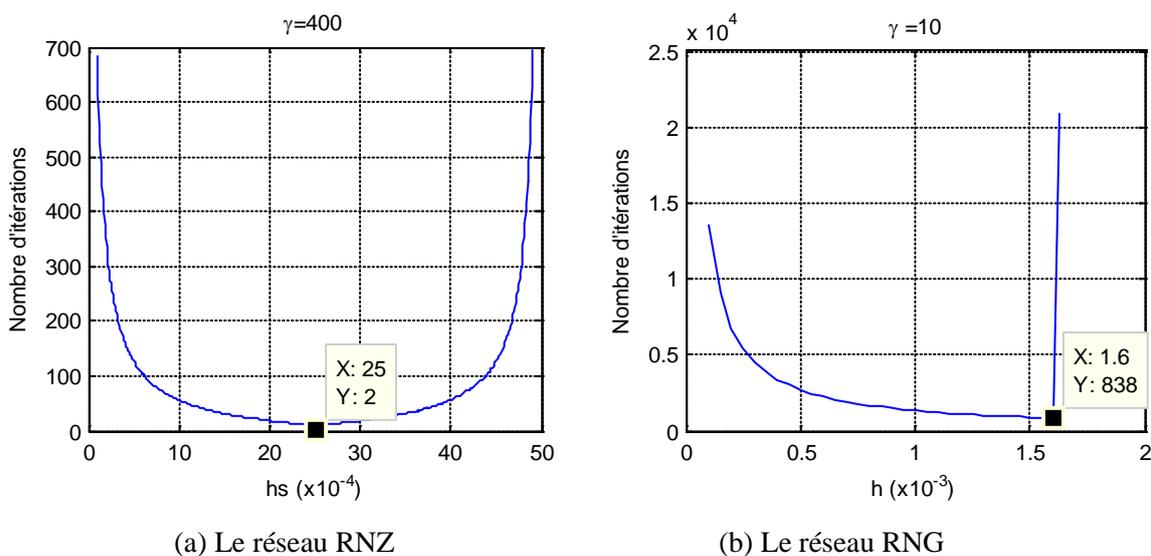


Figure 3.14. Pas de discrétisation optimale pour atteindre une erreur de convergence de 10^{-10} .

3.5.4. Effet de la variance du bruit

Pour voir l'effet de la variance σ_e^2 sur la convergence des deux réseaux, on a considéré le même exemple précédent en fixant le paramètre $h = 10^{-4}$. En variant la valeur de γ , les deux réseaux sont simulés pour des signaux de différentes valeurs de σ_e^2 . La figure (3.15.a) montre que le nombre nécessaire d'itérations pour atteindre une erreur de convergence de 10^{-10} est très peu sensible à la valeur de σ_e^2 . On constate aussi que la valeur optimale de γ donnant un minimum d'itérations vérifie l'équation $\gamma \cdot h \approx 1$ quelque soit la valeur de σ_e^2 . Les figures (3.15.b,c,d) montrent les performances du réseau pour $\gamma_{opt} = 10^4$. On remarque bien que le réseau de Zhang est peu sensible à la valeur de σ_e^2 ; le temps de convergence est similaire pour toutes les valeurs de la variance.

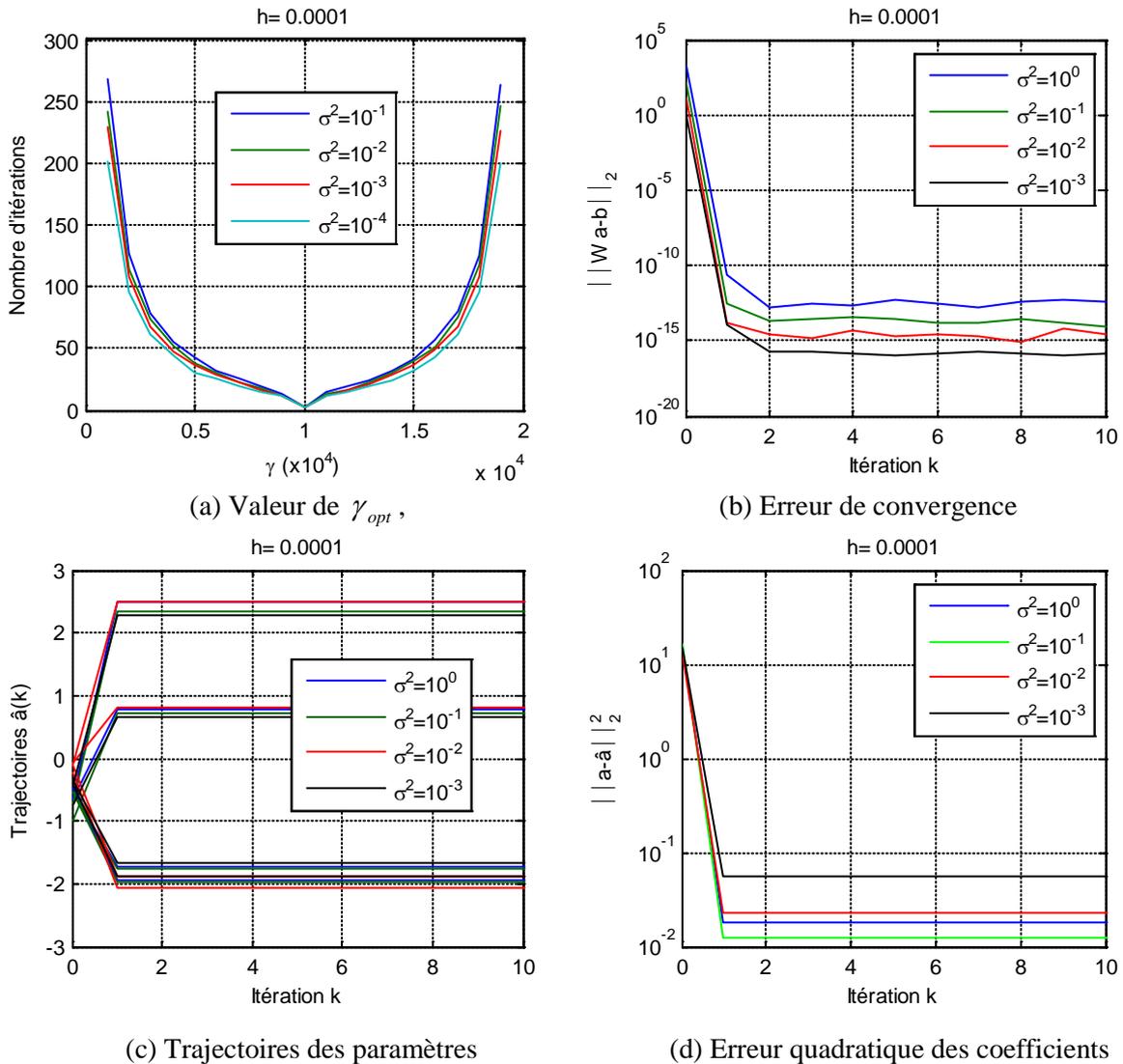


Figure 3.15. Performances du réseau de Zhang suivant la valeur de σ_e^2 .

Pour le réseau RNG, la situation n'est plus similaire, le réseau est fortement influencé par la valeur de σ_e^2 . La figure (3.16.a) montre la valeur optimale de γ pour avoir un minimum d'itérations k_{\min} pour atteindre une erreur de convergence de 10^{-10} suivant la valeur de σ_e^2 , on constate que la valeur γ_{opt} est très liée à la variance du bruit. Le tableau ci-dessous donne quelques exemples illustratifs. Les figures (3.16.b,c,d) montrent les performances du réseau pour une valeur de γ égale à 10. On remarque que plus la variance du signal d'excitation est grande, plus la convergence est rapide.

σ_e^2	1	0.1	0.01	0.001	0.0001
γ_{opt}	13,56	167	1320	21500	180000
k_{\min}	1115	956	1033	733	881

Tableau 3.1. Les valeurs de k_{\min} correspondant aux γ_{opt} pour différentes valeurs de σ_e^2

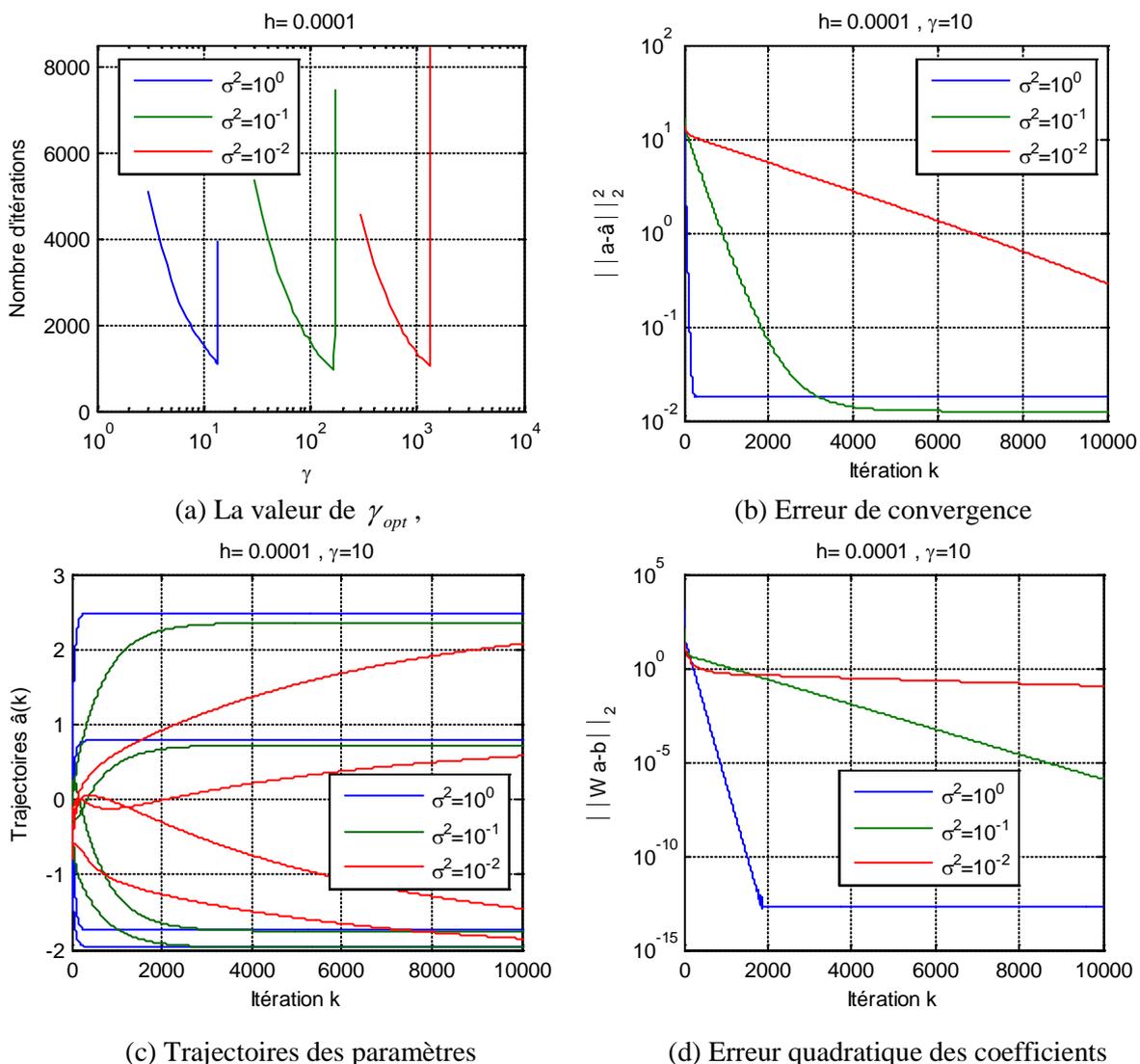


Figure 3.16. Performances du réseau RNG suivant la valeur de σ_e^2 .

3.5.5. Comparaison avec d'autres méthodes

Pour mieux juger les performances du réseau de neurones, on compare les paramètres estimés par ce réseau avec ceux obtenus avec d'autres méthodes conventionnelles. Le tableau ci-dessous donne un résumé des résultats obtenu par 100 essais avec les différentes méthodes. On remarque que les deux réseaux de neurones se confondent avec la méthode des moindres carrés. On constate aussi que le réseau de Zhang est très rapide devant le réseau basé sur le gradient.

		a_1	a_2	a_3	a_4	$\ a - \hat{a}\ _2^2$	$\ Wa - b\ _2$	k_{conv}	
σ^2	Paramètres réels	-2.037	2.433	-1.783	0.701	0	-	-	
	Moindres carrés	-2.0115	2.3806	-1.7283	0.6721	0.0073	-	-	
10^{-4}	Burg	-2.0058	2.3581	-1.6972	0.6550	0.0162	-	-	
	Yule-Walker	-1.7568	1.9212	-1.2837	0.4495	0.6540	-	-	
	RNG $\gamma = 50$	-2.0115	2.3806	-1.7283	0.6721	0.0073	5.6802e-017	10270	
	RNZ $\gamma = 10^4$	Linéaire	-2.0115	2.3806	-1.7283	0.6721	0.0073	2.4957e-017	2
		Sigmoïde $\xi = 2$	-2.0115	2.3806	-1.7283	0.6721	0.0073	4.8542e-017	3
	10^{-1}	Moindres carrés	-2.0203	2.3919	-1.7247	0.6714	0.0063	-	-
Burg		-2.0155	2.3710	-1.6968	0.6554	0.0140	-	-	
Yule-Walker		-1.7522	1.8988	-1.2501	0.4383	0.7205	-	-	
RNG $\gamma = 50$		-2.0203	2.3919	-1.7247	0.6714	0.0063	5.4000e-014	13240	
RNZ $\gamma = 10^4$		Linéaire	-2.0203	2.3919	-1.7247	0.6714	0.0063	2.6047e-014	2
		Sigmoïde $\xi = 2$	-2.0203	2.3919	-1.7247	0.6714	0.0063	2.5371e-014	230

Tableau 3.2. Résumé des résultats obtenus avec les réseaux proposés et autres méthodes.

3.6. Estimation de la densité spectrale

Dans la deuxième partie de la présente simulation, on s'intéresse à la comparaison des deux réseaux avec les méthodes conventionnelles pour l'estimation spectrale. Pour ce fait on commence par comparer les deux spectres obtenus avec les deux réseaux, puis on compare le meilleur des deux avec les méthodes conventionnelles. Le schéma d'implémentation utilisé est représenté dans la figure (3.17) où Le sous-système contient l'un des deux réseaux représentés dans les figures (3.7) et (3.8). La fonction [1 a] sert à ajouter le $a(0) = 1$ au vecteur \mathbf{a} . A la sortie out1, on récupère la densité spectrale du signal via le réseau de neurone.

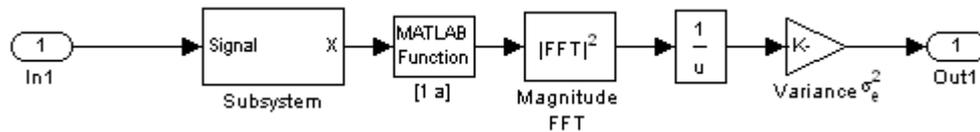


Figure 3.17. Implémentation du réseau de Zhang pour l'estimation spectrale.

La figure (3.18) montre le schéma de simulation pour comparer les deux réseaux de neurones pour estimer la densité spectrale d'un processus autorégressif.

Le signal utilisé est issu d'un filtre tout pôles dont les coefficients sont égaux à ceux utilisés dans l'exemple précédent et dont l'entrée est un bruit gaussien de moyenne nulle et de variance $\sigma_e^2 = 10^{-2}$.

Les paramètres suivants ont été utilisés :

- L'ordre du modèle a été fixé à 4.
- Le nombre d'échantillons $N = 64$ par segment.
- La FFT est calculée sur 256 points.
- Les coefficients de convergence des deux réseaux sont fixés à $\tau_{RNZ} = 0.2$, $\tau_{RNG} = 0.1$
- La fonction d'activation linéaire a été sélectionnée.

On note ici que dans la partie précédente, les deux réseaux itèrent sur le même signal d'entrée, par contre dans la présente simulation, à chaque itération, un nouveau segment de 64 échantillons est disponible. Dans ce cas l'état présent du réseau constitue un état initial pour le segment suivant dans l'itération suivante.

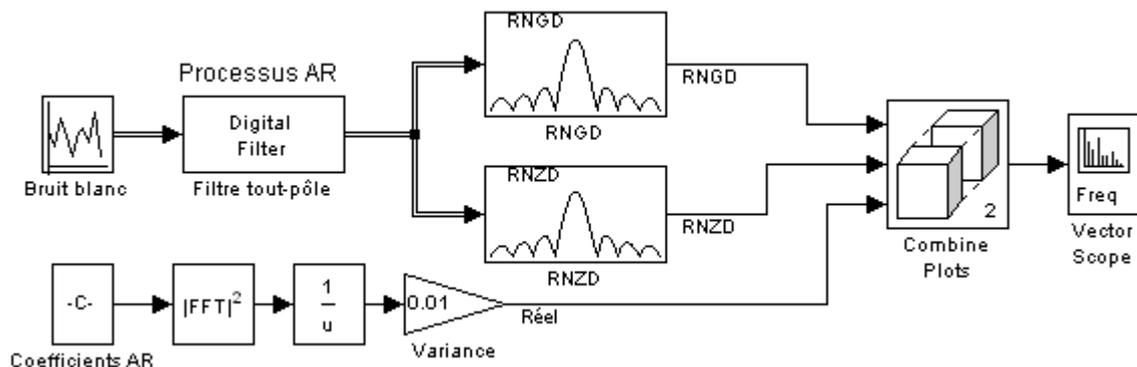


Figure 3.18. Schéma de simulation pour comparer les performances des deux réseaux.

Les figures (3.19) et (3.20) montrent les résultats obtenus sur un temps simulation de 0.02s qui est équivalent à 313 segments. La figure (3.19.a) présente l'erreur de convergence des deux réseaux. On remarque bien que le RNZ converge juste après quelques segments, par

contre le RNG nécessite une centaine de segments pour la convergence. Ceci est montré encore par la figure (3.19.b).

La figure (3.20.a) montre la densité spectrale estimée après 25 segments. On constate que le RNZ estime parfaitement la densité spectrale réelle, par contre la densité estimée par le RNG est totalement erronée. Au fur et à mesure les segments arrivent, la densité estimée par le RNG s'améliore avec une nette supériorité du RNZ, ceci est prouvé par la figure (3.20.b).

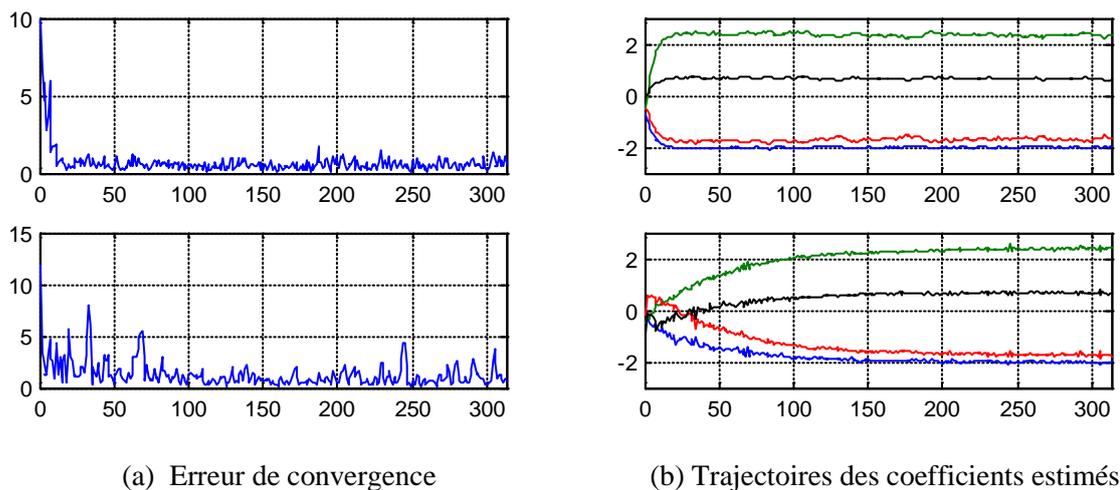
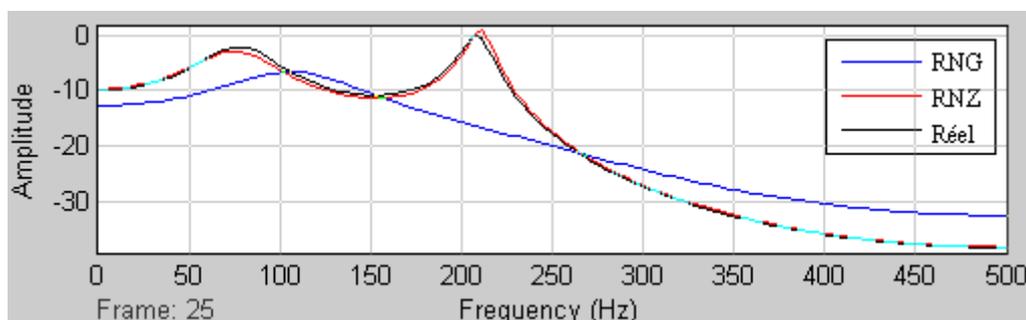
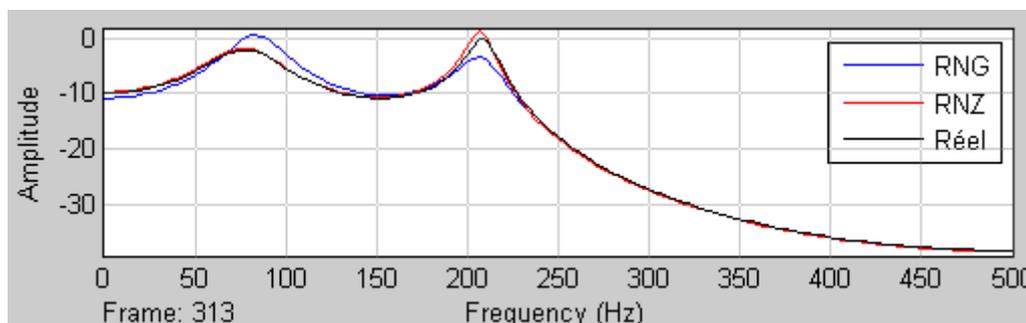


Figure 3.19. Performances des deux réseaux.



(a) Après 25 segments



(b) Après 313 segments

Figure 3.20. Densité spectrale estimée par les deux réseaux.

Puisque le RNZ offre une supériorité par rapport au RNG du point de vue rapidité et précision, on le garde pour une comparaison avec la méthode de Burg et celle de la covariance modifiée. La figure (3.21) montre le schéma de simulation utilisé.

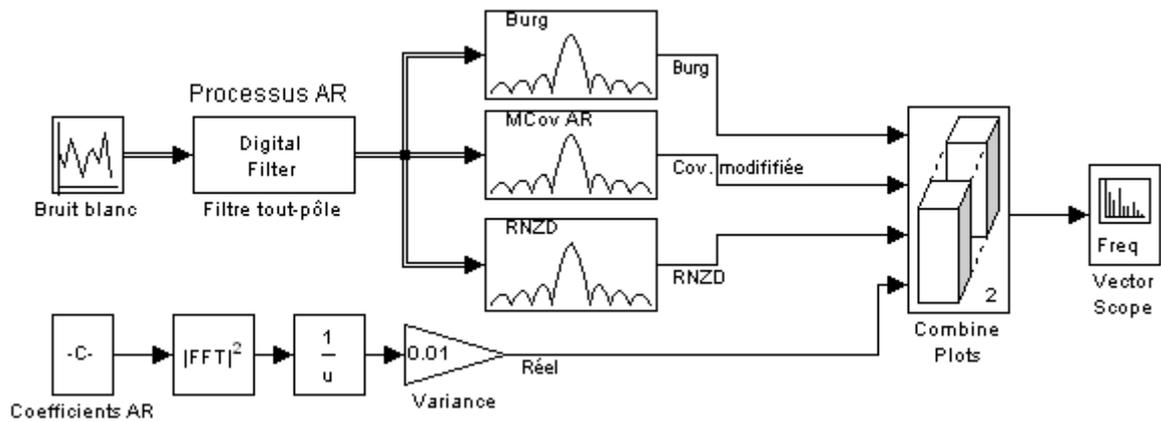


Figure 3.21. Implémentation du réseau de Zhang pour l'estimation spectrale.

Comme montré dans la figure (3.22.b), les trajectoires des coefficients obtenus par le RNZ ont une variation douce par rapport à ceux obtenus par la méthode de Burg ou par la méthode de la covariance modifiée. Le tableau (3) illustre la moyenne et l'écart type des coefficients estimés. On note ici que la moyenne et l'écart type des coefficients obtenus par le RNZD ont été calculés à partir du 10me segment.

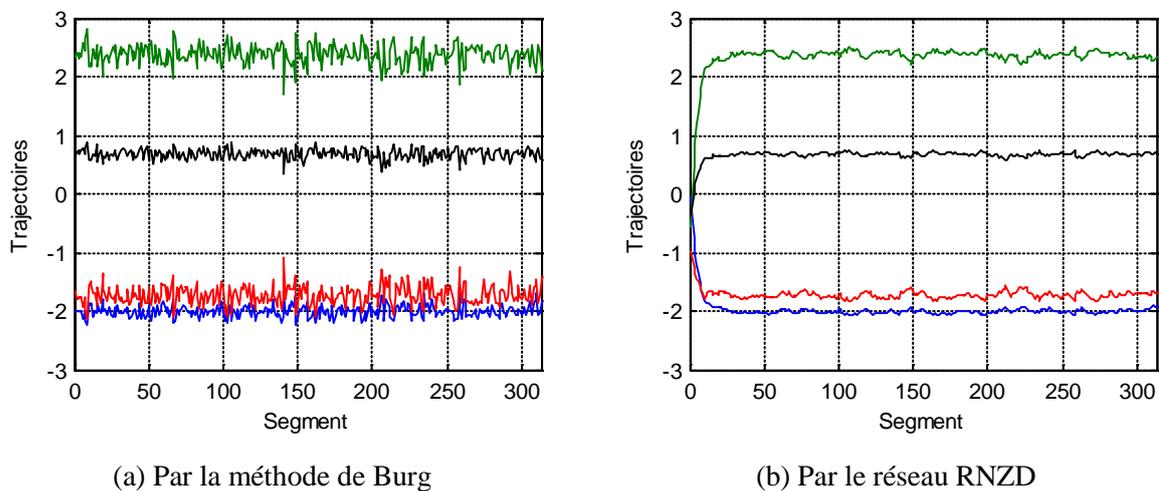


Figure 3.22. Comparaison des trajectoires des coefficients obtenus avec la méthode de Burg.

Pour la densité spectrale estimée, on remarque sur la figure (3.23) que la densité estimée par le réseau de Zhang est plus proche de la vraie densité que les deux autres méthodes. Cette remarque ne doit pas être généralisée, parce que la densité estimée est très liée au segment considéré.

		a_1	a_2	a_3	a_4
Paramètres réels		-2.0371	2.4332	-1.7832	0.7019
Moyenne	RNZ	-2.0045	2.3725	-1.7276	0.6750
	Burg	-2.0062	2.3744	-1.7262	0.6773
	Cov. modifiée	-2.0091	2.3790	-1.7292	0.6780
Ecart- type	RNZ	0.0393	0.0613	0.0520	0.0297
	Burg	0.1026	0.1701	0.1612	0.0910
	Cov. modifiée	0.0992	0.1652	0.1601	0.0910

Tableau 3.3. Propriétés statistiques des coefficients estimés.

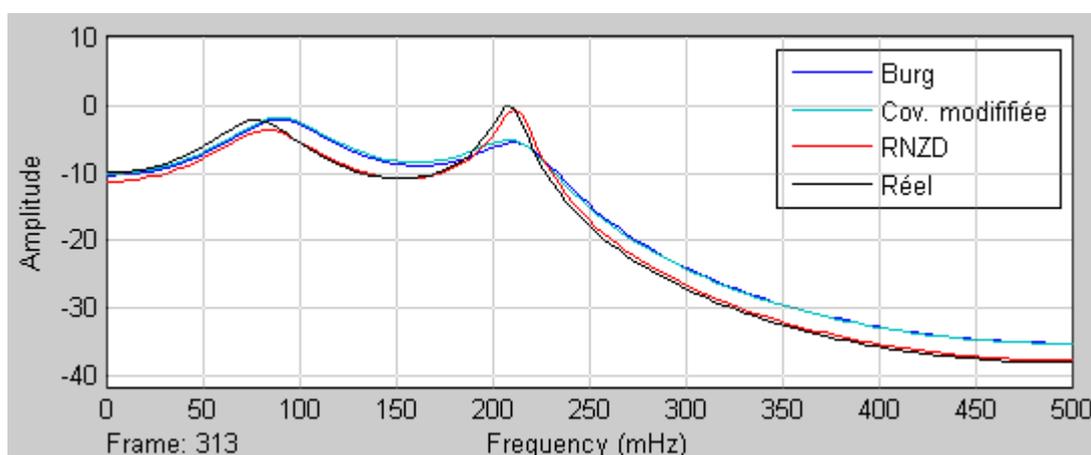


Figure 3.23. Densité spectrale estimée par les différentes méthodes après 313 segments.

3.7. Conclusion

Dans ce chapitre nous avons présenté deux types de réseaux récurrents pour l'estimation des coefficients d'un modèle autorégressif. Le premier est basé sur la minimisation de la norme d'une erreur utilisant le gradient, le suivant est basé sur la minimisation d'un vecteur-erreur élément par élément [44]. D'après les résultats des simulations, le deuxième réseau prouve une meilleure performance du point de vue convergence par rapport au premier. On a trouvé qu'une ou deux itérations sont suffisantes pour la convergence du réseau de Zhang dans le cas où les paramètres optimaux du réseau sont choisis.

Un autre avantage du réseau de Zhang et qu'on le considère le plus intéressant et sa capacité de maintenir ses bonnes performances dans les applications en temps réel. On a montré que lorsque le modèle neuronal de Zhang est excité par une nouvelle séquence à chaque itération, quelques itérations sont suffisantes dans le régime transitoire pour que les coefficients estimés atteignent les coefficients vrais avec une certaine tolérance.

Chapitre 4

Estimateur neuronal par minimum de variance

4.1. Introduction

La méthode d'estimation spectrale par minimum de variance est l'une des meilleures méthodes dites à haute résolution, dans le sens qu'elles atteignent de meilleures performances pour l'estimation d'une fréquence que les méthodes basées sur la transformée de Fourier [59].

L'estimateur spectral basé sur le minimum de variance nécessite une inversion de la matrice de corrélation des données et la multiplication de la matrice inversée par le vecteur de Fourier et de son conjugué. Les algorithmes les plus rapides utilisés pour évaluer cet estimateur sont l'algorithme de Musicus [12] et l'algorithme de Marple *et al.* [20]. Cependant ces deux algorithmes nécessitent l'estimation des paramètres AR dont la complexité du calcul est considérable dans le cas où l'ordre du modèle est grand. Cette tâche sera allégée par l'utilisation du réseau de Zhang en résolvant les équations de Yule-Walker [44].

4.2. Estimateur spectral par minimum de variance

4.2.1. Principe

Soit un ensemble de N échantillons d'une séquence aléatoire $x(n)$, cette séquence passe à travers un filtre à réponse impulsionnelle finie d'ordre p . La sortie de ce filtre est de la forme :

$$y(n) = \sum_{k=0}^p x(n-k)h_p(k) = \mathbf{x}_p^T(n)\mathbf{h}_p \quad (4.1)$$

où $\mathbf{x}_p^T(n) = [x(n) x(n-1) \cdots x(n-p)]$ est un vecteur qui contient les $p+1$ derniers échantillons, et $\mathbf{h}_p^T = [h_p(0) h_p(1) \cdots h_p(p)]$ est un vecteur d'ordre $p+1$ qui contient les coefficients du filtre.

La puissance à la sortie du filtre est donnée par [59] :

$$\begin{aligned}\rho_p &= E\{|y(n)|^2\} = E\{|\mathbf{x}_p^T(n)\mathbf{h}_p|^2\} = \mathbf{h}_p^H E\{\mathbf{x}_p^*(n)\mathbf{x}_p^T(n)\}\mathbf{h}_p \\ &= \mathbf{h}_p^H \mathbf{R}_p \mathbf{h}_p\end{aligned}\quad (4.2)$$

où

$$\mathbf{R}_p = \begin{bmatrix} r(0) & r^*(1) & \cdots & r^*(p) \\ r(1) & r(0) & \cdots & r^*(p-1) \\ \vdots & \vdots & \ddots & \vdots \\ r(p) & r(p-1) & \cdots & r(0) \end{bmatrix}$$

est la matrice d'autocorrélation dont les éléments ne sont autres que l'autocorrélation de la séquence $\mathbf{x}_p(n)$

$$r(m) = E\{x_p(n)x_p^*(n+m)\} \quad (4.3)$$

L'estimateur spectral est conçu de façon à minimiser la variance de la sortie du filtre sous la contrainte que le gain du filtre à la fréquence f_0 est égal à 1 c.-à-d.:

$$\mathbf{a}_p^H(f_0)\mathbf{h}_p = 1 \quad (4.4)$$

avec :

$$\mathbf{a}_p^H(f_0) = [1 \quad e^{-j2\pi f_0} \quad \cdots \quad e^{-j2\pi f_0 p}] \quad (4.5)$$

Pour résoudre ce problème, on utilise la méthode de Lagrange :

$$J = \mathbf{h}_p^H \mathbf{R}_p \mathbf{h}_p + \lambda (1 - \mathbf{a}_p^H(f_0)\mathbf{h}_p) \quad (4.6)$$

Le problème revient à chercher les coefficients du \mathbf{h}_p qui minimise la fonction J . En dérivant l'expression (4.6) par rapport à \mathbf{h}_p , on trouve :

$$\frac{\partial J}{\partial \mathbf{h}_p} = 2\mathbf{R}\mathbf{h}_p - \lambda \mathbf{a}_p^H(f_0) \quad (4.7)$$

En mettant l'expression (4.7) égale à zéro, on trouve :

$$\frac{\partial J}{\partial \mathbf{h}_p} = 0 \Rightarrow \mathbf{h}_p = \frac{\lambda}{2} \mathbf{R}^{-1} \mathbf{a}_p(f_0) \quad (4.8)$$

En remplaçant l'équation (4.8) dans la contrainte $\mathbf{a}_p^H(f_0)\mathbf{h}_p = 1$, on obtient :

$$\frac{\lambda}{2} = \frac{1}{\mathbf{a}_p^H(f_0)\mathbf{R}^{-1}\mathbf{a}_p(f_0)} \quad (4.9)$$

En remplaçant l'expression (4.9) dans (4.8), l'expression du filtre devient :

$$\mathbf{h}_{MV}(f) = \frac{\mathbf{R}^{-1}\mathbf{a}(f_0)}{\mathbf{a}_p^H(f_0)\mathbf{R}^{-1}\mathbf{a}_p(f_0)} \quad (4.10)$$

La variance à la sortie du filtre est déduite de (4.10) et (4.2):

$$\rho_p = \frac{1}{\mathbf{a}_p^H(f_0)\mathbf{R}^{-1}\mathbf{a}_p(f_0)} \quad (4.11)$$

Comme la variance exprime la puissance à la sortie du filtre, en multipliant la variance par le pas d'échantillonnage qui est égale à 1s, on obtient une quantité en puissance par Hertz ou tout simplement densité spectrale de puissance.

L'évaluation de la densité spectrale de puissance dans l'intervalle de Nyquist $f \in [-1/2 \ 1/2]$ définit l'estimateur spectral par minimum de variance basé sur l'autocorrélation de la séquence $\mathbf{x}_p(n)$

$$\rho_p(f) = \frac{1}{\mathbf{a}_p^H(f)\mathbf{R}^{-1}\mathbf{a}_p(f)} \quad (4.12)$$

4.2.2. Algorithme de Musicus

Dans [12], Musicus a développé un algorithme rapide basé sur la structure spéciale de la matrice \mathbf{R} . Si on définit R_{ij} le ij ème élément de la matrice \mathbf{R}^{-1} , l'expression du dénominateur qu'on le note D de l'équation (4.12) aura la forme :

$$\begin{aligned} D &= \begin{bmatrix} 1 & e^{j2\pi ft} & \dots & e^{j2\pi fpt} \end{bmatrix} \begin{bmatrix} R_{11}^{-1} & R_{12}^{-1} & \dots & R_{1p+1}^{-1} \\ R_{21}^{-1} & R_{22}^{-1} & \dots & R_{2p+1}^{-1} \\ \vdots & \vdots & \ddots & \vdots \\ R_{p+11}^{-1} & R_{p+12}^{-1} & \dots & R_{p+1p+1}^{-1} \end{bmatrix} \begin{bmatrix} 1 \\ e^{-j2\pi ft} \\ \vdots \\ e^{-j2\pi fpt} \end{bmatrix} \\ &= \left[R_{11}^{-1} + R_{21}^{-1}e^{j2\pi ft} + \dots + R_{p+11}^{-1}e^{j2\pi fpt} \right] + \left[R_{12}^{-1}e^{-j2\pi ft} + R_{22}^{-1} + \dots + R_{p+12}^{-1}e^{j2\pi f(p-1)t} \right] + \dots \\ &\quad + \left[R_{1p+1}^{-1}e^{-j2\pi fpt} + R_{2p+1}^{-1}e^{-j2\pi f(p-1)t} + \dots + R_{p+1p+1}^{-1} \right] \end{aligned}$$

Cette dernière expression peut être généralisée sous la forme :

$$D = \sum_{t=-(p+1)}^{p+1} \varphi(t) e^{-j2\pi ft} \quad (4.13)$$

où:

$$\varphi(t) = \sum_{l=\max(0,-t)}^{\min(N-t,p+1)} R_{l,l+t}^{-1} \quad (4.14)$$

Puisque la matrice \mathbf{R} est Toeplitz, la formule de Gohberg-Semencul affirme que cette dernière peut être exprimée sous la forme d'une différence de deux produits de matrices triangulaires et Toeplitz [84] :

$$\mathbf{R}^{-1} = \frac{1}{\rho_p^{AR}} \mathbf{A}_p \mathbf{A}_p^H - \frac{1}{\rho_p^{AR}} \mathbf{B}_p \mathbf{B}_p^H \quad (4.15)$$

où ρ_p^{AR} est donnée par l'expression (1.16) et les matrices:

$$\mathbf{A}_p = \begin{bmatrix} 1 & 0 & \dots & 0 \\ a_p(1) & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_p(p) & a_p(p-1) & \dots & 1 \end{bmatrix}, \mathbf{B}_p = \begin{bmatrix} 0 & 0 & \dots & 0 \\ a_p^*(p) & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_p^*(1) & a_p^*(2) & \dots & 0 \end{bmatrix} \quad (4.16)$$

sont constituées des paramètres AR, $a_p(m)$ pour $1 \leq m \leq p$, obtenus par la solution des équations de Yule-Walker :

$$R_p \cdot \begin{bmatrix} 1 \\ a_p(1) \\ \vdots \\ a_p(p) \end{bmatrix} = \begin{bmatrix} \rho_p^{AR} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4.17)$$

Le développement de l'expression (4.15) nous permet de formuler l'expression du (l, k) ^{ème} élément de la matrice \mathbf{R}^{-1} comme :

$$R_{l,t}^{-1} = \frac{1}{\rho_p^{AR}} \sum_{m=0}^l a_p(m) \cdot a_p^*(m+t-l) - a_p^*(p+1-m) \cdot a_p(p+1-m-t+l), \quad t \geq l \quad (4.18)$$

$$\varphi(t) = \sum_{l=\max(0,-t)}^{\min(N-t,p+1)} \frac{1}{\rho_p^{AR}} \sum_{m=0}^l a_p(m) \cdot a_p^*(m+t) - a_p^*(p+1-m) \cdot a_p(p+1-m-t) \quad (4.19)$$

Si on s'intéresse au cas $t \geq 0$, l'expression devient :

$$\begin{aligned} \varphi(t) &= \sum_{l=0}^{p-t} \frac{1}{\rho_p^{AR}} \sum_{m=0}^l a_p(m) \cdot a_p^*(m+t) - a_p^*(p+1-m) \cdot a_p(p+1-m-t) \\ &= \frac{1}{\rho_p^{AR}} \sum_{m=0}^{p-t} (p+1-t-2m) a_p(m) a_p^*(m+t), \quad k = 0, 1, \dots, N \end{aligned} \quad (4.20)$$

Pour résumer, l'algorithme de Musicus consiste à estimer la densité spectrale en trois étapes :

- Calculer les coefficients AR par l'algorithme de Levinson
- Calculer les coefficients $\varphi(t)$ pour $p \geq t \geq 0$ et déduire les coefficients $\varphi(t) = \varphi^*(-t)$ pour $-p \leq t \leq -1$.

- Calculer la densité spectrale en évaluant l'expression (4.13) par la transformée de Fourier rapide.

4.2.3. Algorithme de Marple

Bien que la transformée de Fourier rapide utilisé dans l'algorithme de Musicus a réduit la complexité de calcul, le calcul des coefficients de la corrélation des paramètres AR reste considérable, S. Lawrence Marple Jr. [20] a découvert une structure additionnelle qui peut réduire encore le calcul par l'utilisation uniquement de la transformée de Fourier rapide et évitant le calcul des coefficients de la corrélation des paramètres AR. Cet algorithme est classé dans la catégorie des algorithmes ultra rapides.

D'après [20], l'expression de la densité spectrale donnée par l'équation (4.12) peut être réécrite sous la forme :

$$\rho_p(f) = \frac{T}{(p+1)A(f)A^*(f) - A(f)B^*(f) - A^*(f)B(f)} \quad (4.21)$$

où :

$$A(f) = \sum_{n=0}^p a_p(n) e^{-j2\pi f T n} \quad (4.21)$$

$$B(f) = \sum_{n=0}^p n a_p(n) e^{-j2\pi f T n} \quad (4.22)$$

On note ici que $A(f)$ et $B(f)$ ne sont autres que la transformée de Fourier des coefficients AR. Pour démontrer l'expression (4.21), il suffit de développer le dénominateur pour aboutir au dénominateur de l'expression (4.13)

$$D = (p+1)A(f)A^*(f) - A(f)B^*(f) - A^*(f)B(f)$$

$$A(f)A^*(f) = \sum_{n=0}^p a_p(n) e^{-j2\pi f T n} \sum_{m=0}^p a_p^*(m) e^{+j2\pi f T m} = \sum_{n=0}^p \sum_{m=0}^p a_p(n) a_p^*(m) e^{-j2\pi f T (n-m)}$$

$$A(f)B^*(f) = \sum_{n=0}^p a_p(n) e^{-j2\pi f T n} \sum_{m=0}^p m a_p^*(m) e^{+j2\pi f T m} = \sum_{n=0}^p \sum_{m=0}^p m a_p(n) a_p^*(m) e^{-j2\pi f T (n-m)}$$

$$A^*(f)B(f) = \sum_{n=0}^p a_p^*(n) e^{+j2\pi f T n} \sum_{m=0}^p m a_p(m) e^{-j2\pi f T m} = \sum_{n=0}^p \sum_{m=0}^p m a_p(n) a_p(m) e^{-j2\pi f T (m-n)}$$

$$D = \sum_{n=0}^p \sum_{m=0}^p (p+1) a_p(n) a_p^*(m) e^{-j2\pi f T (n-m)} - \sum_{n=0}^p \sum_{m=0}^p m a_p(n) a_p^*(m) e^{-j2\pi f T (n-m)} - \sum_{n=0}^p \sum_{m=0}^p m a_p(n) a_p(m) e^{-j2\pi f T (m-n)}$$

Si on pose $r = n - m$

$$\begin{aligned}
D &= \sum_{r=-p}^p \sum_{m=0}^p (p+1)a_p(r+m)a_p^*(m)e^{-j2\pi fTr} - \sum_{n=0}^p \sum_{m=0}^p ma_p(n)a_p^*(m)e^{-j2\pi fT(n-m)} \\
&\quad - \sum_{n=0}^p \sum_{m=0}^p ma_p(n)a_p^*(m)e^{-j2\pi fT(m-n)} \\
&= \sum_{r=-p}^p \sum_{m=0}^p (p+1)a_p(r+m)a_p^*(m)e^{-j2\pi fTr} - \sum_{n=0}^p \sum_{m=0}^p (m+n)a_p(n)a_p^*(m)e^{-j2\pi fT(n-m)}
\end{aligned}$$

Si on pose de plus $m + n = r + 2m$, on aura :

$$\begin{aligned}
D &= \sum_{r=-p}^p \sum_{m=0}^p (p+1)a_p(r+m)a_p^*(m)e^{-j2\pi fTr} - \sum_{r=-p}^p \sum_{m=0}^p (r+2m)a_p(r+m)a_p^*(m)e^{-j2\pi fTr} \\
&= \sum_{r=-p}^p \sum_{m=0}^p (p+1-r-2m)a_p(r+m)a_p^*(m)e^{-j2\pi fTr}
\end{aligned}$$

En divisant cette expression par ρ_p^{AR} , on trouve l'expression du dénominateur dans l'équation (4.13). La différence la plus remarquable entre ces deux expressions est que la dernière expression n'utilise que la transformée de Fourier ce qui résulte que la complexité du calcul est réduite à $O(p \log_2 p)$ opérations.

4.3. Réseau de neurones AR/Marple

4.3.1. Principe du réseau proposé

L'algorithme de Marple évite totalement l'inversion de la matrice d'autocorrélation, de plus il n'utilise que des blocs FFT pour évaluer les termes $A(f)$ et $B(f)$.

A partir des équations de Yule-Walker, on peut déduire :

$$\begin{bmatrix} r(0) & r^*(1) & \cdots & r^*(p-1) \\ r(1) & r(0) & \cdots & r^*(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(p-1) & r(p-2) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{bmatrix} = \begin{bmatrix} -r(1) \\ -r(2) \\ \vdots \\ -r(p) \end{bmatrix} \quad (4.23)$$

ou sous la forme matricielle :

$$\mathbf{R}\mathbf{a} = \mathbf{r} \quad (4.24)$$

Donc il suffit d'utiliser un réseau similaire à celui qu'on a décrit dans la section (3.5) du chapitre précédent pour l'estimation des coefficients autorégressifs [44].

$$\mathbf{R}\dot{\mathbf{a}}(t) = -\gamma f(\mathbf{R}\mathbf{a}(t) - \mathbf{r}) \quad (4.25)$$

En utilisant une transformation de Fourier via des blocs FFT sur L point sur les coefficients autorégressifs et les coefficients autorégressifs pondérés, on peut aboutir à une estimation de la densité spectrale suivant le schéma de la figure (4.1).

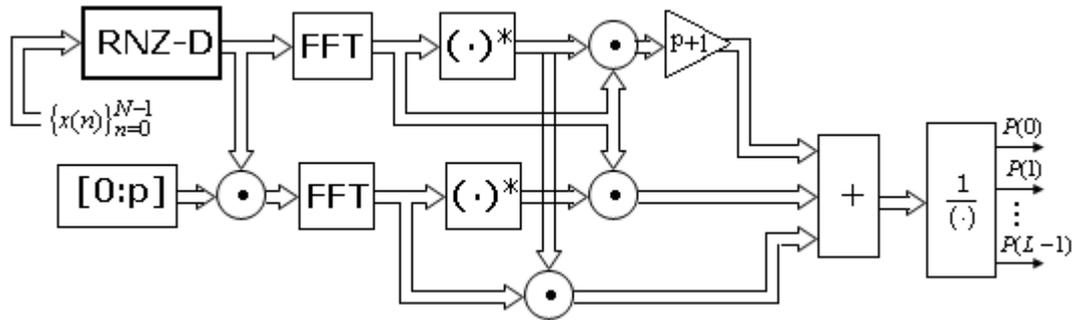


Figure 4.1. Schéma synoptique de l'estimateur neuronal AR/Marple.

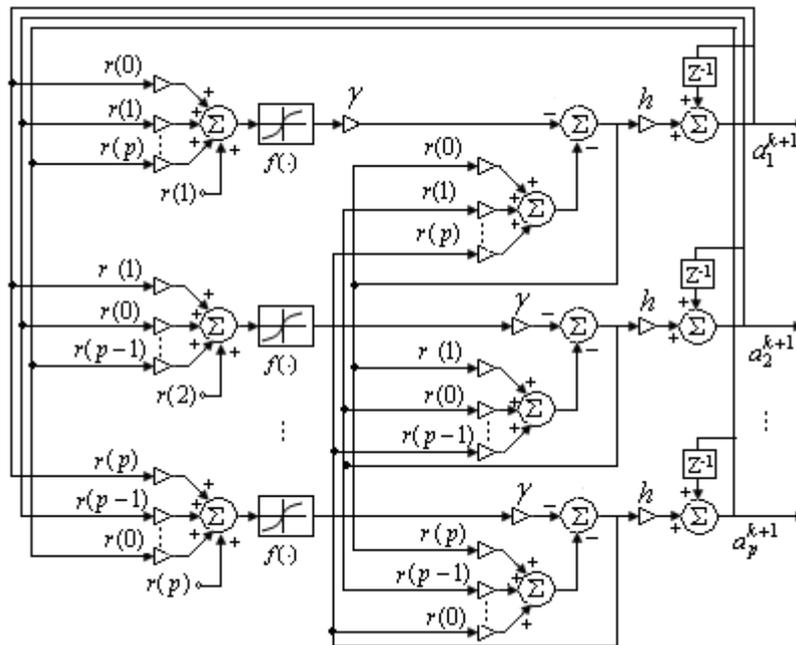


Figure 4.2. Architecture du réseau RNZD.

4.3.2. Résultats de simulations

Le modèle du RNZD-Marple a été implémenté sous Simulink selon la figure (4.3). Le bloc 'signal' sert à générer une séquence d'échantillons qu'on veut estimer sa densité spectrale en utilisant le réseau RNZD-Marple. Le 'RNZD' représenté dans la figure (4.3.b) est similaire au RNZD vu au chapitre précédent.

Dans ce qui suit on s'intéresse à l'estimation spectrale d'un signal composé de sinusoides bruitées. Le modèle du réseau est simulé pour différentes valeurs de la variance du bruit et pour différents ordres.

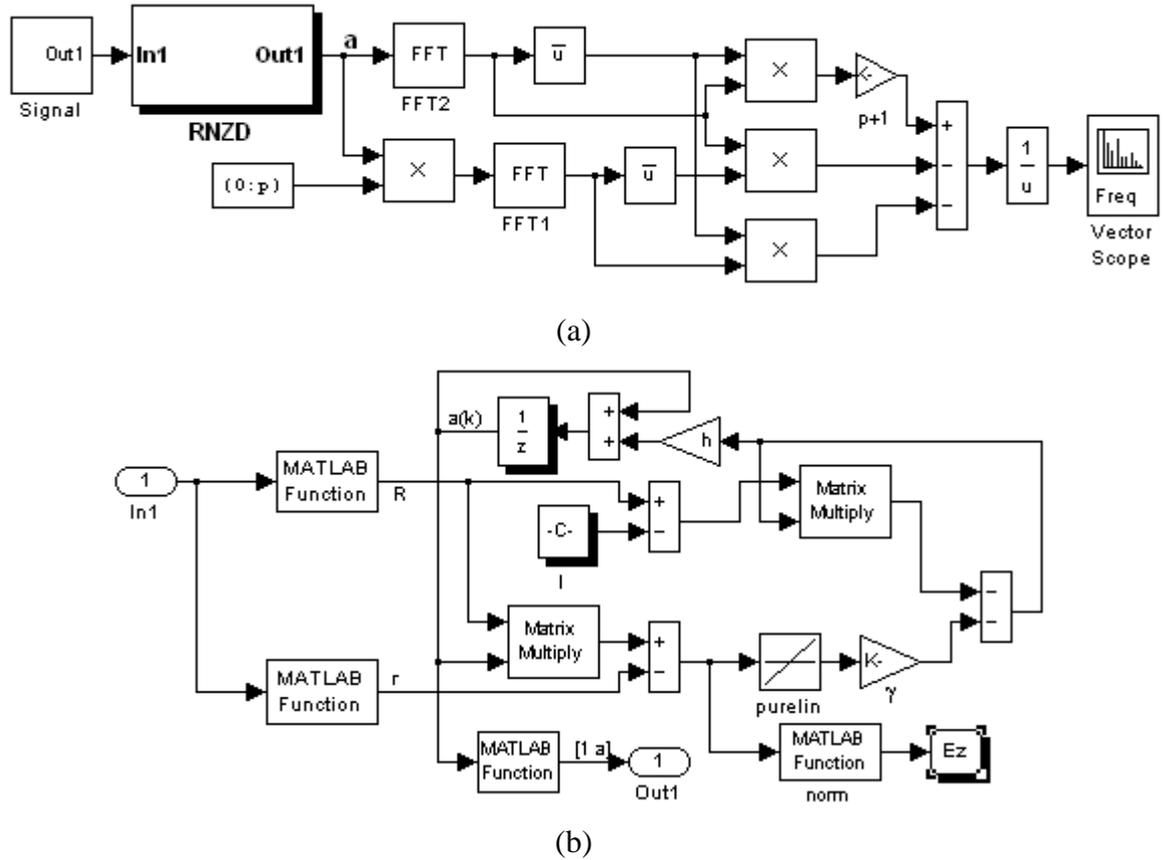
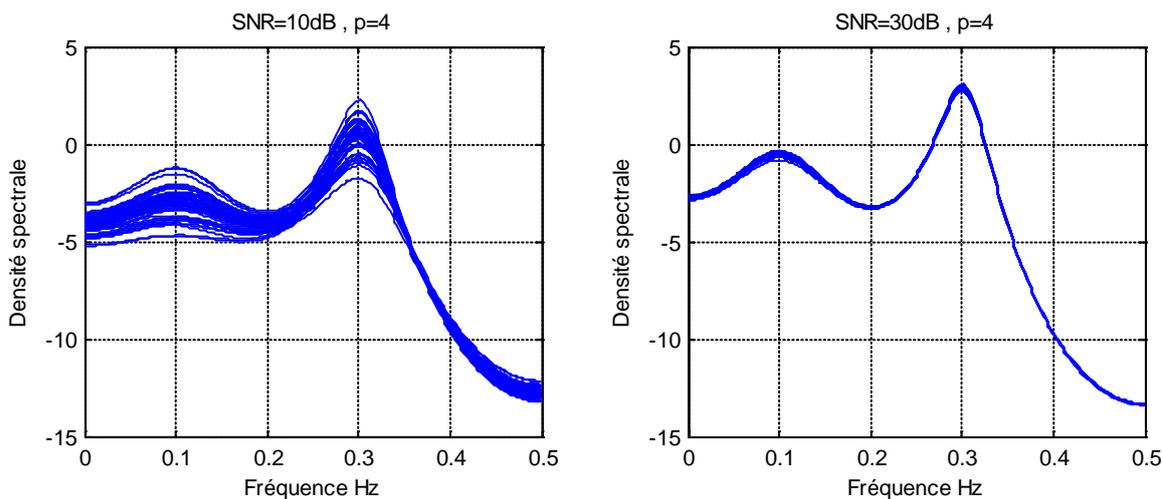


Figure 4.3. Schéma de simulation. (a) le modèle RNZD-Marple, (b) le bloc RNZD.

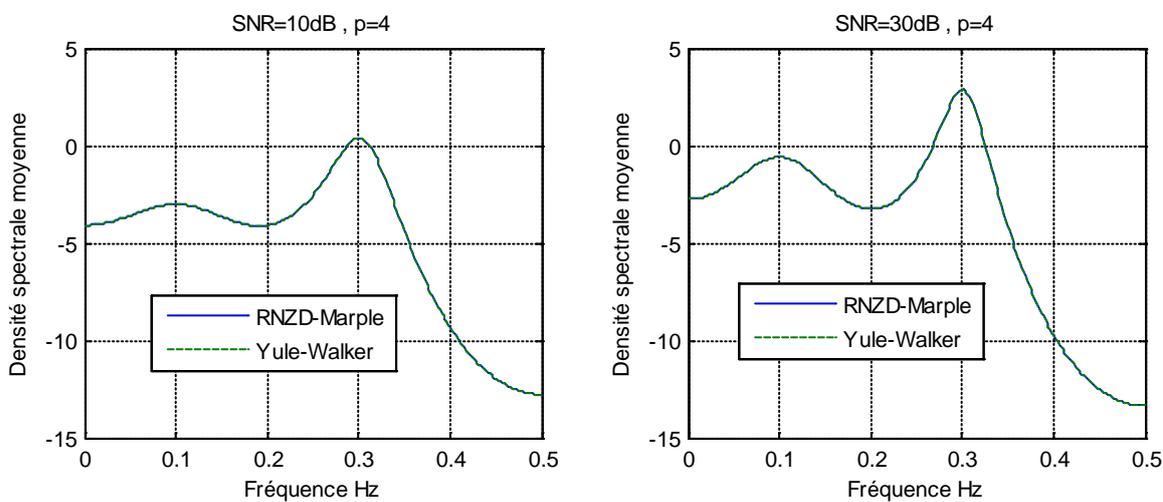
Dans la présente simulation les paramètres suivants sont utilisés :

- le coefficient de convergence $\gamma = 10^4$.
- Le pas de discrétisation $h = 10^{-4}$.
- Le signal utilisé consiste en deux sinusoïdes de fréquences normalisées égales à 0.1, et 0.3 Hz, respectivement et d'amplitudes 1 et 1.5.
- La variance du bruit $SNR_1 = 10dB$, $SNR_2 = 30dB$
- L'ordre du modèle $p_1 = 4$, $p_2 = 8$
- Nombre de points de la FFT : $F = 512$

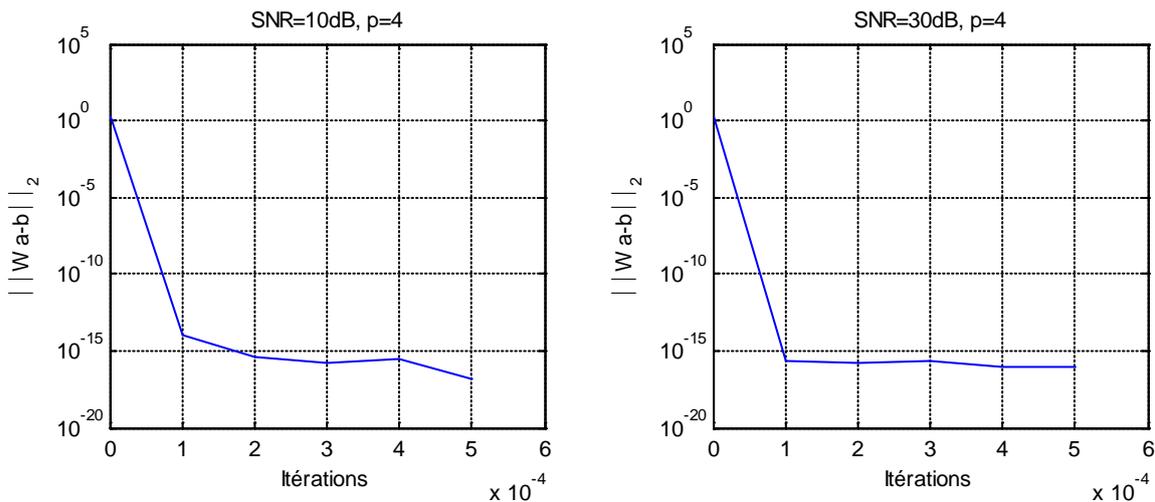
D'après les figures (4.4) et (4.5), on constate que l'approche proposée donne les mêmes performances que la méthode de Marple utilisant les équations de Yule-Walker. Ceci est dû à la convergence du réseau quel que soit l'ordre du modèle ou la sévérité du bruit.



(a) DSP de 50 réalisations.

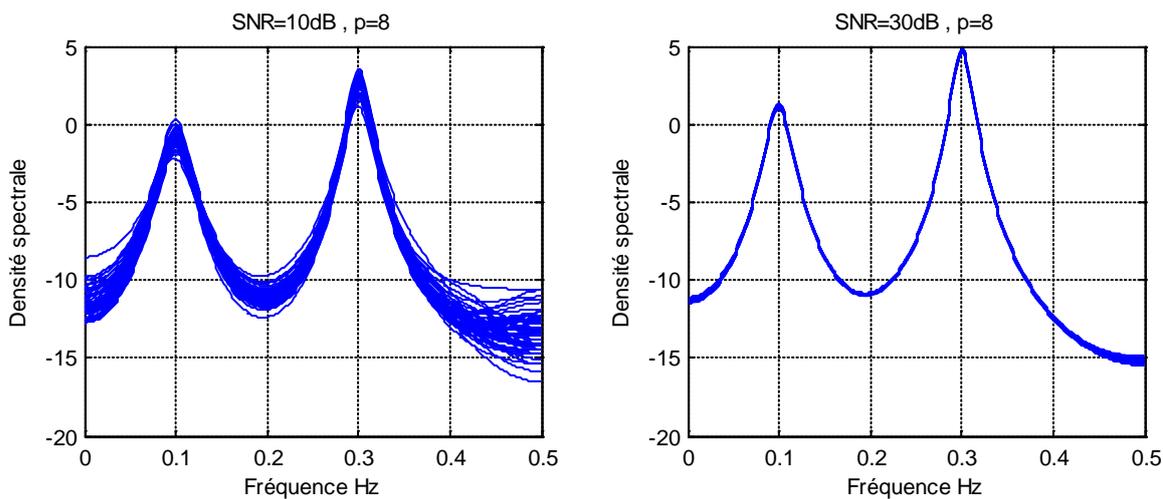


(b) DSP moyenne.

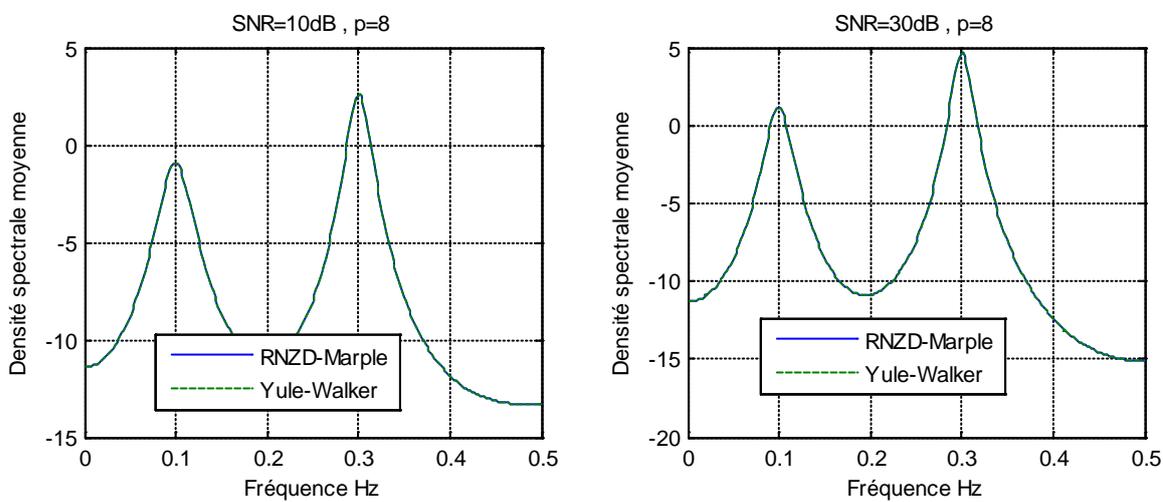


(c) Erreur de convergence

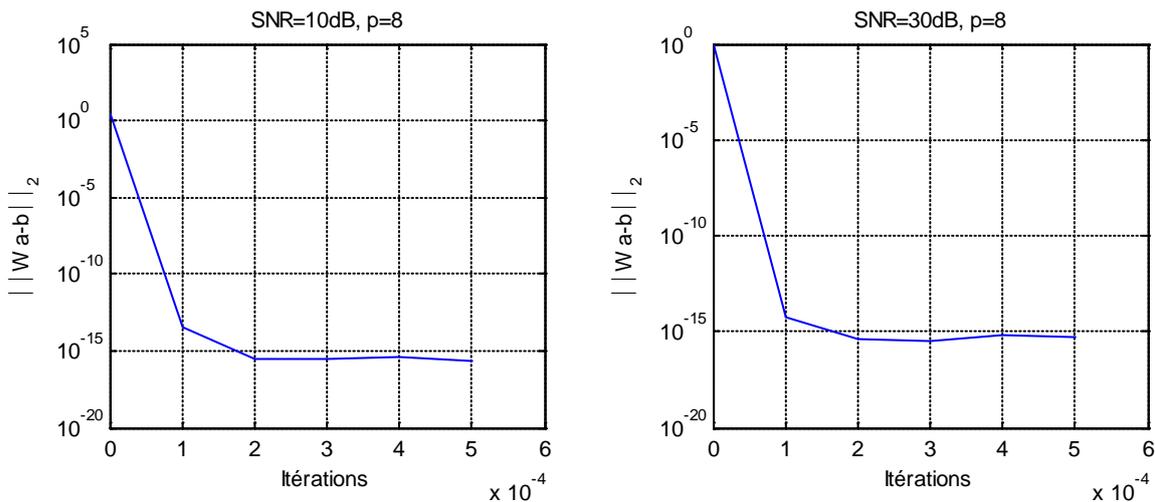
Figure 4.4. Résultats de simulation du modèle RNZD-Marple pour $p = 4$ et différentes valeurs de SNR .



(a) DSP de 50 réalisations.



(b) DSP moyenne



(c) Erreur de convergence

Figure 4.5. Résultats de simulation du modèle RNZD-Marple pour $p = 8$ et différentes valeurs de SNR .

4.4. Architecture réduite proposée

Dans la littérature, il est très connu que les réseaux de neurones se reposent sur le calcul de la multiplication d'une matrice par un vecteur où la matrice contient les poids de connections et le vecteur l'activation des neurones [22]. De plus pour la même matrice de connections, la multiplication doit être faite à chaque itération par un nouveau vecteur issu de la sortie du réseau. C'est pour ces raisons, on cherche à utiliser la structure spéciale de la matrice de connections pour réduire la complexité de calcul et évidemment pour réduire l'architecture du réseau. Dans cette section, on va proposer une architecture réduite du réseau de neurones où les sommes pondérées des entrées sont considérées comme la multiplication de la matrice des connections par le vecteur d'entrée.

Donc on va investiguer une architecture réduite pour le réseau décrit dans la section (4.3.1)

En partant de l'équation (4.25), cette équation peut être mise sous la forme:

$$\dot{\mathbf{a}}(t) = \mathbf{a}'(t) - \mathbf{R}\mathbf{a}(t) - \gamma f(\mathbf{R}\mathbf{a}(t) - \mathbf{r}) \quad (4.26)$$

Une discrétisation de ce modèle en utilisant la règle d'Euler nous permet d'écrire:

$$\mathbf{a}((k+1)h) = \mathbf{a}(kh) - \gamma hf(\mathbf{z}_1(kh) - \mathbf{b}) - \mathbf{z}_2(kh) \quad (4.27)$$

où h et k dénotent le pas d'échantillonnage et l'indice d'itérations respectivement. $\mathbf{z}_1(kh)$ et $\mathbf{z}_2(kh)$ sont deux vecteurs issus de la multiplication de la matrice Toeplitz des connections par le vecteur d'entrée:

$$\mathbf{z}_1(kh) = \mathbf{R}\mathbf{a}(kh) \quad (4.28)$$

$$\mathbf{z}_2(kh) = \mathbf{R}(\mathbf{a}((k+1)h) - \mathbf{a}(kh)) \quad (4.29)$$

Ces deux produits sont évalués par l'algorithme décrit ci-dessous.

4.4.1. Calcul rapide du produit matrice par vecteur

Une matrice Toeplitz est complètement définie par sa première colonne et sa première ligne. Soit $[a_{-n+1}, \dots, a_0, \dots, a_{n-1}]$ un vecteur composé par l'empilement de la première colonne et de la première ligne d'une matrice $\mathbf{A} \in \mathfrak{R}^{n \times n}$ ayant la forme:

$$\mathbf{A} = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots & a_{n-1} \\ a_{-1} & a_0 & a_1 & \cdots & a_{n-2} \\ a_{-2} & a_{-1} & a_0 & \cdots & a_{n-3} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{-n+1} & a_{-n+2} & a_{-n+3} & \cdots & a_0 \end{bmatrix} \quad (4.30)$$

et soit un vecteur arbitraire $\mathbf{x} = [x_0, x_1, \dots, x_n]^T$. Pour calculer le produit \mathbf{Ax} , la matrice \mathbf{A} est transformée en une matrice circulaire $\mathbf{C} \in \mathfrak{R}^{2n \times 2n}$ comme suit [83-86]:

$$\mathbf{C} = \begin{bmatrix} \mathbf{A} & \mathbf{S} \\ \mathbf{S} & \mathbf{A} \end{bmatrix} \quad (4.31)$$

où

$$\mathbf{S} = \begin{bmatrix} 0 & a_{-n+1} & a_{-n+2} & \cdots & a_{-1} \\ a_{n-1} & 0 & a_{-n+1} & \cdots & a_{-2} \\ a_{n-2} & a_{n-1} & 0 & \cdots & a_{n-3} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_1 & a_2 & a_3 & \cdots & 0 \end{bmatrix} \quad (4.32)$$

La matrice \mathbf{S} ne sera jamais calculée explicitement du fait que \mathbf{C} est une matrice Toeplitz dont les colonnes sont décrites par simple rotation de la première colonne de \mathbf{C} et ayant la forme:

$$\mathbf{c}_1 = [a_0 \ a_{-1} \ a_{-2} \ \cdots \ a_{-n+1} \ 0 \ a_{n-1} \ a_{n-2} \ \cdots \ a_1]^T \quad (4.33)$$

Maintenant on va reformuler le produit \mathbf{Ax} comme:

$$\mathbf{C} \cdot \begin{bmatrix} \mathbf{x} \\ \mathbf{0}_n \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{S} \\ \mathbf{S} & \mathbf{A} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x} \\ \mathbf{0}_n \end{bmatrix} = \begin{bmatrix} \mathbf{Ax} \\ \mathbf{Sx} \end{bmatrix} \quad (4.34)$$

On note ici que le vecteur $[\mathbf{x} \ \mathbf{0}_n]^T$ n'est simplement que le vecteur \mathbf{x} ajouté de zéro à la longueur de \mathbf{c}_1 . Pour faciliter l'écriture, le nouveau vecteur sera noté \mathbf{x}_p . Par conséquent l'équation (4.34) sera réécrite sous la forme:

$$\mathbf{C} \cdot \mathbf{x}_p = \begin{bmatrix} \mathbf{Ax} \\ \mathbf{Sx} \end{bmatrix} \quad (4.35)$$

Le produit \mathbf{Ax} peut être calculé avec une grande efficacité en utilisant l'algorithme suivant :

Calculer $\mathbf{X}_p = FFT(\mathbf{x}_p)$

Calculer $\mathbf{w} = FFT(\mathbf{c}_1)$

Calculer le produit élément par élément $\mathbf{H} = \mathbf{X}_p \cdot * \mathbf{w}$

Calculer $\mathbf{z} = IFFT(\mathbf{H})$

Les premiers n éléments du vecteur \mathbf{z} constituent le produit \mathbf{Ax} ; $\mathbf{Ax} = \mathbf{z}(1:n)$

Puisque l'algorithme FFT/IFFT se fait en $O(n \log n)$ opérations, donc on conclut que le produit \mathbf{Ax} peut être lui aussi fait en $O(n \log n)$ opérations [82].

La figure (4.6) montre le schéma bloc l'algorithme rapide pour calculer le produit \mathbf{Ax} .

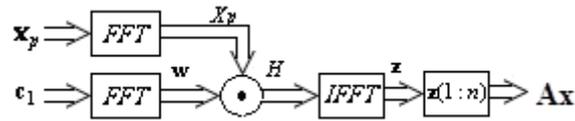


Figure 4.6. Schéma synoptique de l’algorithme rapide pour calculer Ax .

4.4.2. Réseau de neurones proposé

Le réseau de neurone de la figure (4.2) contient deux produits matrice par vecteur. Ces deux produits sont évalués par l’algorithme ci-dessus. Le schéma fonctionnel du réseau réduit est reporté sur la figure (4.7 a). La transformée de Fourier discrète sur $2p$ points du vecteur c_1 et dont les éléments constituent les poids de connections du réseau de neurones (les w_i dans la figure (4.7 b) sont calculés une seule fois au début du processus itératif et reste inchangés. Donc on n’a besoin que de deux blocks FFT/IFFT dans chaque itération.

La figure (4.7 b) montre l’architecture détaillée du réseau RNZD réduit (RNZDR). On constate que la transformée de Fourier de la colonne c_1 constitue la pondération des connections. Le réseau est constitué de $4p$ connections pondérées au lieu de p^2 dans le réseau originale, de deux blocks FFT/IFFT, $3p$ additions et $5p$ multiplications.

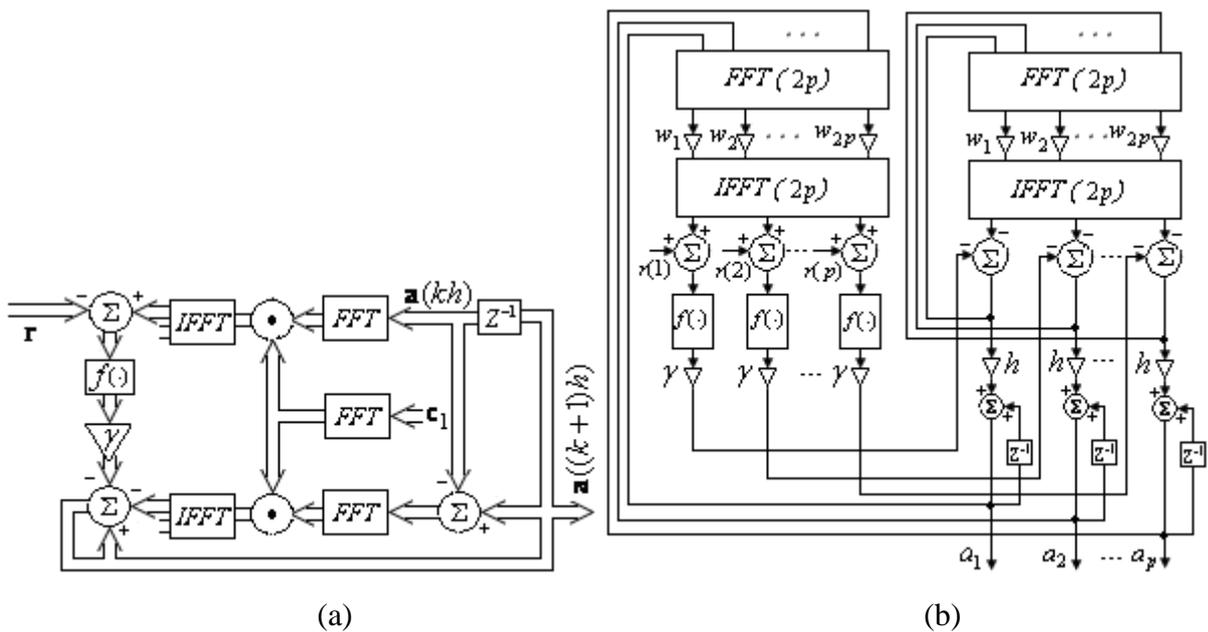


Figure 4.7. (a) Schéma synoptique du modèle RNZDR et (b) son architecture détaillée.

4.4.3. Complexité du réseau

Comme couramment utilisé dans la littérature, la complexité de calcul d'un réseau de neurones est définie par le nombre total de multiplications et d'additions à exécuter dans chaque itération. Le fait que l'algorithme FFT/IFFT sur p points requière $0.5p \log p$ multiplications et $p \log p$ additions de deux éléments, donc le réseau proposé nécessite $6p + 4p \log 2p$ multiplications et $3p + 8p \log 2p$ additions par itération. Par contre le réseau original nécessite $2p^2 + 2p$ multiplications, $2p$ additions de $p+1$ éléments, p additions de 2 éléments. Comme résultat, la complexité du réseau modifié est de $O(p \log p)$ au lieu de $O(p^2)$ pour le réseau original. Ceci est illustré par la figure (4.8).

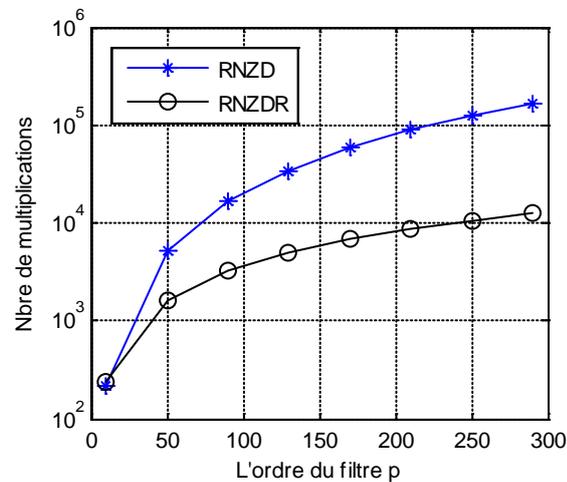


Figure 4.8. Nombre de multiplications nécessaires pour RNZD et sa version réduite

4.5. Performances de la méthode proposée

4.5.1. Précision

Le modèle du RNZD-Marple réduit (RNZDR) a été implémenté sous Simulink selon la figure (4.9 a). Les blocs 'subsystem1' et 'subsystem2' sont détaillés dans la figure (4.9 b). On a appliqué le même signal utilisé précédemment. On constate d'après la figure (4.10) que le réseau à architecture réduite est similaire au réseau normal du point de vue performance.

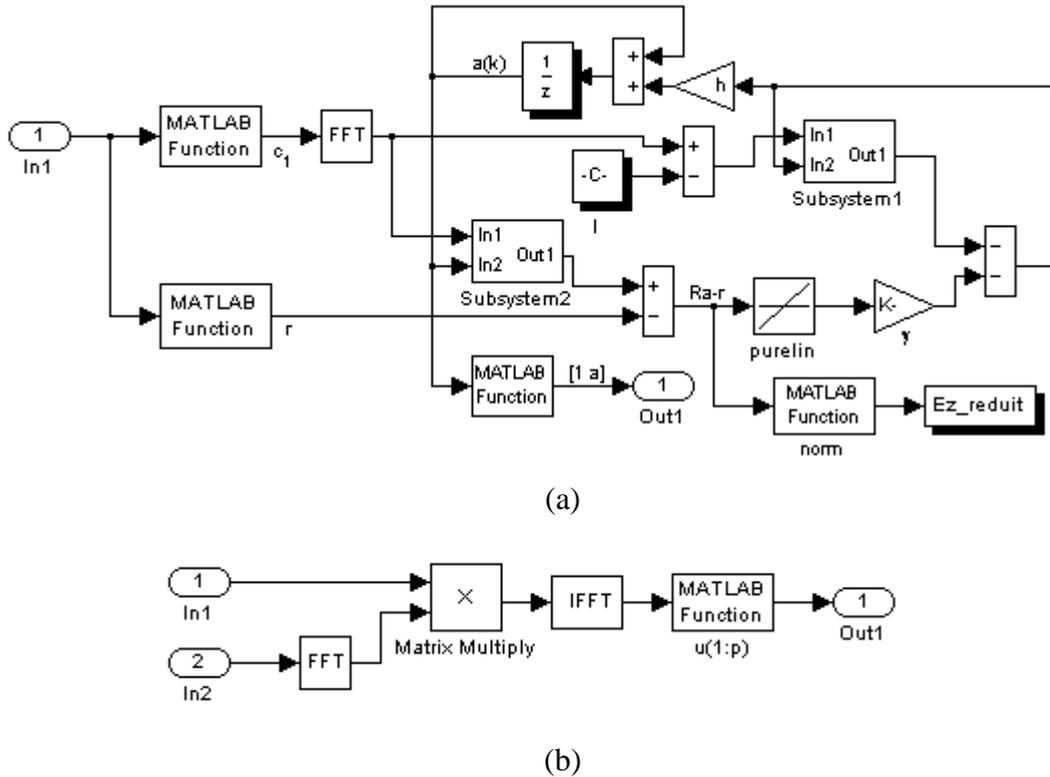


Figure 4.9. Schéma de simulation. (a) le modèle RNZDR, (b) le bloc ‘subsystem1,2’

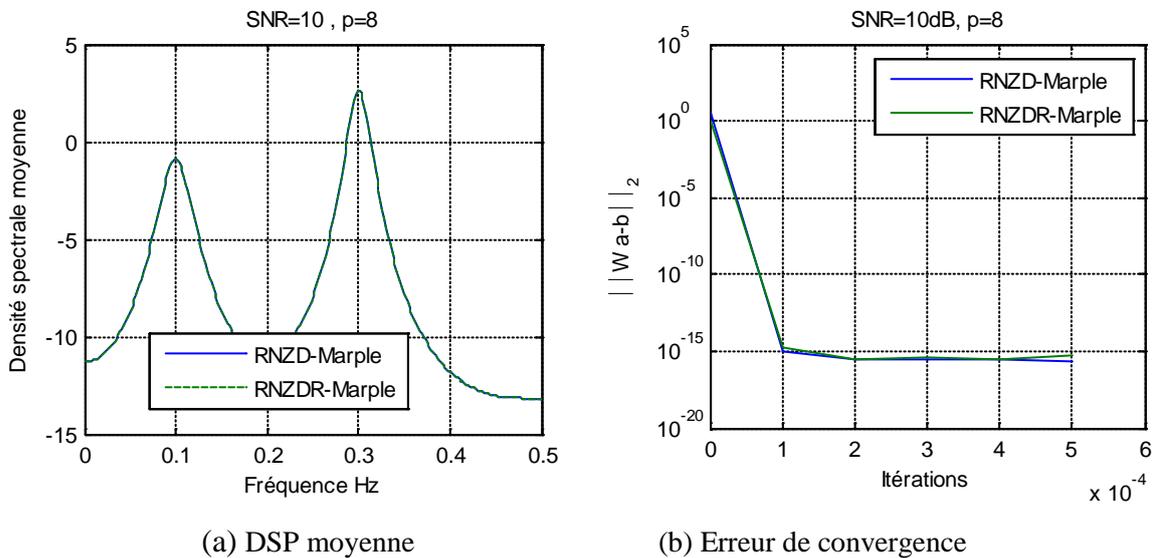


Figure 4.10. Résultats de simulation des modèles RNZD-Marple et RNZDR-Marple.

4.5.2. Comparaison de la complexité

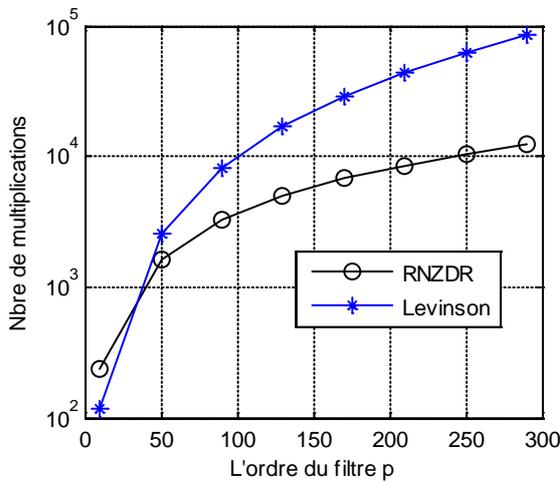
Comme on a mentionné dans l’introduction de ce chapitre, l’algorithme rapide de Musicus et l’algorithme ultrarapide de Marple utilise l’algorithme de Levinson pour estimer les paramètres $a_p(i)$ d’où la nécessité de $p^2 + 2p - 1$ multiplications et $p^2 - p$ additions dans

la première phase de l'estimation [12,20]. Donc la contribution du réseau proposé réside dans l'estimation de ces paramètres en $6p + 4p \log_2 2p$ multiplications et $3p + 8p \log_2 2p$ additions. Le tableau (4.1) montre une comparaison de la complexité de la méthode proposée avec des différentes méthodes citées dans [20].

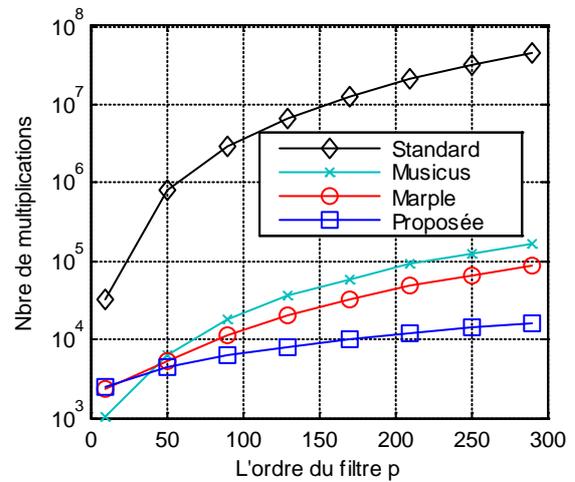
	Nombre de multiplications	Nombre d'additions
Standard [09]	$F(p+1)^2 + (p+1)^3$	$p(p+2)F + p^{(p+1)}$
Musicus [12]	$p^2 + 2p - 1 + p(p+2) - \frac{p}{2} - 1$ $+ \left(\frac{\log(2p+1)}{\log(F)}\right) \frac{F}{2} \log(F) + F$	$p^2 - p + \frac{p(p+1)}{2} + \left(\frac{\log(2p+1)}{\log(F)}\right) F \log(F)$
Marple [20]	$p^2 + 2p - 1 + 5F + p - 1$ $+ 2 \times \left(\frac{\log(p+1)}{\log(F)}\right) \frac{F}{2} \log(F)$	$p^2 - p + 2 \times \left(\frac{\log(2p+1)}{\log(F)}\right) F \log(F) + 2F$
Proposée	$6p + 4p \log 2p + 5F + p - 1$ $+ 2 \times \left(\frac{\log(p+1)}{\log(F)}\right) \frac{F}{2} \log(F)$	$3p + 8p \log 2p + 2 \times \left(\frac{\log(2p+1)}{\log(F)}\right) F \log(F) + 2F$

Tableau 4.1. Comparaison de la complexité avec d'autres méthodes [20]

La figure (4.11 a) montre une comparaison de la complexité du calcul du réseau dans ces deux versions normale et réduite. Tandis que la figure (4.11.b), elle montre l'évolution du nombre de multiplications avec la longueur du filtre pour les différentes méthodes du tableau. On constate que le réseau proposé a une nette amélioration de la complexité par rapport aux autres méthodes.



(a) RNZDR/algorithmme de Levinson



(b) RNZDR-Marple/ Standard, Musicus et Marple

Figure 4. 11. Complexité du réseau RNZD réduit.

4.6. Conclusion

Dans ce chapitre, on a proposé d'introduire le réseau de neurones de Zhang dans l'estimation de la densité spectrale de puissance utilisant la méthode du minimum de variance. Le réseau proposé sert à estimer les paramètres AR dans l'algorithme ultra rapide de Marple au lieu de l'algorithme de Levinson.

La structure Toeplitz des équations de Yule-Walker nous a permis aussi de proposer une architecture réduite du réseau de Zhang utilisant la transformée de Fourier rapide.

Le réseau proposé offre une nette amélioration de la complexité de calculs par rapport aux méthodes de Musicus et Marple.

Chapitre 5

Estimation spectrale multidimensionnelle par le réseau de Zhang

5.1. Introduction

Dans le premier chapitre, on a présenté deux méthodes non paramétriques basées sur le filtrage adaptatif; la méthode de Capon et la méthode APES pour un signal monodimensionnel. Dans ce chapitre, on s'intéresse à l'estimation du spectre d'amplitudes des signaux multidimensionnels (2D, 3D). L'estimation spectrale multidimensionnelle dont le spectre est représenté par des vecteurs d'ondes est rencontrée dans plusieurs domaines comme le radar, le sonar, la télédétection, et la télécommunication [86-97]. Dans la méthode de Capon ou la méthode APES, l'inversion de la matrice de corrélation s'impose, et comme les dimensions de la matrice de corrélation augmentent avec les dimensions du signal, l'utilisation de ces méthodes dans les applications en temps réel exige un temps de calcul très intense. Pour pallier ce problème, plusieurs algorithmes ont été proposés. Ces algorithmes sont essentiellement basés sur la récursivité et l'utilisation du produit polynomial via la transformée de Fourier rapide [13,16,18,88,89]. Comme indiqué dans le début de cet paragraphe, on est contraint d'inverser une matrice de grande taille. Une solution s'avère intéressante consiste à utiliser le calcul parallèle dont les réseaux de neurones récurrents constituent une alternative prometteuse [24]. L'approche proposée consiste à utiliser le réseau de Zhang pour le calcul direct de l'inverse de la matrice de corrélation, puis utiliser la transformée de Fourier rapide pour l'évaluation de l'expression du spectre.

5.2. Méthodes de Capon et APES multidimensionnelles

5.2.1. Spectre des signaux 2D

Les méthodes étudiées ont été étendues pour l'estimation des spectres d'amplitudes des signaux de deux dimensions qui a un grand intérêt dans plusieurs applications dont la plus

marquée est celle des radars à synthèse d'ouverture [46-49]. On considère un signal 2D composé de K sinusoides noyées dans un bruit complexe additif de deux dimensions [91-92] :

$$x(n, \bar{n}) = \sum_{k=1}^K \alpha_k e^{j2\pi(f_k n + \bar{f}_k \bar{n})} + v(n, \bar{n}) ; n = 0, 1, \dots, N-1; \bar{n} = 0, 1, \dots, \bar{N}-1 \quad (5.1)$$

α_k est l'amplitude réel ou complexe de la $k^{\text{ième}}$ 2D sinusoides à la paire fréquentielle (f_k, \bar{f}_k) et $v(n, \bar{n})$ est un 2D bruit complexe gaussien, centré et de variance σ^2 . Le problème à poser est similaire à celui des signaux 1D vu dans le premier chapitre, il consiste à estimer le spectre d'amplitude à partir de la matrice des données $\{x(n, \bar{n})\}_{n=0, \bar{n}=0}^{N-1, \bar{N}-1}$.

L'équation (5.1) peut être mise sous la forme:

$$\mathbf{X} = \Psi \Lambda \bar{\Psi}^T + \mathbf{V} \quad (5.2)$$

$$\text{avec : } \mathbf{X} = \begin{bmatrix} x(0,0) & \dots & x(0, \bar{N}-1) \\ x(1,0) & \dots & x(1, \bar{N}-1) \\ \vdots & \vdots & \vdots \\ x(N-1,0) & \dots & x(N-1, \bar{N}-1) \end{bmatrix}; \mathbf{V} = \begin{bmatrix} v(0,0) & \dots & v(0, \bar{N}-1) \\ v(1,0) & \dots & v(1, \bar{N}-1) \\ \vdots & \vdots & \vdots \\ v(N-1,0) & \dots & v(N-1, \bar{N}-1) \end{bmatrix};$$

$$\Psi = \begin{bmatrix} 1 & \dots & 1 \\ e^{j2\pi f_1} & \dots & e^{j2\pi f_k} \\ \vdots & \vdots & \vdots \\ e^{j(N-1)2\pi f_1} & \dots & e^{j(N-1)2\pi f_k} \end{bmatrix}; \bar{\Psi} = \begin{bmatrix} 1 & \dots & 1 \\ e^{j2\pi \bar{f}_1} & \dots & e^{j2\pi \bar{f}_k} \\ \vdots & \vdots & \vdots \\ e^{j(\bar{N}-1)2\pi \bar{f}_1} & \dots & e^{j(\bar{N}-1)2\pi \bar{f}_k} \end{bmatrix}; \Lambda = \begin{bmatrix} \alpha_1 & 0 \dots & 0 \\ 0 & \alpha_2 \dots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \dots & \alpha_K \end{bmatrix}$$

Pour estimer la densité spectrale du signal par la méthode de Capon et APES, on suppose pour une paire fréquentielle (f, \bar{f}) , le signal est constitué de deux termes comme on a vu pour le cas 1D, une sinusoides pure et un terme de bruit:

$$x(n, \bar{n}) = \alpha(f, \bar{f}) e^{j2\pi(fn + \bar{f}\bar{n})} + v_{(f, \bar{f})}(n, \bar{n}) \quad (5.3)$$

Par la subdivision de la matrice d'observations \mathbf{X} en $L \times \bar{L}$ sous matrices de taille $(M \times \bar{M})$.

Chaque sous matrice est de la forme:

$$\mathbf{X}_{l, \bar{l}} = \begin{bmatrix} x(l, \bar{l}) & \dots & x(l, \bar{l} + \bar{M} - 1) \\ \vdots & \ddots & \vdots \\ x(l + M - 1, \bar{l}) & \dots & x(l + M - 1, \bar{l} + \bar{M} - 1) \end{bmatrix}; \begin{cases} l = 0, 1, \dots, L-1 \\ \bar{l} = 0, 1, \dots, \bar{L}-1 \end{cases} \quad (5.4)$$

avec :

$$L = N - M + 1; \bar{L} = \bar{N} - \bar{M} + 1 \quad (5.5)$$

Soit le vecteur $\mathbf{y}_{l, \bar{l}}$ issu de l'empilement des colonnes de la matrice $\mathbf{X}_{l, \bar{l}}$.

$$\mathbf{y}_{l, \bar{l}} = \text{vec}(\mathbf{X}_{l, \bar{l}}) = [x(l, \bar{l}) \dots x(l + M - 1, \bar{l}) \dots x(l, \bar{l} + \bar{M} - 1) \dots x(l + M - 1, \bar{l} + \bar{M} - 1)]^T \quad (5.6)$$

Cette dernière définition nous permet de définir une nouvelle matrice d'observations de la forme:

$$\mathbf{Y} = [\mathbf{y}_{0,0} \quad \cdots \quad \mathbf{y}_{L-1,0} \quad \mathbf{y}_{0,1} \quad \cdots \quad \mathbf{y}_{L-1,L-1}] \quad (5.7)$$

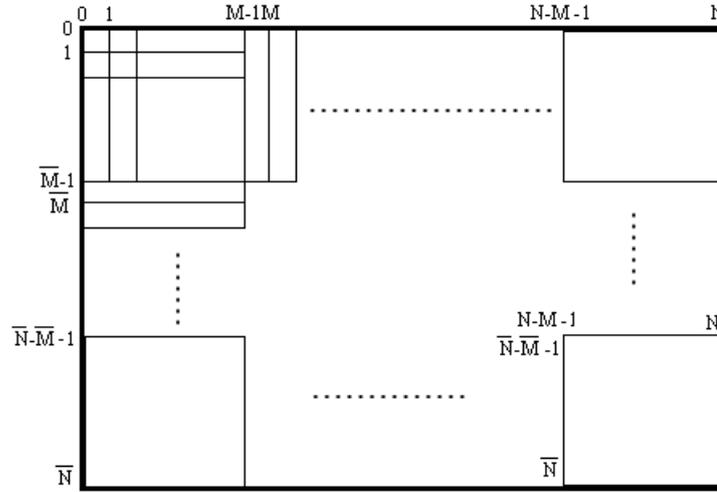


Figure 5.1. Segmentation de la matrice d'observations.

En introduisant le vecteur de Fourier 2D, pour deux entiers M et \bar{M} quelconques :

$$\mathbf{a}_{M\bar{M}}(f, \bar{f}) = \mathbf{a}_{\bar{M}}(\bar{f}) \otimes \mathbf{a}_M(f) = \begin{bmatrix} 1 \\ e^{j2\pi\bar{f}} \\ \vdots \\ e^{j2\pi(\bar{M}-1)\bar{f}} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ e^{j2\pi f} \\ \vdots \\ e^{j2\pi(M-1)f} \end{bmatrix} \quad (5.8)$$

Où \otimes désigne le produit de Kronecker. Pour rappeler, Le produit de Kronecker de deux matrices A, B est une matrice bloc $C : C = A \otimes B$ où $[C]_{i,j} = a_{i,j} B$ désigne l'élément de la $i^{\text{ème}}$ ligne bloc et de la $j^{\text{ème}}$ colonne bloc de C .

On définit maintenant la matrice $\mathbf{A}(f, \bar{f})$ comme :

$$\mathbf{A}(f, \bar{f}) = \mathbf{a}_{M\bar{M}}(f, \bar{f}) \mathbf{a}_{M\bar{M}}^T(f, \bar{f}) \quad (5.9)$$

On répète les mêmes opérations qu'on a faites avec la matrice \mathbf{X} mais cette fois-ci sur la matrice $\mathbf{V}(f, \bar{f})$ dont les éléments représentent le terme du bruit de l'équation (5.2) et les sinusoides en dehors de la paire (f, \bar{f}) . La matrice résultante $\mathbf{W}(f, \bar{f})$ est de la forme:

$$\mathbf{W}(f, \bar{f}) = [\mathbf{v}_{0,0} \quad \cdots \quad \mathbf{v}_{L-1,0} \quad \mathbf{v}_{0,1} \quad \cdots \quad \mathbf{v}_{L-1,L-1}] \quad (5.10)$$

où les vecteurs $\mathbf{v}_{l,\bar{l}}$ sont issus de l'empilement des colonnes de la matrice :

$$\mathbf{v}_{l,\bar{l}} = \begin{bmatrix} v_{(f,\bar{f})}(l, \bar{l}) & \cdots & v_{(f,\bar{f})}(l, \bar{l} + \bar{M} - 1) \\ \vdots & \ddots & \vdots \\ v_{(f,\bar{f})}(l + M - 1, \bar{l}) & \cdots & v_{(f,\bar{f})}(l + M - 1, \bar{l} + \bar{M} - 1) \end{bmatrix} \quad (5.11)$$

Finalement, on peut formuler le système d'équations de la forme:

$$\mathbf{Y} = \alpha(f, \bar{f})\mathbf{A}(f, \bar{f}) + \mathbf{W}(f, \bar{f}) \quad (5.12)$$

avec :

$$\mathbf{A}(f, \bar{f}) = \mathbf{a}_{M, \bar{M}}(f, \bar{f})\mathbf{a}_{L, \bar{L}}^T(f, \bar{f}) \quad (5.13)$$

5.2.1.1. Méthode de Capon 2D

Soit \mathbf{H} une matrice $(M \times \bar{M})$ qui contient les coefficients du filtre, et \mathbf{Z} une matrice $(L \times \bar{L})$ qui contient les données obtenues par le passage des données \mathbf{X} à travers le filtre \mathbf{H} :

$$\mathbf{Z} = \mathbf{H} * \mathbf{X} \quad (5.14)$$

$$z(l, \bar{l}) = \sum_{m=0}^{M-1} \sum_{\bar{m}=0}^{\bar{M}-1} h^*(m, \bar{m})x(l-m, \bar{l}-\bar{m}) = \text{vec}^H(\mathbf{H})\mathbf{y}_{l, \bar{l}} \quad (5.15)$$

Les coefficients du filtre \mathbf{H} sont choisis de façon à minimiser la puissance du signal en sortie du filtre avec la contrainte qu'une exponentielle, de fréquence (f, \bar{f}) , à l'entrée du filtre passe sans distorsion [90-91] :

$$\min_{\mathbf{H}(f, \bar{f})} \|\mathbf{H}(f, \bar{f}) * \mathbf{X}\|^2 \quad (5.16)$$

Sous la contrainte :

$$\text{vec}^H(\mathbf{H}(f, \bar{f}))\mathbf{a}_{M, \bar{M}}(f, \bar{f}) = 1 \quad (5.17)$$

Suivant les mêmes démarches utilisées dans le cas de Capon 1D, on obtient l'expression du filtre :

$$\mathbf{H}_{\text{Capon2D}}(f, \bar{f}) = \frac{\mathbf{R}^{-1}\mathbf{a}_{M, \bar{M}}(f, \bar{f})}{\mathbf{a}_{M, \bar{M}}^H(f, \bar{f})\mathbf{R}^{-1}\mathbf{a}_{M, \bar{M}}(f, \bar{f})} \quad (5.18)$$

La densité spectrale estimée par la méthode de Capon $\hat{\alpha}(f)$ est donnée par :

$$\hat{\alpha}_{\text{Capon2D}}(f, \bar{f}) = \frac{\mathbf{a}_{M, \bar{M}}^H(f, \bar{f})\mathbf{R}^{-1}\mathbf{G}(f, \bar{f})}{\mathbf{a}_{M, \bar{M}}^H(f, \bar{f})\mathbf{R}^{-1}\mathbf{a}_{M, \bar{M}}(f, \bar{f})} \quad (5.19)$$

Avec :

$$\mathbf{G}(f, \bar{f}) = \frac{1}{L\bar{L}} \sum_{l=0}^{L-1} \sum_{\bar{l}=0}^{\bar{L}-1} \mathbf{y}_{l, \bar{l}} e^{-j2\pi(fl + \bar{f}\bar{l})} \quad (5.20)$$

$$\mathbf{R} \approx \hat{\mathbf{R}} = \frac{1}{L\bar{L}} \sum_{l=0}^{L-1} \sum_{\bar{l}=0}^{\bar{L}-1} \mathbf{y}_{l, \bar{l}} \mathbf{y}_{l, \bar{l}}^H \quad (5.21)$$

5.2.1.2. Méthode APES 2D

Le principe de la méthode APES 2D est similaire à celui de Capon 2D, pour une paire de fréquence (f, \bar{f}) , on cherche à concevoir un filtre $\mathbf{H}(f, \bar{f})$ qui minimise l'erreur entre la sortie du filtre et la composante du signal à la fréquence désirée. Ce filtre est obtenu par la minimisation de la fonction objective [90] :

$$\min_{\mathbf{H}(f, \bar{f}), \alpha(f, \bar{f})} \left\| \text{vec}^H(\mathbf{H}(f, \bar{f}) * \mathbf{X}) - \alpha(f, \bar{f}) \mathbf{a}_{L, \bar{L}}(f, \bar{f}) \right\|^2 \quad (5.22)$$

Sous la contrainte :

$$\text{vec}^H(\mathbf{H}(f, \bar{f})) \mathbf{a}_{M, \bar{M}}(f, \bar{f}) = 1 \quad (5.23)$$

La solution de ce problème d'optimisation donne :

$$\mathbf{H}_{APES2D}(f, \bar{f}) = \frac{\mathbf{Q}^{-1} \mathbf{a}_{M, \bar{M}}(f, \bar{f})}{\mathbf{a}_{M, \bar{M}}^H(f, \bar{f}) \mathbf{Q}^{-1} \mathbf{a}_{M, \bar{M}}(f, \bar{f})} \quad (5.24)$$

La densité spectrale des données $\{x(n, \bar{n})\}_{n=0, \bar{n}=0}^{N-1, \bar{N}-1}$ est :

$$\hat{\alpha}_{APES2D}(f, \bar{f}) = \frac{\mathbf{a}_{M, \bar{M}}^H(f, \bar{f}) \mathbf{Q}^{-1} \mathbf{G}(f, \bar{f})}{\mathbf{a}_{M, \bar{M}}^H(f, \bar{f}) \mathbf{Q}^{-1} \mathbf{a}_{M, \bar{M}}(f, \bar{f})} \quad (5.25)$$

$$\text{avec : } \hat{\mathbf{Q}} = \hat{\mathbf{R}} - \mathbf{G}(f, \bar{f}) \mathbf{G}^H(f, \bar{f}) \quad (5.26)$$

Pour éviter l'inversion de la matrice $\hat{\mathbf{Q}}$ pour chaque fréquence, on utilise [90] :

$$\hat{\mathbf{Q}}^{-1}(f, \bar{f}) = \hat{\mathbf{R}}^{-1} + \frac{\hat{\mathbf{R}}^{-1} \mathbf{G}(f, \bar{f}) \mathbf{G}^H(f, \bar{f}) \hat{\mathbf{R}}^{-1}}{1 - \mathbf{G}^H(f, \bar{f}) \hat{\mathbf{R}}^{-1} \mathbf{G}(f, \bar{f})} \quad (5.27)$$

5.2.2. Spectre des signaux 3D

On considère un signal 3D composé de K sinusoides noyées dans un bruit complexe additif de trois dimensions [92] :

$$x(n, \bar{n}, \bar{\bar{n}}) = \sum_{k=1}^K \alpha_k e^{j2\pi(f_k n + \bar{f}_k \bar{n} + \bar{\bar{f}}_k \bar{\bar{n}})} + v(n, \bar{n}, \bar{\bar{n}}) \quad (5.28)$$

$$n = 0, 1, \dots, N-1; \bar{n} = 0, 1, \dots, \bar{N}-1; \bar{\bar{n}} = 0, 1, \dots, \bar{\bar{N}}-1$$

α_k est l'amplitude réel ou complexe de la $k^{\text{ième}}$ 3D sinusoides à la triplette fréquentielle $(f_k, \bar{f}_k, \bar{\bar{f}}_k)$ et $v(n, \bar{n}, \bar{\bar{n}})$ est un 3D bruit complexe gaussien, centré et de variance σ^2 . Le problème consiste à estimer le spectre d'amplitude à partir de la matrice des données

$$\{x(n, \bar{n}, \bar{\bar{n}})\}_{n=0, \bar{n}=0, \bar{\bar{n}}=0}^{N-1, \bar{N}-1, \bar{\bar{N}}-1}$$

5.2.2.1. Méthode de Capon 3D

Soit $\mathbf{h}_{\overline{MM}\overline{M}} \in \mathbb{C}^{\overline{MM}\overline{M} \times 1}$ un filtre à réponse impulsionnelle finie, et $\mathbf{y}_{\overline{MM}\overline{M}}(l, \bar{l}, \bar{\bar{l}})$ le vecteur des données correspondant:

$$\mathbf{y}_{\overline{MM}\overline{M}}(l, \bar{l}, \bar{\bar{l}}) = \text{vec}(\mathbf{X}_{\overline{M}, \overline{M}, \overline{M}}(l, \bar{l}, \bar{\bar{l}})) \quad (5.29)$$

où :

$$\mathbf{X}_{\overline{M}, \overline{M}, \overline{M}}(l, \bar{l}, \bar{\bar{l}}) = \begin{bmatrix} x(l, \bar{l}, \bar{\bar{l}}) & \dots & x(l, \bar{l} + \overline{M} - 1, \bar{\bar{l}}) \\ \vdots & \ddots & \vdots \\ x(l + \overline{M} - 1, \bar{l}, \bar{\bar{l}}) & \dots & x(l + \overline{M} - 1, \bar{l} + \overline{M} - 1, \bar{\bar{l}}) \end{bmatrix} \quad (5.30)$$

$$l = 0, 1, \dots, L - 1, \bar{l} = 0, 1, \dots, \overline{L} - 1, \bar{\bar{l}} = 0, 1, \dots, \overline{\overline{L}} - 1$$

$$L = N - M + 1; \overline{L} = \overline{N} - \overline{M} + 1; \overline{\overline{L}} = \overline{\overline{N}} - \overline{\overline{M}} + 1 \quad (5.31)$$

En définissant un vecteur 3D fréquence:

$$\mathbf{a}_{\overline{MM}\overline{M}}(f, \bar{f}, \bar{\bar{f}}) = \mathbf{a}_{\overline{M}}(\bar{f}) \otimes \mathbf{a}_{\overline{M}}(\bar{\bar{f}}) \otimes \mathbf{a}_M(f) \quad (5.32)$$

Le filtre 3-D Capon peut être conçu comme:

$$\min_{\mathbf{h}_{\overline{MM}\overline{M}}} \mathbf{h}_{\overline{MM}\overline{M}}^H \mathbf{R} \mathbf{h}_{\overline{MM}\overline{M}} \quad (5.33)$$

Sous la contrainte:

$$\mathbf{h}_{\overline{MM}\overline{M}}^H \mathbf{a}_{\overline{MM}\overline{M}}(f, \bar{f}, \bar{\bar{f}}) = 1 \quad (5.34)$$

où:

$$\mathbf{R} \approx \hat{\mathbf{R}} = \frac{1}{\overline{LL}} \sum_{l=0}^{L-1} \sum_{\bar{l}=0}^{\overline{L}-1} \sum_{\bar{\bar{l}}=0}^{\overline{\overline{L}}-1} \mathbf{y}_{\overline{MM}\overline{M}}(l, \bar{l}, \bar{\bar{l}}) \mathbf{y}_{\overline{MM}\overline{M}}^H(l, \bar{l}, \bar{\bar{l}}) \quad (5.35)$$

Par conséquent le spectre d'amplitude estimé via le filtre de Capon 3D est donné par:

$$\hat{\alpha}_{\text{Capon3D}}(f, \bar{f}, \bar{\bar{f}}) = \frac{\mathbf{a}_{\overline{MM}\overline{M}}^H(f, \bar{f}, \bar{\bar{f}}) \mathbf{R}^{-1} \mathbf{G}_{\overline{MM}\overline{M}}(f, \bar{f}, \bar{\bar{f}})}{\mathbf{a}_{\overline{MM}\overline{M}}^H(f, \bar{f}, \bar{\bar{f}}) \mathbf{R}^{-1} \mathbf{a}_{\overline{MM}\overline{M}}(f, \bar{f}, \bar{\bar{f}})} \quad (5.36)$$

où:

$$\mathbf{G}_{\overline{MM}\overline{M}}(f, \bar{f}, \bar{\bar{f}}) = \frac{1}{\overline{LL}\overline{\overline{L}}} \sum_{l=0}^{L-1} \sum_{\bar{l}=0}^{\overline{L}-1} \sum_{\bar{\bar{l}}=0}^{\overline{\overline{L}}-1} \mathbf{y}_{\overline{MM}\overline{M}}(l, \bar{l}, \bar{\bar{l}}) e^{-j2\pi(f\bar{l} + \bar{\bar{l}})} \quad (5.37)$$

5.2.2.2. Méthode APES 3D

Le principe de la méthode APES3D est similaire à celui de Capon 3D. On remplace tout simplement la matrice $\mathbf{R}_{\overline{MM}\overline{M}, \overline{MM}\overline{M}}^{-1}$ par la matrice $\mathbf{Q}_{\overline{MM}\overline{M}, \overline{MM}\overline{M}}^{-1}$ dans l'expression du spectre :

$$\hat{\alpha}_{APES3D}(f, \bar{f}, \bar{\bar{f}}) = \frac{\mathbf{a}_{MM\bar{M}}^H(f, \bar{f}, \bar{\bar{f}})\mathbf{Q}^{-1}\mathbf{G}_{MM\bar{M}}(f, \bar{f}, \bar{\bar{f}})}{\mathbf{a}_{MM\bar{M}}^H(f, \bar{f}, \bar{\bar{f}})\mathbf{Q}^{-1}\mathbf{a}_{MM\bar{M}}(f, \bar{f}, \bar{\bar{f}})} \quad (5.38)$$

avec:

$$\hat{\mathbf{Q}} = \hat{\mathbf{R}} - \mathbf{G}_{MM\bar{M}}(f, \bar{f}, \bar{\bar{f}})\mathbf{G}_{MM\bar{M}}^H(f, \bar{f}, \bar{\bar{f}}) \quad (5.39)$$

Pour éviter l'inversion de la matrice $\hat{\mathbf{Q}}$ pour chaque fréquence, on utilise [45] :

$$\hat{\mathbf{Q}}^{-1}(f, \bar{f}, \bar{\bar{f}}) = \hat{\mathbf{R}}^{-1} + \frac{\hat{\mathbf{R}}^{-1}\mathbf{G}(f, \bar{f}, \bar{\bar{f}})\mathbf{G}^H(f, \bar{f}, \bar{\bar{f}})\hat{\mathbf{R}}^{-1}}{1 - \mathbf{G}^H(f, \bar{f}, \bar{\bar{f}})\hat{\mathbf{R}}^{-1}\mathbf{G}(f, \bar{f}, \bar{\bar{f}})} \quad (5.40)$$

5.3. Inversion de la matrice de corrélation par le réseau de Zhang

5.3.1. Réseau de neurones à dynamique explicite

5.3.1.1. Principe

Soit $\mathbf{R} \in \mathbb{C}^{M \times M}$ la matrice à inverser, le réseau de Zhang est conçu de façon à résoudre l'équation suivante [92-94]:

$$\mathbf{R}\mathbf{X}(t) = \mathbf{I} \quad (5.41)$$

où $\mathbf{X}(t)$ représente l'inverse instantanée de la matrice \mathbf{R} et $\mathbf{I} \in \mathbb{R}^{M \times M}$ dénote la matrice identité. La matrice $\mathbf{X}(t)$ converge vers la matrice théorique \mathbf{R}^{-1} quand le réseau atteint un état stable. En prenant la dérivée de la matrice-erreur $\boldsymbol{\varepsilon}(t) = \mathbf{R}\mathbf{X}(t) - \mathbf{I}$ à minimiser :

$$\dot{\boldsymbol{\varepsilon}}(t) = \frac{d\boldsymbol{\varepsilon}(t)}{dt} = -\boldsymbol{\Gamma}\mathbf{F}_c(\boldsymbol{\varepsilon}(t)) \quad (5.42)$$

où $\boldsymbol{\Gamma} \in \mathbb{R}^{M \times M}$ est une matrice des paramètres utilisée pour régler la cadence de la convergence du réseau et $\mathbf{F}_c(\cdot)$ dénote la matrice-fonction d'activation.

Dans l'expression (5.41), les matrices \mathbf{R} et \mathbf{I} sont fixes, d'où :

$$\dot{\boldsymbol{\varepsilon}}(t) = \mathbf{R}\dot{\mathbf{X}}(t) \quad (5.43)$$

En remplaçant cette expression ainsi que l'expression de $\boldsymbol{\varepsilon}(t)$ dans (5.42), on obtient :

$$\mathbf{R}\dot{\mathbf{X}}(t) = -\boldsymbol{\Gamma}\mathbf{F}_c(\mathbf{R}\mathbf{X}(t) - \mathbf{I}) \quad (5.44)$$

Pour assurer que tous les éléments de la matrice $\mathbf{X}(t)$ converge avec la même cadence et en même temps, il suffit que les valeurs propres de la matrice $\boldsymbol{\Gamma}$ soient proches. Pour la simplification, on prend $\boldsymbol{\Gamma} = \gamma\mathbf{I}$ avec $\eta > 0$. Cette proposition nous conduit à :

$$\mathbf{R}\dot{\mathbf{X}}(t) = -\gamma\mathbf{F}_c(\mathbf{R}\mathbf{X}(t) - \mathbf{I}) \quad (5.45)$$

5.3.1.2. Discrétisation du réseau

La version discrète du réseau de Zhang est obtenue en utilisant la méthode d'Euler :

$$\dot{\mathbf{X}}(t) = (\mathbf{X}(k+1) - \mathbf{X}(k)) / h \quad (5.46)$$

où h dénotes le pas d'échantillonnage, $\mathbf{X}(k)$ est le $k^{\text{ème}}$ échantillon de $\mathbf{X}(t)$ pris à l'instant $t = kh$. Le modèle discret du réseau de Zhang est donné par:

$$\mathbf{RX}(k+1) = \mathbf{RX}(k) - \tau F_c(\mathbf{RX}(k) - \mathbf{I}) \quad (5.47)$$

avec $\tau = \gamma h > 0$.

Par l'hypothèse que matrice \mathbf{R} soit non singulière, c.-à-d. que \mathbf{R}^{-1} existe, le modèle discret de Zhang peut être réécrit sous la forme :

$$\mathbf{X}(k+1) = \mathbf{X}(k) - \tau \mathbf{R}^{-1} F_c(\mathbf{RX}(k) - \mathbf{I}) \quad (5.48)$$

Comme \mathbf{R}^{-1} est inconnue, et $\mathbf{X}(k)$ peut être très proche de \mathbf{R}^{-1} après quelque itérations, dans ces conditions \mathbf{R}^{-1} dans l'équation (5.48) peut être remplacée par $\mathbf{X}(k)$. Par conséquent, on peut réécrire l'équation (5.48) comme [23]:

$$\mathbf{X}(k+1) = \mathbf{X}(k) - \tau \mathbf{X}(k) F_c(\mathbf{RX}(k) - \mathbf{I}) \quad (5.49)$$

Pour assurer la convergence, le réseau doit être initialisé par [95]:

$$\mathbf{X}(0) = \frac{2}{\text{Trace}(\mathbf{R})} \mathbf{I} \quad (5.50)$$

avec $\text{Trace}(\mathbf{R})$ désigne la somme des éléments de la diagonale de la matrice \mathbf{R} .

5.3.1.3. Implémentation du réseau RNZD1

Pour l'implémentation du réseau de Zhang, on introduit des modifications sur l'équation (5.49) en se basant sur le produit de Kronecker \otimes et l'opérateur de vectorisation $\text{vec}(\mathbf{X})$. L'opérateur $\text{vec}(\mathbf{X})$ est un vecteur obtenu par l'empilement de toutes les colonnes de la matrice \mathbf{X} . Appliquant une vectorisation sur l'équation (5.49), on obtient [32]:

$$\text{vec}(\mathbf{X}(k+1) - \mathbf{X}(k)) = \text{vec}(\mathbf{X}(k+1)) - \text{vec}(\mathbf{X}(k)) = -\tau \text{vec}(\mathbf{X}(k) F_c(\mathbf{RX}(k) - \mathbf{I})) \quad (5.51)$$

En utilisant la propriété de la vectorisation d'un produit de deux matrices:

$$\text{vec}(\mathbf{RX}) = (\mathbf{I} \otimes \mathbf{R}) \text{vec}(\mathbf{X}) \quad (5.52)$$

L'équation (5.51) devient:

$$\text{vec}(\mathbf{X}(k+1)) - \text{vec}(\mathbf{X}(k)) = -\tau (\mathbf{I} \otimes \mathbf{X}(k)) \text{vec}(F_c(\mathbf{RX}(k) - \mathbf{I})) \quad (5.53)$$

$$\begin{aligned} \text{vec}(\mathbf{X}(k+1)) - \text{vec}(\mathbf{X}(k)) &= -\tau(\mathbf{I} \otimes \mathbf{X}(t))F_c(\text{vec}(\mathbf{R}\mathbf{X}(t)) + \text{vec}(-\mathbf{I})) \\ &= -\tau(\mathbf{I} \otimes \mathbf{X}(t))F_c((\mathbf{I} \otimes \mathbf{R})\text{vec}(\mathbf{X}(t)) - \text{vec}(\mathbf{I})) \end{aligned} \quad (5.54)$$

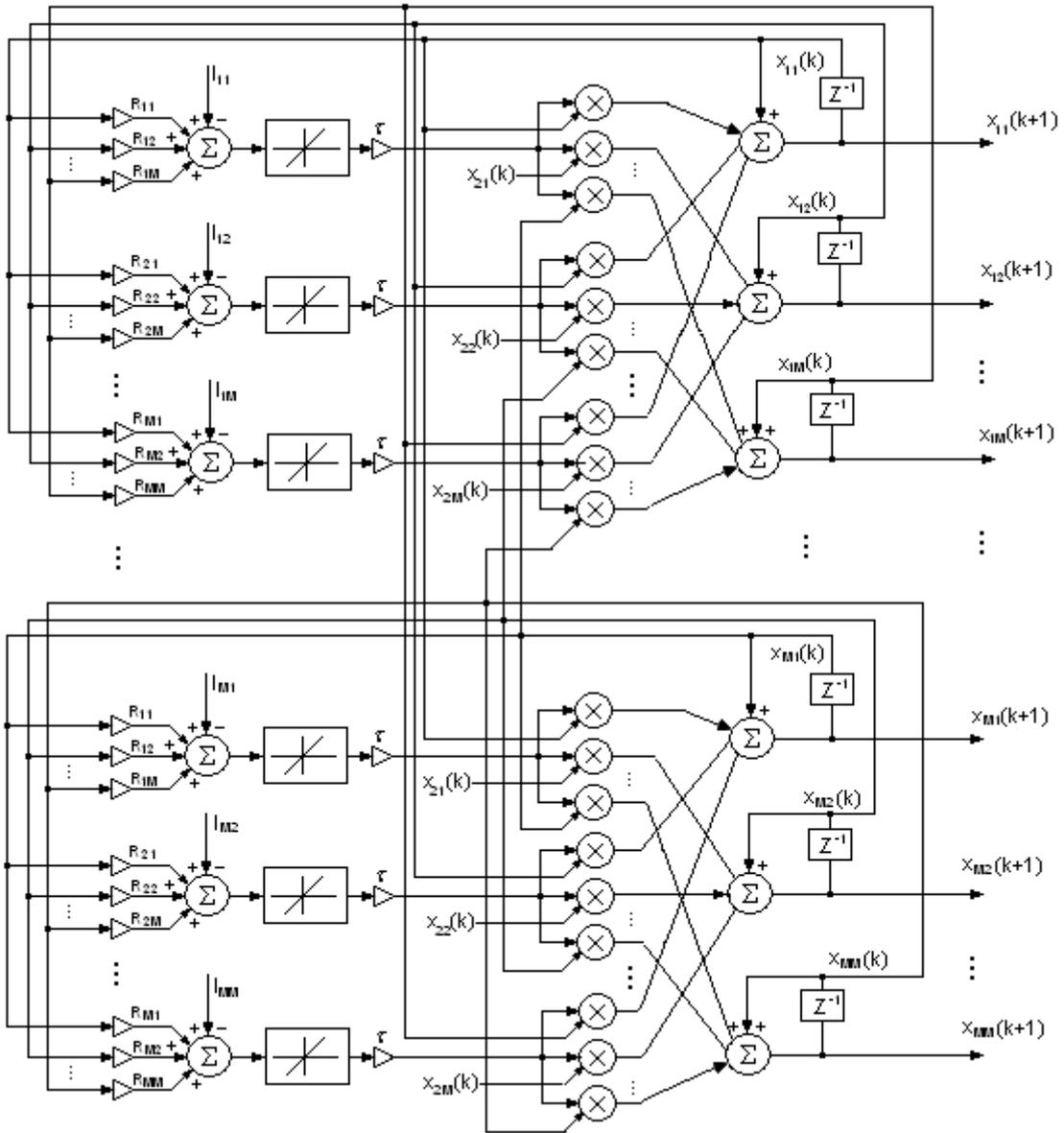


Figure 5.2. Architecture du réseau RNZD1.

L'équation (5.54) nous permet d'implémenter aisément le modèle discret du réseau de Zhang décrit par l'équation (5.49) et représenté sur la figure (5.2). On remarque que le réseau RNZD1 est composé de M sous réseaux similaires dont chacun est composé M neurones. Si on définit la complexité du réseau comme le nombre d'opérateurs utilisés, on constate que le réseau contient $2M^3 + M^2$ multiplicateurs, $2M^2$ additionneurs, M^2 opérateurs de retard et M^2 fonctions d'activation. Comme tous les sous blocs opèrent simultanément, donc le temps de réponse du réseau est celui du sous réseau. De plus, les neurones eux aussi sont parallèles, ceci conduit à dire que le temps de réponse du réseau est égal au temps de propagation de

l'entrée vers la sortie à travers une fonction d'activation, $2M$ multiplications et 2 additions de M entrées.

5.3.2. Réseau de neurones à dynamique implicite

5.3.2.1. Principe

Dans cette section l'inversion de la matrice \mathbf{X} est équivalente à la solution d'un système de M équations. Si on désigne $\mathbf{x}^i = [x_1^i, x_2^i, \dots, x_{p+1}^i]^T$ le i^{me} colonne de la matrice \mathbf{X} et \mathbf{I}^i le i^{me} colonne de la matrice \mathbf{I} , l'équation (5.41) peut être réécrite sous la forme:

$$\mathbf{R}\mathbf{x}^i = \mathbf{I}^i, \quad i = 1, \dots, M \quad (5.55)$$

On voit que l'équation (5.55) est similaire à l'équation (4.24). Donc on peut déduire que le réseau de Zhang pour l'inversion de la matrice est composé de M sous réseaux similaires dont chacun a pour rôle de calculer une seule colonne de la matrice inverse cherchée. Chaque sous-réseau est basé sur la solution de l'équation $\mathbf{R}\mathbf{x}^i - \mathbf{I}^i = \mathbf{0}$.

En suivant les mêmes démarches que dans la section (4.3.1), la dynamique de chaque réseau est régie par :

$$\mathbf{R}\dot{\mathbf{x}}^i(t) = -\gamma \mathbf{F}_c(\mathbf{R}\mathbf{x}^i(t) - \mathbf{I}^i) \quad (5.56)$$

5.3.2.2. Implémentation du réseau RNZD2

A partir de l'équation (5.56), on peut exprimer l'équation dynamique du réseau sous la forme implicite :

$$\dot{x}_j^i(t) = -\gamma f\left(\sum_{k=1}^{p+1} r_{jk} x_k^i(t) - I_j\right) - \left(\sum_{k=1, k \neq j}^n r_{jk} \dot{x}_k^i(t)\right) + (1 - r_{jj})\dot{x}_j^i(t) \quad (5.57)$$

où $x_j^i(t)$ désigne le j^{me} élément de la i^{me} colonne de la matrice $\mathbf{X}(t)$.

La version discrète du réseau de Zhang RNZD2 est obtenue on remplaçant $\dot{\mathbf{x}}^i(t)$ de l'équation (5.56) par $(\mathbf{x}^i(k+1)h - \mathbf{x}^i(kh))/h$.

L'architecture du réseau est illustrée dans la figure (5.3). On remarque aussi que RNZD2 est composé de M sous réseaux similaires. Chaque sous réseau contient $2M^2 + 2M$ multiplicateurs, $2M$ additionneurs de M données, $2M$ additionneurs de 2 données, M opérateurs de retard et M fonctions d'activation. Pour la totalité du réseau il suffit de multiplier par M . Le temps de réponse du réseau est égal au temps de propagation

de l'entrée vers la sortie à travers une fonction d'activation, $2M$ multiplicateurs, 2 additionneurs de M entrées et 2 additionneurs de 2 entrées.

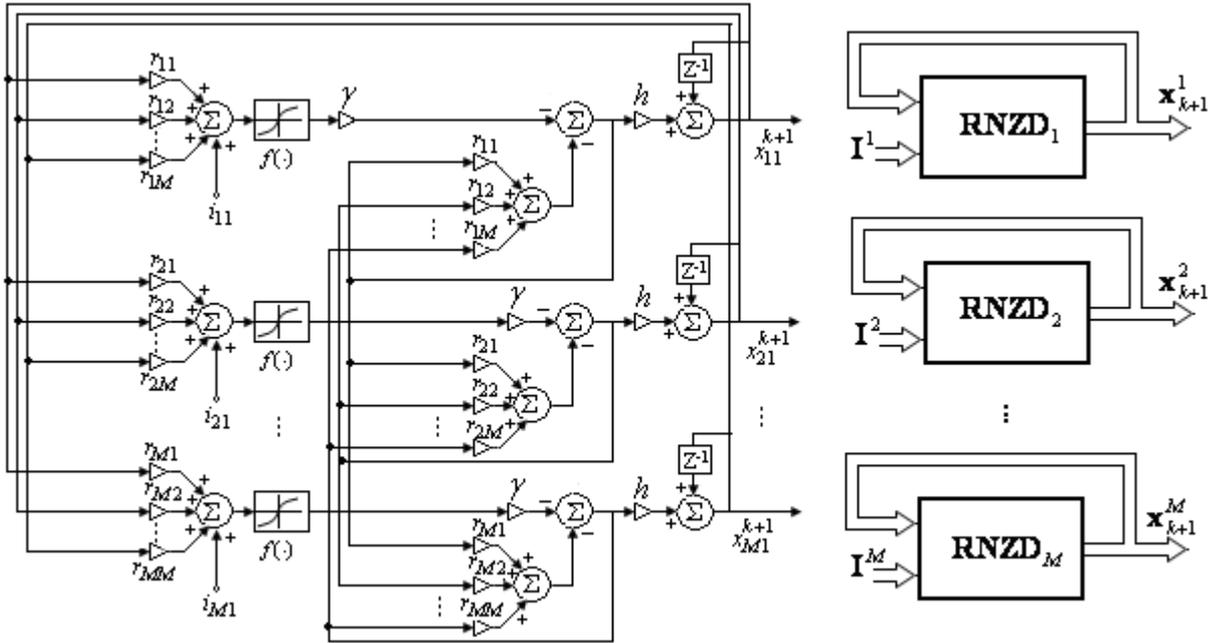


Figure 5.3. Architecture du réseau RNZD2.

5.4. Architecture de l'estimateur spectral

En partant de l'expression de la densité spectrale obtenue par la méthode de Capon, on peut construire le modèle de l'estimateur neuronal en développant le numérateur et le dénominateur de l'expression (1.99).

$$\text{Soit: } \mathbf{R}^{-1} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1M} \\ x_{21} & x_{22} & \cdots & x_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ x_{M1} & x_{M2} & \cdots & x_{MM} \end{bmatrix} \quad (5.58)$$

Suite à la structure du réseau RNZD1, on calcule le produit $\mathbf{R}^{-1} \mathbf{a}_M(f)$ puis on le multiplie par $\mathbf{a}_M^H(f)$. On aura :

$$\mathbf{a}_M^H(f) \mathbf{R}^{-1} \mathbf{a}_M(f) = \begin{bmatrix} 1 & e^{-j2\pi fT} & \cdots & e^{-j2\pi fM} \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1M} \\ x_{21} & x_{22} & \cdots & x_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ x_{M1} & x_{M2} & \cdots & x_{MM} \end{bmatrix} \begin{bmatrix} 1 \\ e^{j2\pi f} \\ \vdots \\ e^{j2\pi fM} \end{bmatrix} \quad (5.59)$$

$$\mathbf{a}_M^H(f) \mathbf{R}^{-1} \mathbf{a}_M(f) = \left[FFT(\mathbf{C}_1^X) \quad FFT(\mathbf{C}_2^X) \quad \cdots \quad FFT(\mathbf{C}_M^X) \right] \cdot \begin{bmatrix} 1 \\ e^{j2\pi f} \\ \vdots \\ e^{j2\pi fM} \end{bmatrix} \quad (5.60)$$

$$\mathbf{a}_M^H(f) \mathbf{R}^{-1} \mathbf{a}_M(f) = FFT(\mathbf{C}_1^X) + FFT(\mathbf{C}_2^X) \cdot e^{j2\pi f} \Big|_{f=\frac{k}{F}} + \dots + FFT(\mathbf{C}_M^X) \cdot e^{j2\pi f M} \Big|_{f=\frac{k}{F}} \quad (5.61)$$

pour $k = 0, 1, \dots, F-1$, F est le nombre de points sur lesquels est évaluée la FFT.

Cette équation est constituée de la transformée de Fourier discrète de chaque colonne de la matrice \mathbf{R}^{-1} qu'on multiplie par des phaseurs.

Pour l'implémentation du numérateur, on procède comme suit :

$$\mathbf{g}(f) = \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(M-1) \end{bmatrix} e^{-j2\pi f 0} + \begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(M) \end{bmatrix} e^{-j2\pi f} + \dots + \begin{bmatrix} x(L-1) \\ x(L) \\ \vdots \\ x(N-1) \end{bmatrix} e^{-j2\pi f(L-1)} \quad (5.62)$$

$$\mathbf{g}(f) = \begin{bmatrix} x(0)e^{-j2\pi f 0} + x(1)e^{-j2\pi f} + \dots + x(L-1)e^{-j2\pi f(L-1)} \\ x(1)e^{-j2\pi f 0} + x(2)e^{-j2\pi f} + \dots + x(L)e^{-j2\pi f(L-1)} \\ \vdots \\ x(M-1)e^{-j2\pi f 0} + x(M)e^{-j2\pi f} + \dots + x(N-1)e^{-j2\pi f(L-1)} \end{bmatrix} = \begin{bmatrix} FFT\{x(i)\}_{i=0}^{L-1} \\ FFT\{x(i)\}_{i=1}^L \\ \vdots \\ FFT\{x(i)\}_{i=M-1}^{N-1} \end{bmatrix} \quad (5.63)$$

On constate que $\mathbf{g}(f)$ est un vecteur constitué de la FFT des segments de longueur L .

$$\mathbf{a}_M^H(f) \mathbf{R}^{-1} \mathbf{g}(f) = \begin{bmatrix} 1 & e^{-j2\pi f} & \dots & e^{-j2\pi f M} \end{bmatrix} \cdot \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1M} \\ x_{21} & x_{22} & \dots & x_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ x_{M1} & x_{M2} & \dots & x_{MM} \end{bmatrix} \cdot \begin{bmatrix} FFT\{x(i)\}_{i=0}^{L-1} \\ FFT\{x(i)\}_{i=1}^L \\ \vdots \\ FFT\{x(i)\}_{i=M-1}^{N-1} \end{bmatrix} \quad (5.64)$$

$$\mathbf{a}_M^H(f) \mathbf{R}^{-1} \mathbf{g}(f) = \begin{bmatrix} FFT(\mathbf{C}_1^X) & FFT(\mathbf{C}_2^X) & \dots & FFT(\mathbf{C}_M^X) \end{bmatrix} \cdot \begin{bmatrix} FFT\{x(i)\}_{i=0}^{L-1} \\ FFT\{x(i)\}_{i=1}^L \\ \vdots \\ FFT\{x(i)\}_{i=M-1}^{N-1} \end{bmatrix} \quad (5.65)$$

$$\mathbf{a}_M^H(f) \mathbf{R}^{-1} \mathbf{g}(f) = FFT(\mathbf{C}_1^X) \circ FFT\{x(i)\}_{i=0}^{L-1} + FFT(\mathbf{C}_2^X) \circ FFT\{x(i)\}_{i=1}^L + \dots + FFT(\mathbf{C}_M^X) \circ FFT\{x(i)\}_{i=M-1}^{N-1} \quad (5.66)$$

où \mathbf{C}_i^X est le $i^{\text{ème}}$ colonne de la matrice \mathbf{X} et \circ désigne le produit élément par élément.

Les transformées de Fourier des colonnes \mathbf{C}_i^X sont déjà calculées dans la partie du dénominateur, donc il ne faut qu'à calculer la transformée de Fourier des segments \mathbf{y}_i puis faire la multiplication point par point. La figure (5.4) illustre l'architecture du l'estimateur proposé utilisant le réseau RNZD2. Si on remplace le bloc RNZD2 par le bloc RNZD1, on obtient le deuxième estimateur.

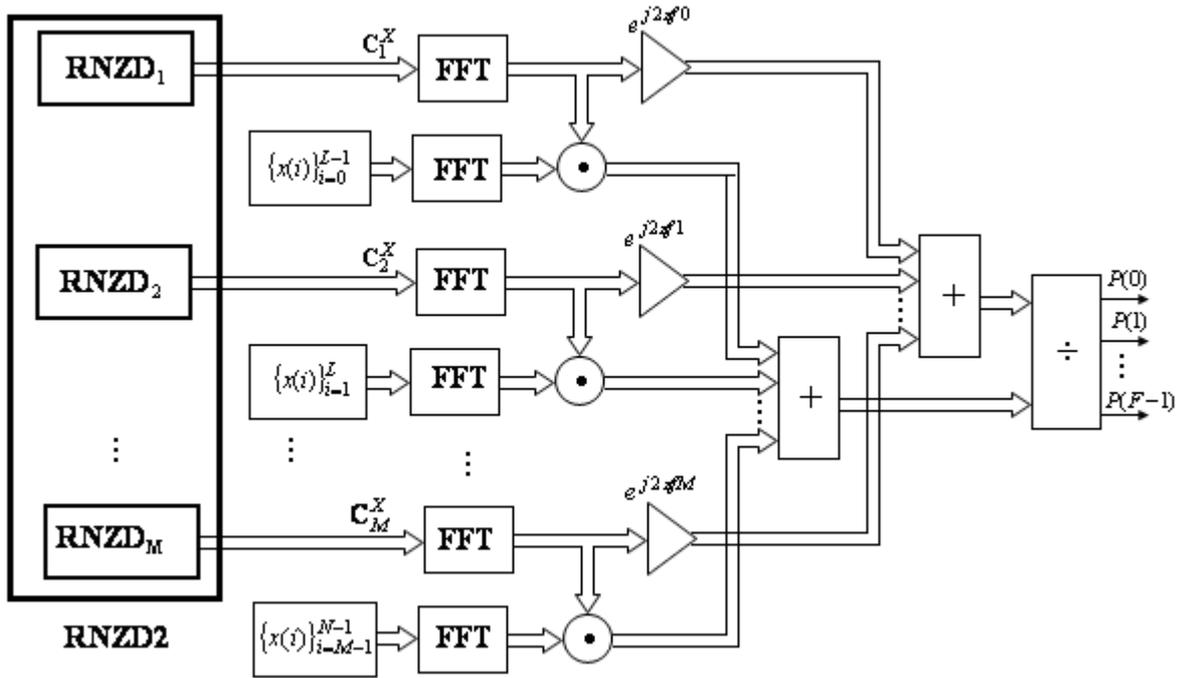


Figure 5.4. Schéma synoptique de l'estimateur neuronal basé sur le réseau RNZD2

5.5. Implémentation de l'estimateur proposé sous Simulink

Dans les deux architectures proposées, les poids ainsi que l'état du réseau sont des valeurs complexes, une autre stratégie consiste à concevoir un réseau à valeurs réelles où les données sont divisées en parties réelles et en parties imaginaires.

5.5.1. Estimateur spectral via RNZD1

Partant de l'équation (5.49) pour le réseau à dynamique explicite, et du fait que:

$$\mathbf{X}(k) = \mathbf{X}^{\text{Re}}(k) + j\mathbf{X}^{\text{Im}}(k) \quad (5.67)$$

$$\mathbf{R} = \mathbf{R}^{\text{Re}} + j\mathbf{R}^{\text{Im}} \quad (5.68)$$

L'équation (5.49) devient :

$$\begin{aligned} \mathbf{X}^{\text{Re}}(k+1) + j\mathbf{X}^{\text{Im}}(k+1) &= \mathbf{X}^{\text{Re}}(k) + j\mathbf{X}^{\text{Im}}(k) \\ &\quad - \tau(\mathbf{X}^{\text{Re}}(k) + j\mathbf{X}^{\text{Im}}(k))F_c\left((\mathbf{R}^{\text{Re}} + j\mathbf{R}^{\text{Im}})(\mathbf{X}^{\text{Re}}(k) + j\mathbf{X}^{\text{Im}}(k)) - \mathbf{I}\right) \end{aligned} \quad (5.69)$$

$$\begin{aligned} \mathbf{X}^{\text{Re}}(k+1) + j\mathbf{X}^{\text{Im}}(k+1) &= \mathbf{X}^{\text{Re}}(k) + j\mathbf{X}^{\text{Im}}(k) - \tau(\mathbf{X}^{\text{Re}}(k) + j\mathbf{X}^{\text{Im}}(k)) \\ &\quad \cdot F_c\left(\mathbf{R}^{\text{Re}}\mathbf{X}^{\text{Re}}(k) - \mathbf{R}^{\text{Im}}\mathbf{X}^{\text{Im}}(k) - \mathbf{I} + j(\mathbf{R}^{\text{Re}}\mathbf{X}^{\text{Im}}(k) + \mathbf{R}^{\text{Im}}\mathbf{X}^{\text{Re}}(k))\right) \end{aligned} \quad (5.70)$$

En utilisant la propriété de la fonction d'activation pour les données complexes [72-75]

$$F_c(\mathbf{R}^{\text{Re}} + j\mathbf{R}^{\text{Im}}) = F_{\text{Re}}(\mathbf{R}^{\text{Re}}) + jF_{\text{Re}}(\mathbf{R}^{\text{Im}}) \quad (5.71)$$

On aura :

$$F_c(\mathbf{R}^{\text{Re}} \mathbf{X}^{\text{Re}}(k) - \mathbf{R}^{\text{Im}} \mathbf{X}^{\text{Im}}(k) + j(\mathbf{R}^{\text{Re}} \mathbf{X}^{\text{Im}}(k) + \mathbf{R}^{\text{Im}} \mathbf{X}^{\text{Re}}(k)) - \mathbf{I}) = F_{\text{Re}}(\mathbf{R}^{\text{Re}} \mathbf{X}^{\text{Re}}(k) - \mathbf{R}^{\text{Im}} \mathbf{X}^{\text{Im}}(k)) + jF_{\text{Re}}(\mathbf{R}^{\text{Re}} \mathbf{X}^{\text{Im}}(k) + \mathbf{R}^{\text{Im}} \mathbf{X}^{\text{Re}}(k)) \quad (5.72)$$

En posant :

$$\begin{aligned} \mathbf{A}^{\text{Re}} &= F_{\text{Re}}(\mathbf{R}^{\text{Re}} \mathbf{X}^{\text{Re}}(k) - \mathbf{R}^{\text{Im}} \mathbf{X}^{\text{Im}}(k) - \mathbf{I}) \\ \mathbf{A}^{\text{Im}} &= F_{\text{Re}}(\mathbf{R}^{\text{Re}} \mathbf{X}^{\text{Im}}(k) + \mathbf{R}^{\text{Im}} \mathbf{X}^{\text{Re}}(k)) \end{aligned} \quad (5.73)$$

L'équation (5.70) devient

$$\begin{aligned} \mathbf{X}^{\text{Re}}(k+1) + j\mathbf{X}^{\text{Im}}(k+1) &= \mathbf{X}^{\text{Re}}(k) + j\mathbf{X}^{\text{Im}}(k) - \tau(\mathbf{X}^{\text{Re}}(k) + j\mathbf{X}^{\text{Im}}(k))(\mathbf{A}^{\text{Re}} + j\mathbf{A}^{\text{Im}}) \\ &= \mathbf{X}^{\text{Re}}(k) + j\mathbf{X}^{\text{Im}}(k) - \tau(\mathbf{X}^{\text{Re}}(k)\mathbf{A}^{\text{Re}} - \mathbf{X}^{\text{Im}}(k)\mathbf{A}^{\text{Im}} + j(\mathbf{X}^{\text{Re}}(k)\mathbf{A}^{\text{Im}} + \mathbf{X}^{\text{Im}}(k)\mathbf{A}^{\text{Re}})) \end{aligned} \quad (5.74)$$

En séparant en partie réelle et partie imaginaire, l'équation (5.74) devient :

$$\mathbf{X}^{\text{Re}}(k+1) = \mathbf{X}^{\text{Re}}(k) - \tau(\mathbf{X}^{\text{Re}}(k)\mathbf{A}^{\text{Re}} - \mathbf{X}^{\text{Im}}(k)\mathbf{A}^{\text{Im}}) \quad (5.75)$$

$$\mathbf{X}^{\text{Im}}(k+1) = \mathbf{X}^{\text{Im}}(k) - \tau(\mathbf{X}^{\text{Re}}(k)\mathbf{A}^{\text{Im}} + \mathbf{X}^{\text{Im}}(k)\mathbf{A}^{\text{Re}}) \quad (5.76)$$

La nouvelle architecture du réseau est composée de deux réseaux coopératifs à valeurs réelles décrits par les deux équations (5.75), (5.76) et représentés dans la figure (5.5).

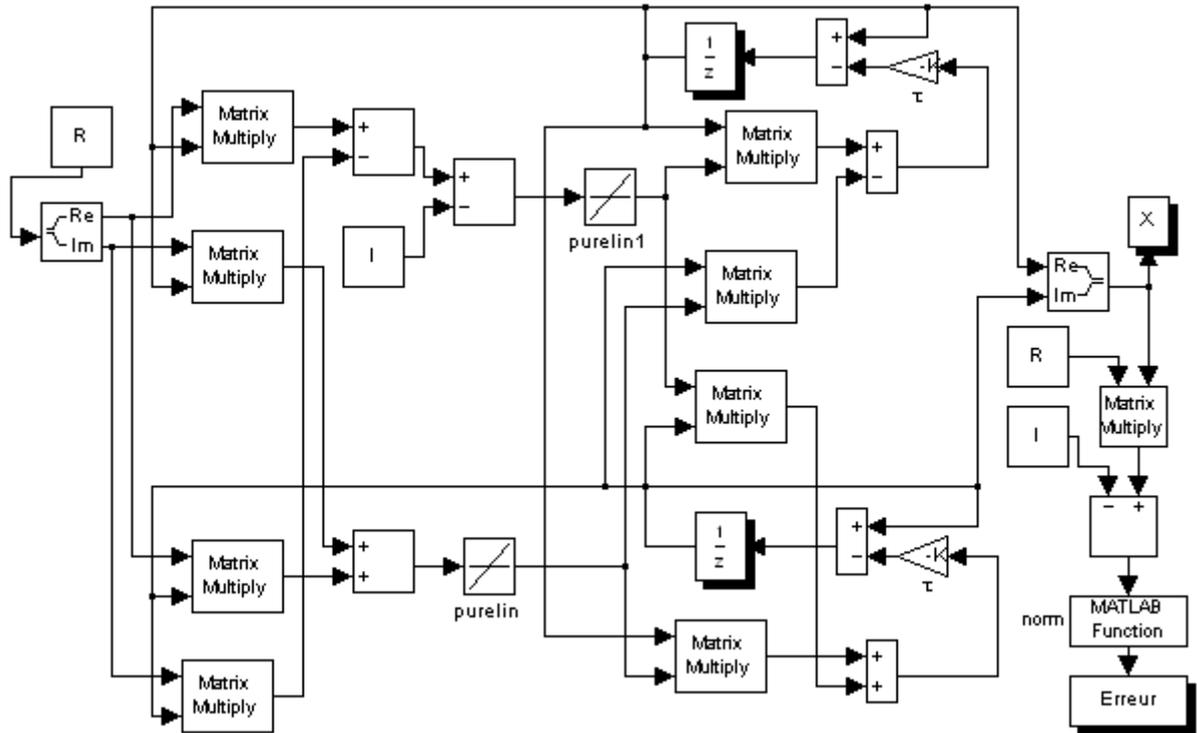


Figure 5.5. Schéma d'implémentation du réseau RNZD1.

5.5.2. Estimateur spectral via RNZD2

Pour le réseau implicite, à partir de l'équation (5.56) et avec la même procédure on obtient:

$$\left(\mathbf{R}^{\text{Re}} + j\mathbf{R}^{\text{Im}}\right)\left(\dot{\mathbf{x}}^{i\text{Re}}(t) + j\dot{\mathbf{x}}^{i\text{Im}}(t)\right) = -\gamma \mathbf{F}_c \left(\left(\mathbf{R}^{\text{Re}} + j\mathbf{R}^{\text{Im}}\right)\mathbf{x}^{i\text{Re}}(t) + j\mathbf{x}^{i\text{Im}}(t)\right) - \mathbf{I}^i \quad (5.77)$$

$$\begin{aligned} &\left(\mathbf{R}^{\text{Re}} \dot{\mathbf{x}}^{i\text{Re}}(t) - \mathbf{R}^{\text{Im}} \dot{\mathbf{x}}^{i\text{Im}}(t)\right) + j\left(\mathbf{R}^{\text{Re}} \dot{\mathbf{x}}^{i\text{Im}}(t) + \mathbf{R}^{\text{Im}} \dot{\mathbf{x}}^{i\text{Re}}(t)\right) = \\ &-\gamma \mathbf{F}_c \left(\left(\mathbf{R}^{\text{Re}} \mathbf{x}^{i\text{Re}}(t) - \mathbf{R}^{\text{Im}} \mathbf{x}^{i\text{Im}}(t) - \mathbf{I}^i\right) + j\left(\mathbf{R}^{\text{Re}} \mathbf{x}^{i\text{Im}}(t) + \mathbf{R}^{\text{Im}} \mathbf{x}^{i\text{Re}}(t)\right)\right) \end{aligned} \quad (5.78)$$

$$\mathbf{R}^{\text{Re}} \dot{\mathbf{x}}^{i\text{Re}}(t) - \mathbf{R}^{\text{Im}} \dot{\mathbf{x}}^{i\text{Im}}(t) = -\gamma \mathbf{F}_{\text{Re}} \left(\mathbf{R}^{\text{Re}} \mathbf{x}^{i\text{Re}}(t) - \mathbf{R}^{\text{Im}} \mathbf{x}^{i\text{Im}}(t) - \mathbf{I}^i\right) \quad (5.79)$$

$$\mathbf{R}^{\text{Re}} \dot{\mathbf{x}}^{i\text{Im}}(t) + \mathbf{R}^{\text{Im}} \dot{\mathbf{x}}^{i\text{Re}}(t) = -\gamma \mathbf{F}_{\text{Im}} \left(\mathbf{R}^{\text{Re}} \mathbf{x}^{i\text{Im}}(t) + \mathbf{R}^{\text{Im}} \mathbf{x}^{i\text{Re}}(t)\right) \quad (5.80)$$

Le modèle discret du réseau représenté dans la figure (5.6) est obtenu par les équations (5.79) et (5.80) après la discrétisation utilisant la formule d'Euler $\dot{\mathbf{x}}(t) = (\mathbf{x}(k+1) - \mathbf{x}(k))/h$ et $\mathbf{x}(t) = \mathbf{x}(k)$.

5.6. Exemples de simulation

Dans cette section, on présente quelque exemple pour montrer la précision et la rapidité de l'inversion de la matrice de corrélation via le réseau de neurones pour estimer le spectre d'amplitudes des signaux 1D, 2D et 3D.

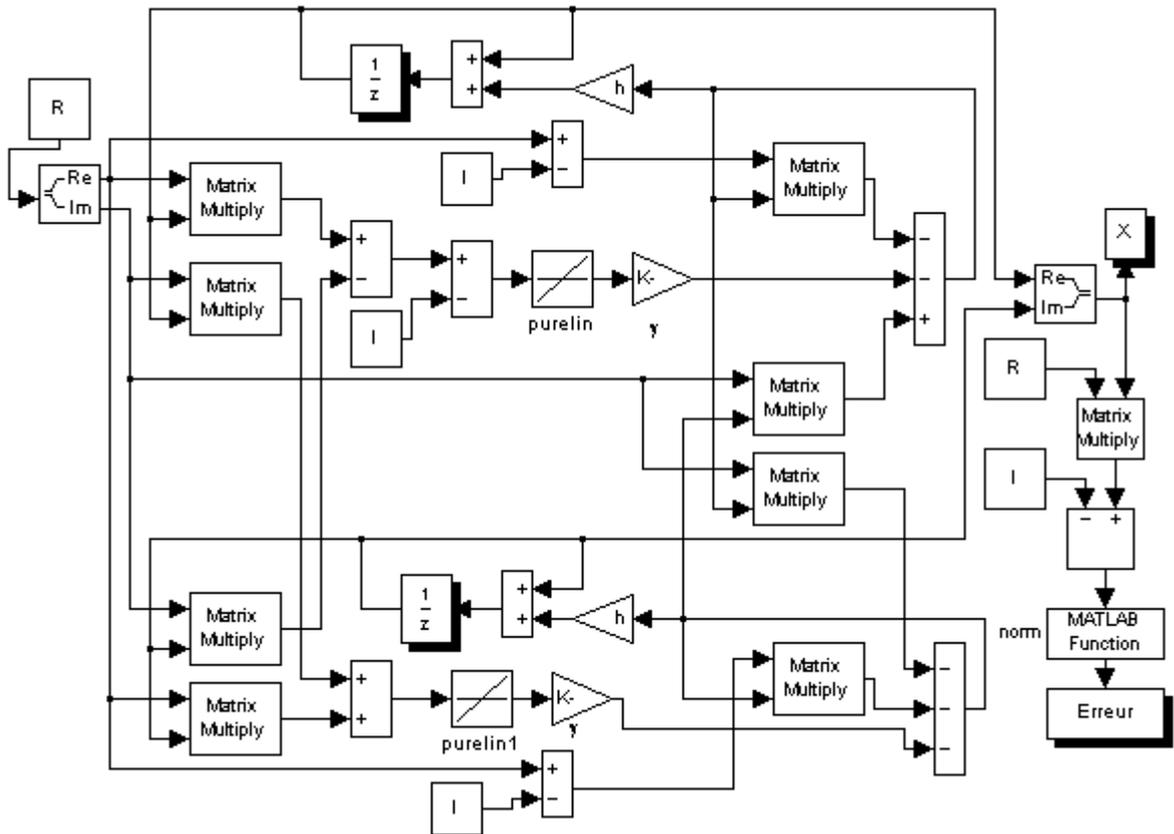


Figure 5.6. Schéma d'implémentation du réseau RNZD2.

5.6.1. Signaux monodimensionnels

La séquence des données utilisée est composée de $N = 128$ échantillons et consiste en quatre sinusoïdes dominantes d'amplitudes 1,1,1,0.3 et neuf autres d'amplitudes 0.1. Ces raies sont localisées aux fréquences 0.0625, 0.0725, 0.25, 0.28, 0.33, 0.35, 0.37, 0.39, 0.41, 0.43, 0.45, 0.47, 0.49 et ayant une phase de $\pi/4$. Ces échantillons sont noyés dans un bruit blanc gaussien, de moyenne nulle et de variance σ^2 . La sévérité du bruit a été mesurée par le SNR.

5.6.1.2. Analyse de la convergence du réseau RNZD

La convergence du réseau RNZD1 est illustrée dans le tableau (5.1). On constate que le temps de convergence du réseau diminue en augmentant le coefficient τ . Mais lorsque τ se rapproche de 1, le réseau diverge. De plus, pour différentes valeurs de M , une même valeur de τ nous donne presque le même nombre d'itérations nécessaires pour la convergence k_c .

M	8		16		32	
	k_c	$\ \mathbf{RX} - \mathbf{I}\ _2 \times 10^{-13}$	k_c	$\ \mathbf{RX} - \mathbf{I}\ _2 \times 10^{-13}$	k_c	$\ \mathbf{RX} - \mathbf{I}\ _2 \times 10^{-13}$
0.1	356	7.4409	359	14.93	373	27.66
0.15	237	4.8568	243	10.97	243	29.41
0.2	180	2.7763	178	8.941	184	25.56
0.25	140	3.3333	143	9.08	145	24.86
0.3	117	3.4951	116	8.7331	123	21.84
0.35	100	2.9240	100	10.07	102	21.15
0.4	87	3.6320	83	11.07	89	22.51
0.45	75	2.8476	75	8.03	78	21.01
0.5	64	2.6690	67	8.091	68	21.82
0.55	59	3.2227	58	9.202	62	23.34
0.6	51	3.2427	53	8.951	57	21.54
0.65	46	2.8909	53	8.961	53	22.77
0.7	42	3.2698	48	9.194	46	23.26
0.75	39	3.2828	39	8.974	45	24.12
0.8	-	-	-	-	39	24.21
0.85	-	-	-	-	36	26.14
0.9	-	-	-	-	32	27.46
0.95	-	-	-	-	29	29.32

Tableau 5.1. Influence du paramètre τ et M sur la convergence du réseau RNZD1.

Toutefois, l'erreur de convergence $\|\mathbf{RX} - \mathbf{I}\|_2$ augmente légèrement avec M . On remarque aussi que plus M augmente, plus le coefficient τ se rapproche de 1 et par conséquent un nombre k_c plus petit. Concernant le réseau RNZD2, on a montré dans chapitre 5 que les performances optimales sont obtenues pour $\tau = \gamma h = 1$.

5.6.1.2. Analyse de la performance du RNZD

Pour évaluer le réseau proposé dans ces deux versions, le spectre du signal a été estimé via la méthode de Capon et la méthode APES en utilisant la méthode directe et les deux réseaux RNZD1 et RNZD2 pour calculer l'inverse de la matrice de corrélation. La figure (5.7) montre le spectre obtenu par les deux méthodes pour $N = 128$, $SNR = 30dB$, et $M = 24$. On remarque que le spectre obtenu en utilisant le RNZD1 ou RNZD2 est similaire à celui obtenu par la méthode directe de Capon ou la méthode directe APES.

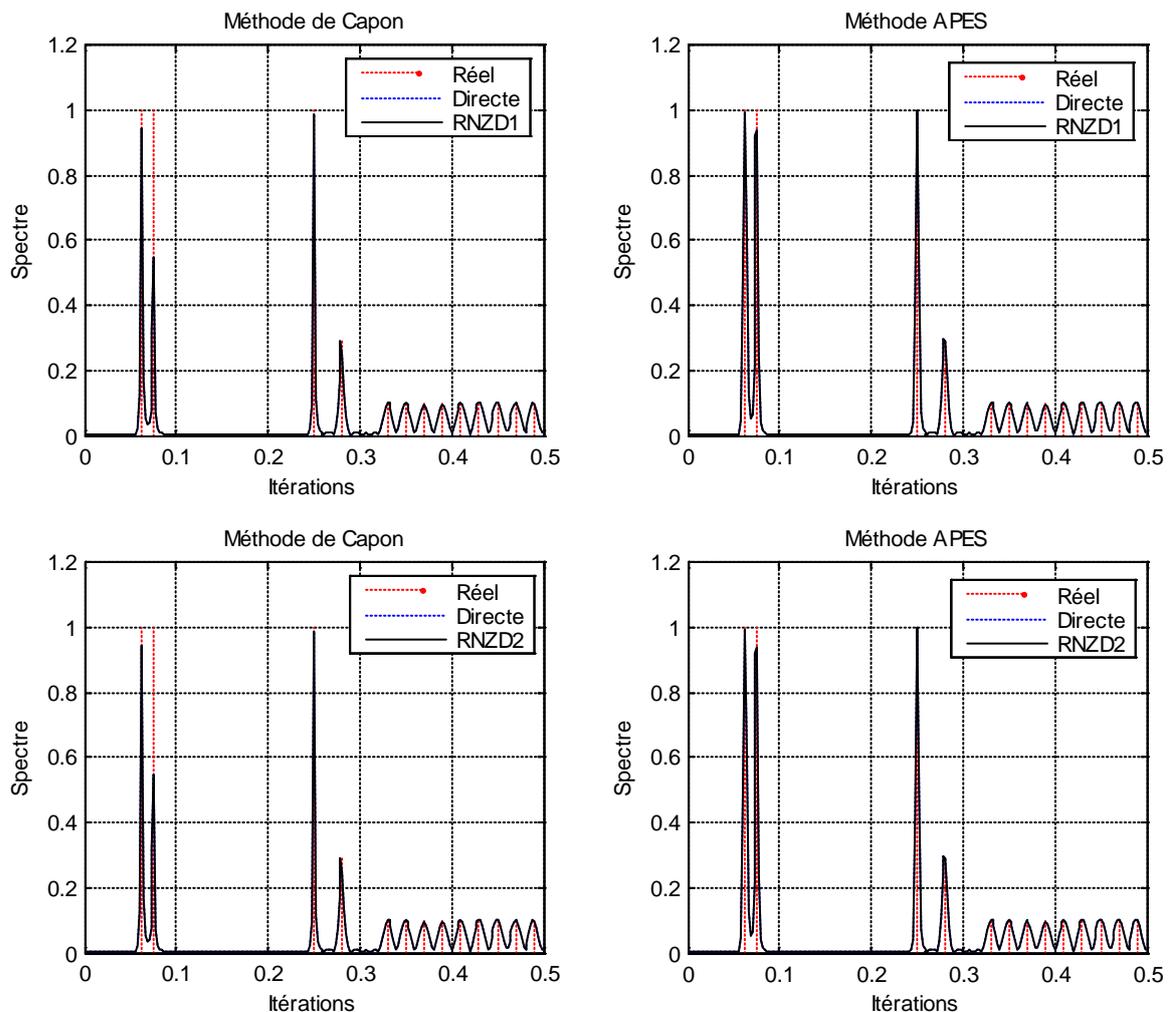


Figure 5.7. Densité spectrale du signal 1D estimée $M = 8$ et $SNR = 30dB$.

Cependant l'avantage d'utiliser le réseau de neurones réside dans le parallélisme de son architecture et par conséquent son temps de réponse réduit. La figure (5.8) montre l'évolution de l'erreur de convergence du réseau RNZD1 pour un $\tau = 0.8$, $h = 10^{-4}$ et celle du réseau RNZD2 pour un $\tau = 1$, $h = 10^{-4}$. On constate que le réseau RNZD2 est plus rapide que le réseau RNZD1, il ne nécessite que deux ou trois itérations pour atteindre une erreur au voisinage de 10^{-13}

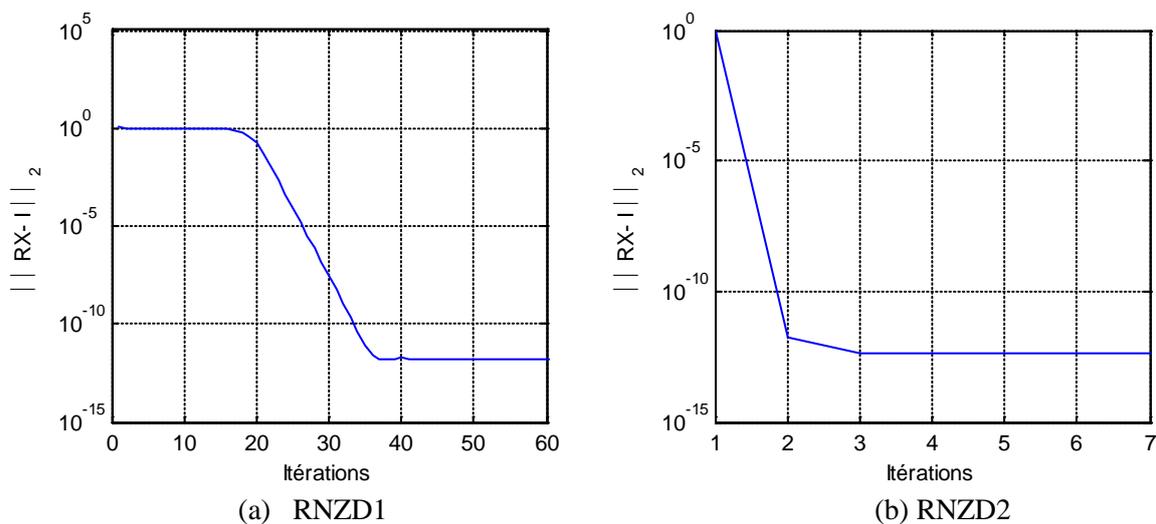


Figure 5.8. Erreur de convergence pour le cas des signaux 1D.

5.6.2. Signaux bidimensionnels

5.6.2.1. Signaux synthétiques

On considère un signal 2D composé de 64×64 échantillons est constitué d'un mélange de trois sinusoïdes d'amplitudes 1, 0.7, 2 et de fréquence (0.1, 0.25), (0.2, 0.35), (0.3, 0.15) respectivement noyées dans un bruit complexe, blanc centré de variance σ^2 . La figure (5.9) montre les deux représentations spatiales et fréquentielle de ce signal.

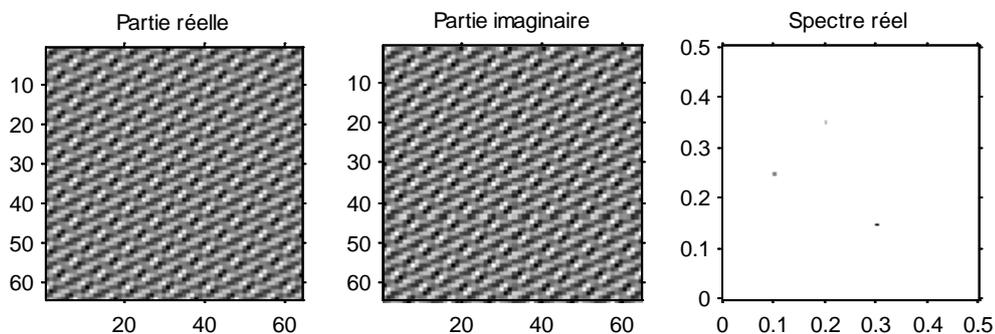


Figure 6.9. Signal de test.

La figure (5.10) montre que le réseau RNZD2 est plus rapide que le réseau RNZD1, il nécessite deux itérations pour atteindre une erreur au voisinage de 10^{-10} . Dans la figure

(5.11), on montre le spectre obtenu par les deux méthodes directes et les deux méthodes basées sur le réseau RNZD pour $M \times \bar{M} = 8 \times 8$ ce qui correspond à une matrice \mathbf{R} de taille 64×64 . On remarque que les spectres obtenus par les méthodes directes et par le réseau RNZD1 sont similaires (les spectres obtenus par RNZD1 et RNZD2 et sont similaires).

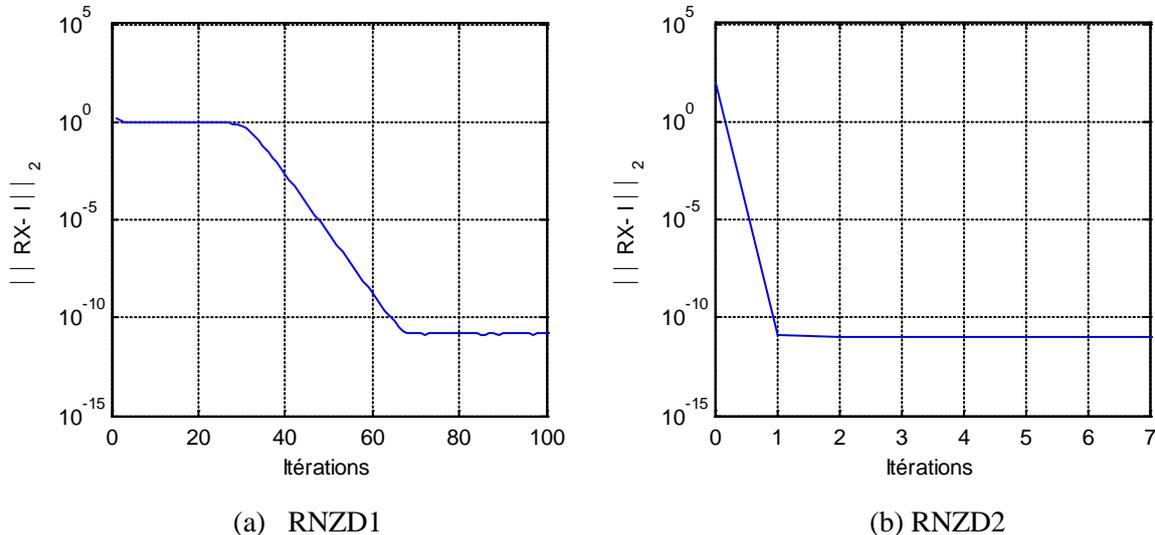


Figure 5.10. Erreur de convergence dans le cas des signaux 2D.

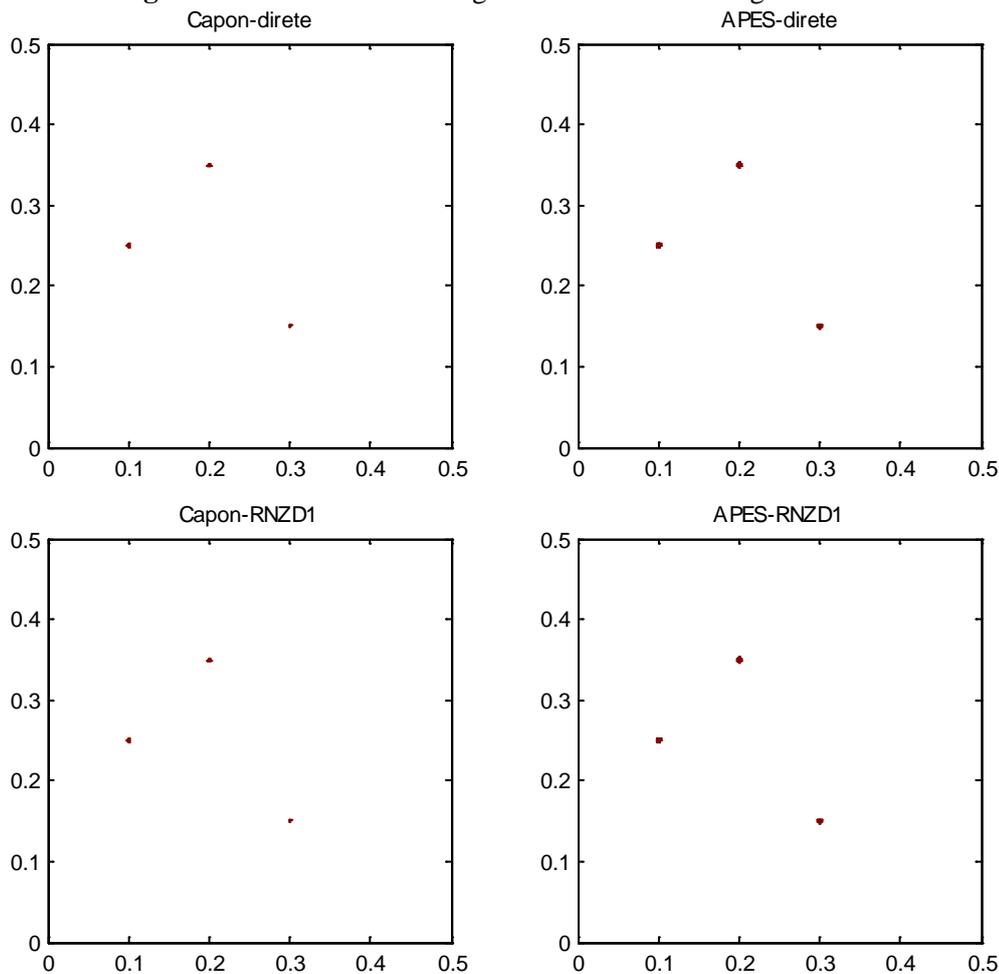


Figure 5.11. Densité spectrale du signal 2D estimée $M \times \bar{M} = 8 \times 8$ et $SNR = 30dB$.

5.6.2.2. Application réelle

Les radars à synthèse d'ouverture sont des radars dont l'antenne est fixée sur une face latérale d'un porteur (avion ou satellite). Elle a une ouverture azimutale assez grande dans la direction du mouvement et latéralement peut aller de l'horizon à la verticale. Comme le radar se déplace, un même point est cependant illuminé plusieurs fois, on obtient une série de données pour chaque point sous le radar. Le signal reçu est donc un point de la transformée de Fourier de l'objet éclairé. Comme le radar se déplace avec son porteur, il reçoit d'autres points de cette transformée. Il suffit d'enregistrer tous ces points et d'en faire ensuite la transformée inverse pour reconstituer une carte de réflectivité de l'objet en deux dimensions [60]. Les données réelles utilisées dans cette simulation sont fournies par le laboratoire de recherche navale (U.S. Naval Research Laboratory) et concernent un avion MIG-25. Elles sont constituées de 64×512 mesures. La figure (5.12) montre le module et la phase d'une portion de 64×64 de cette image et sa transformée de Fourier.

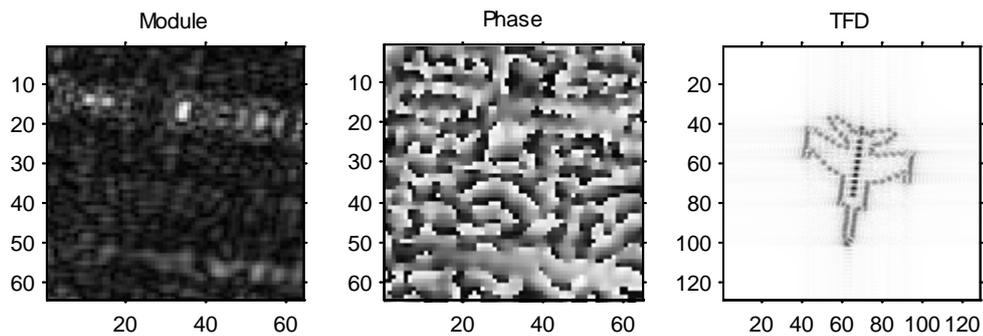


Figure 5.12. Module, phase et spectre des données de l'avion MIG-25.

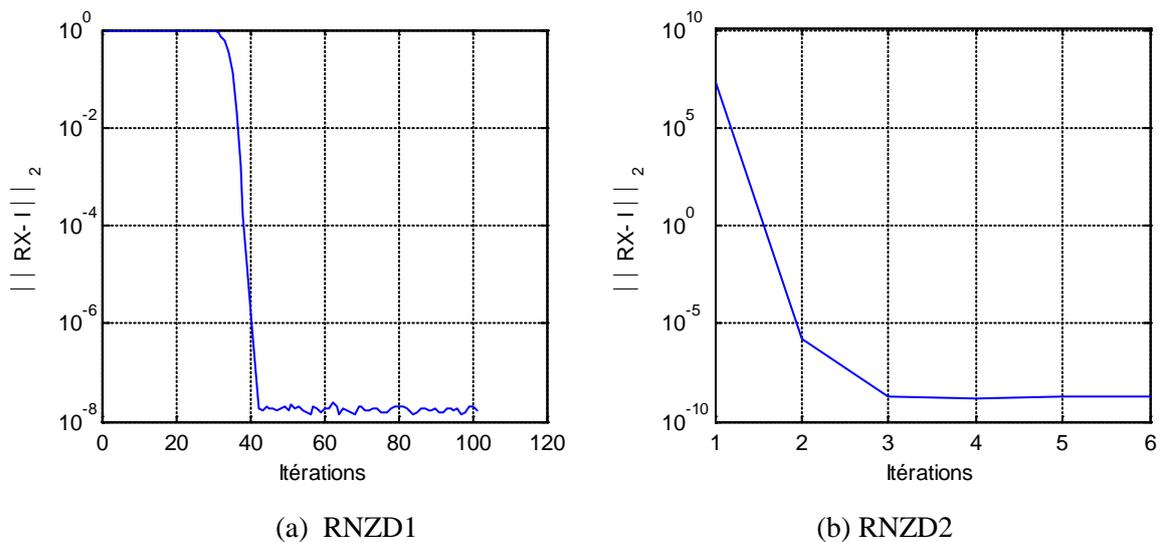


Figure 5.13. Erreur de convergence pour le cas des signaux 2D réels.

La figure (5.13) montre l'évolution de l'erreur de convergence du réseau RNZD1 et RNZD2, on constate que le réseau RNZD2 est plus rapide que le réseau RNZD1, il ne nécessite que trois itérations pour atteindre une erreur au voisinage de 10^{-9} . La figure (5.14) montre l'image reconstruite par la méthode de Capon2D, la méthode APES2D et par les deux réseaux évalués sur 256×256 points. La taille du filtre utilisé de 4×4 .

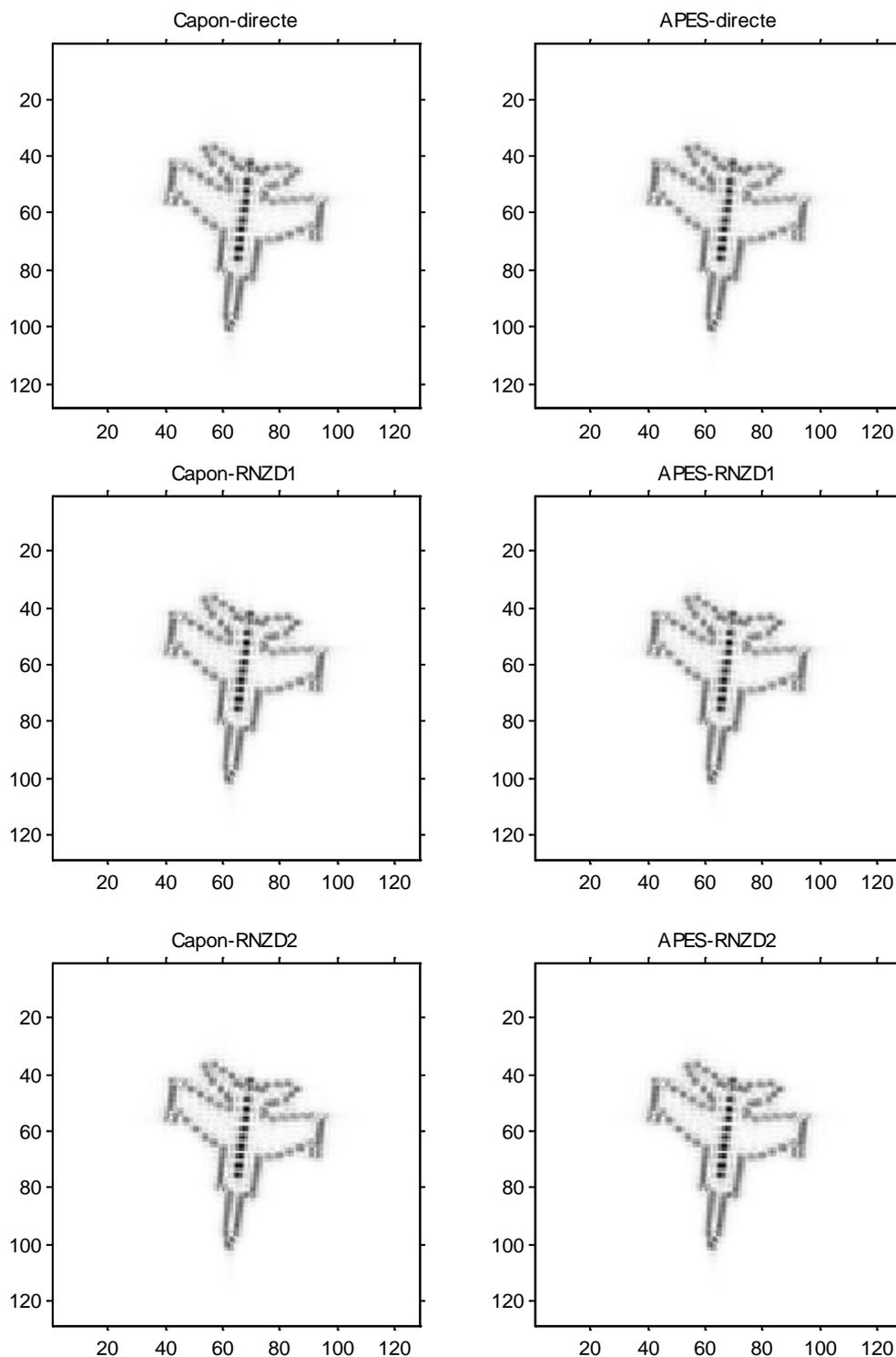


Figure 5.14. Image reconstruite avec un filtre $M \times \bar{M} = 4 \times 4$.

On remarque d'après ces figures que les images reconstruites sont similaires. On note ici que pour une plus grande taille du filtre, le réseau RNZD2 diverge suite au mal conditionnement de la matrice.

5.6.3. Signaux tridimensionnels

On considère un signal 3D composé de $16 \times 16 \times 16$ échantillons est constitué d'un mélange de trois sinusoides d'amplitudes unitaires et de fréquence $(0.031, 0.203, 0.156)$, $(0.125, 0.14, 0.078)$, $(0.203, 0.094, 0.203)$ respectivement noyées dans un bruit complexe, blanc centré de variance σ^2 . Les fréquences estimées utilisant les différentes méthodes sont montrées dans le tableau (5.1). On remarque que les deux réseaux RNZD1 et RNZD2 donnent les mêmes performances.

	$f_x = f_k$	$f_y = \bar{f}_k$	$f_z = \bar{\bar{f}}_k$
Fréquences réelles	0.031	0.203	0.156
	0.125	0.14	0.078
	0.203	0.094	0.203
Capon directe	0.03125	0.20313	0.15625
	0.11719	0.14063	0.078125
	0.20313	0.09375	0.20313
APES-directe	0.03125	0.20313	0.15625
	0.125	0.14063	0.078125
	0.20313	0.09375	0.20313
Capon-RNZD1	0.03125	0.20313	0.15625
	0.11719	0.14063	0.078125
	0.20313	0.09375	0.20313
APES-RNZD1	0.03125	0.20313	0.15625
	0.125	0.14063	0.078125
	0.20313	0.09375	0.20313
Capon-RNZD2	0.03125	0.20313	0.15625
	0.11719	0.14063	0.078125
	0.20313	0.09375	0.20313
APES-RNZD2	0.03125	0.20313	0.15625
	0.125	0.14063	0.078125
	0.20313	0.09375	0.20313

Tableau 5.2. Fréquences estimées pour les quatre méthodes avec $N = 16$, $M = \bar{M} = \bar{\bar{M}} = 8$.

Les figures (5.15) et (5.16) montrent le spectre 3D en utilisant les projections dans les plans $f_x - f_y$, $f_x - f_z$ et $f_y - f_z$ obtenu par la méthode de CAPON et la méthode APES utilisant le réseau RNZD2 respectivement.

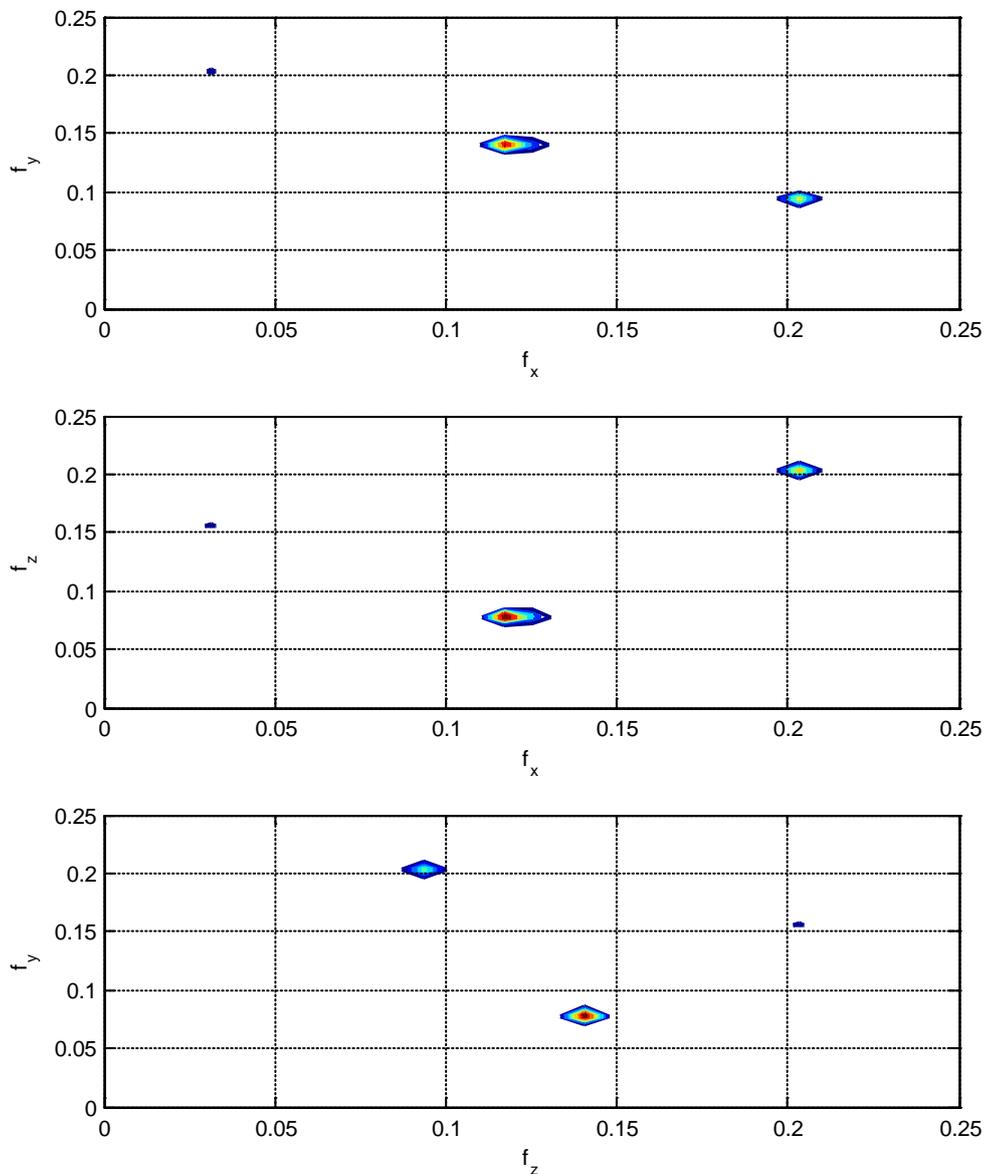
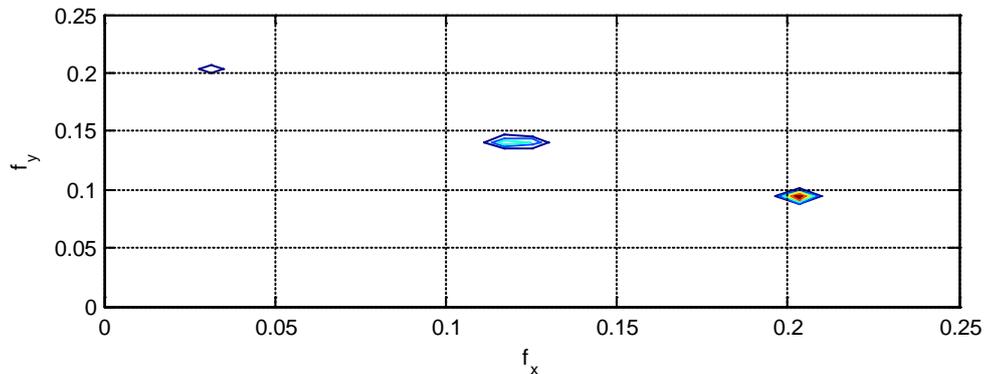


Figure 5.15. Projections de la densité spectrale estimée via la méthode de Capon 3D



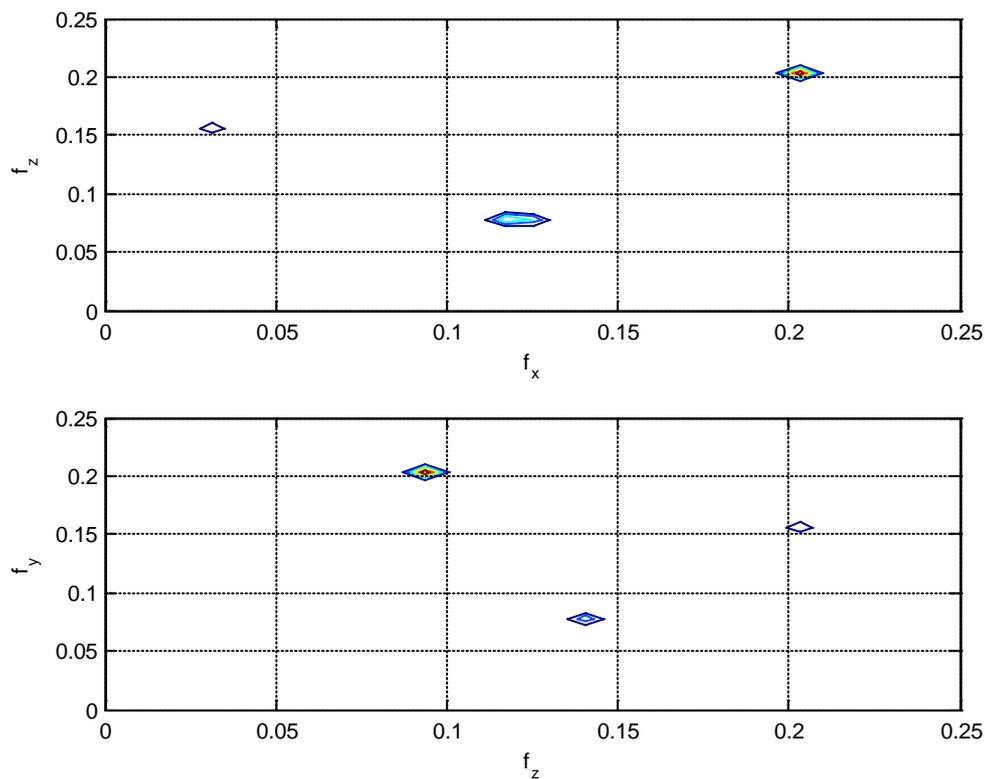


Figure 5.16. Projections de la densité spectrale estimée via la méthode APES 3D.

Pour le cas des signaux 3D, la même remarque qu'on a fait avec les signaux 2D reste valable, on constate d'après la figure (5.17) que 3 itérations seulement sont suffisantes pour la convergence du réseau RNZD2, alors que le réseau RNZD1, il nécessite quelque dizaines d'itérations. On note ici que la matrice à inverser est de dimension 512x512, ce qui prouve l'efficacité du réseau RNZD2.

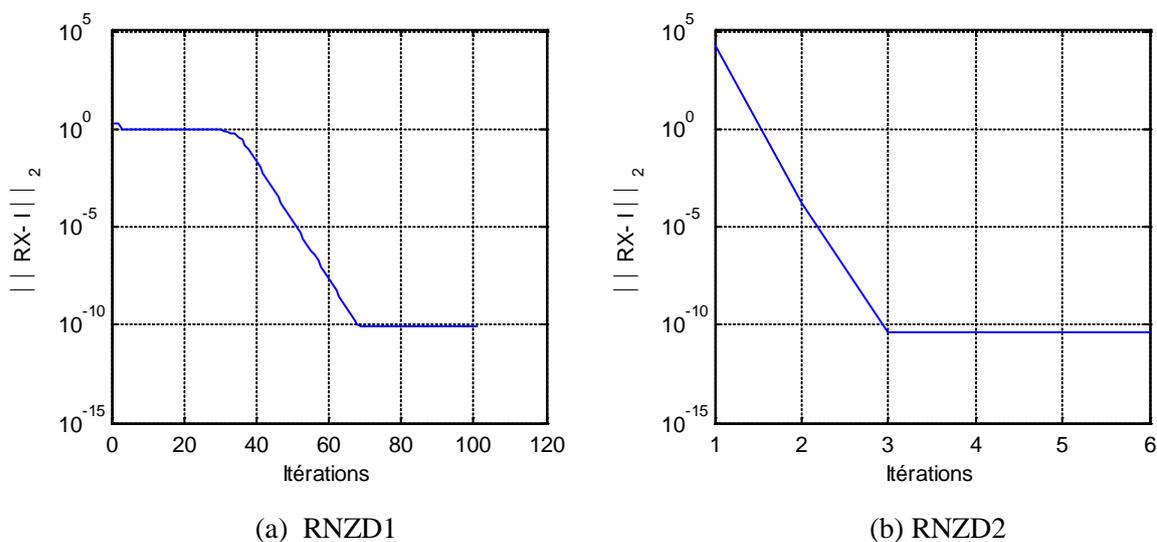


Figure 5.17. Erreur de convergence pour le cas des signaux 3D.

5.7. Conclusion

Dans ce chapitre, nous avons introduit le réseau de Zhang pour inverser la matrice de corrélation dans la méthode de Capon et la méthode APES pour l'estimation du spectre d'amplitudes. Deux modèles de réseau de Zhang ont été utilisés, l'un utilise la dynamique explicite (RNZD1), l'autre utilise la dynamique implicite (RNZD2).

Les résultats de simulation nous permettent de conclure que la méthode de Capon offre une grande aptitude dans l'estimation des fréquences même s'ils sont très étroites, alors que la méthode APES présente une bonne performance pour l'estimation de l'amplitude et la phase avec la connaissance à priori de la position spectrale des raies. De plus le réseau RNZD2 converge dans deux ou trois itérations par rapport au réseau RNZD1 ce qui lui qualifie pour les signaux multidimensionnels.

Conclusion Générale

Le travail réalisé dans cette thèse consiste à utiliser les réseaux de neurones récurrents dans le problème de l'estimation spectrale. Suivant la méthode de l'estimation spectrale considérée ; paramétrique ou non paramétrique, plusieurs modèles de réseaux ont été proposés. Le premier type utilisé est le réseau de Hopfield étendu aux données complexes. Dans ce modèle, le signal utilisé est supposé une somme de sinusoides complexes noyées dans du bruit et dont les fréquences des raies sont connues ou bien non connues. Deux cas sont distingués et ont été étudiés. Dans le premier cas, le problème de l'estimation spectrale est mis sous la forme d'une régression linéaire dont l'ordre est égal au nombre de sinusoides contenu dans le signal. Le réseau est construit de deux façons, l'une consiste à séparer les données en partie réelle et partie imaginaire, et de concevoir un réseau de Hopfield dont l'état contient la partie réelle et la partie imaginaire des amplitudes estimées. L'autre consiste à concevoir deux réseaux à valeurs réelles, l'un pour estimer la partie réelle, l'autre pour estimer la partie imaginaire des amplitudes estimées. Dans le deuxième cas, on s'intéresse à l'estimation simultanée des trois paramètres du signal ; l'amplitude, la phase et la fréquence. Le réseau est conçu autour d'un réseau de Hopfield modifié. Le deuxième type est un réseau proposé récemment par Zhang. Ce réseau est basé sur la minimisation d'un vecteur ou matrice des erreurs élément par élément au lieu de la puissance de l'erreur utilisée dans les réseaux conventionnels. Suivant le problème considéré, le réseau de Zhang est utilisé soit pour résoudre un système linéaire, soit pour inverser une matrice. Pour le premier cas, le réseau est utilisé dans l'estimation de la densité spectrale de puissance d'un processus AR par la méthode de covariance. Deux variantes d'architecture ont été proposées, l'une est basée sur une dynamique explicite du réseau et l'autre est basée sur une dynamique implicite. Cette dernière approche a été étendue pour l'estimation de la densité spectrale par la méthode de minimum de variance. L'estimateur neuronal considéré dans ce cas est constitué de deux parties, une partie neuronale pour l'estimation des paramètres AR utilisant la méthode de Yule-Walker, et une pour l'évaluation de la densité spectrale utilisant des blocs FFT selon l'algorithme de Marple. Suite à la structure spéciale de la matrice de corrélation dans la méthode Yule-Walker, une architecture réduite du réseau de Zhang pour calculer les paramètres AR a été proposée où le produit de la matrice des poids par le vecteur d'entrée a été calculé efficacement en réduisant la mémoire et la complexité par l'utilisation de la

transformée de Fourier rapide. Dans la méthode de Capon et la méthode APES pour l'estimation du spectre d'amplitudes d'un signal multidimensionnel, nous avons introduit le réseau de Zhang dans ces deux versions pour l'inversion de la matrice de corrélation. D'après les résultats des simulations, on a pu retirer les remarques suivantes:

- Bien que le réseau de Hopfield présente une bonne robustesse pour l'estimation des amplitudes complexes dans le cas où les fréquences sont connues, il est très sensible à l'état initial dans le cas où il est conçu pour estimer les fréquences en plus des amplitudes complexes suite à l'existence des minimums locaux.
- Le réseau de Zhang pour estimer les paramètres d'un modèle AR prouve une meilleure performance du point de vue convergence où deux itérations sont suffisantes pour la convergence du réseau dans le cas où les paramètres optimaux du réseau sont choisis.
- L'architecture réduite proposée du réseau qui possède la même performance que la méthode de Marple nous a permis une nette amélioration de la complexité. Cette faible complexité est obtenue en exploitant la structure Toeplitz de la matrice de corrélation.
- Le réseau de Zhang à dynamique implicite pour inverser la matrice de corrélation est très efficace surtout pour les signaux multidimensionnels.

Ces résultats ont fait l'objet des publications et communications suivantes:

- A. Benchabane, A. Bennia, F. Charif and A. Ahmed-Taleb, "Multi-Dimensional Capon Spectral Estimation Using Zhang Neural Networks", *Multidimensional systems and signals processing*. vol. 24, pp. 583-598. Springer 2013.
- A. Benchabane, A. Bennia and F. Charif, "Complex-Valued Hopfield Neural Network for Amplitude Estimation of Sinusoidal Signals", *GESJ: Computer Science and telecommunication*, vol. 30, no.1. 2011.
- A. Benchabane, A. Bennia, F. Charif and N. Djedi, "Estimation spectrale d'un signal sinusoidal multi-dimensionnel par le réseau de Hopfield", *Int. Conf. electronics and oil*, ICEO11, Ouargla, Algeria , 1-2 Mars 2011.
- A. Benchabane, A. Bennia and F. Charif, "Real / Complex Valued Hopfield Neural Network Models for Amplitude Estimation of Sinusoidal Signals", *Int. Congr. Models, optimisation & security of systems*, Tiaret, Algeria ,Mai 2010.
- A. Benchabane, A. Bennia and F. Charif, "Continuous-time Zhang Neural Networks for AR Spectral Estimator", *Int. Conf. electronics and oil*, ICEO13, Ouargla, Algeria, 5-6 Mars 2013.

Bibliographie

- [1] E. A. Robinson, “A historical perspective of spectrum estimation”, *Proc. IEEE*, vol. 70, pp. 885–907, 1982.
- [2] P. M. Djuric and S. M. Kay, *Spectrum Estimation and Modeling*, Digital Signal Processing Handbook, Ed. CRC Press LLC, 1999.
- [3] M. S. Hayes, *Statistical Digital Signal Processing and Modeling*, John Wiley & Sons, New York, 1996.
- [4] S. M. Kay, *Modern Spectral Estimation*, Prentice Hall, Englewood Cliffs, NJ, 1988.
- [5] G. Blanchet and M. Charbit, *Signaux et images sous Matlab*, HEES Science Europe Ltd, Paris, 2001.
- [6] A. D. Poularikas and Z. M. Ramdan, *Adaptive filtering primer with matlab*, Taylor & Francis Group. CRC Press, 2006.
- [7] J. Capon, “High-resolution frequency-wavenumber spectrum analysis”, *Proc. IEEE*, vol. 57, pp. 1408–1418, 1969.
- [8] H. Li, P. Stoica and J. Li, “Capon estimation of covariance sequences”, *Circuits, Systems, and Signal Processing*, vol. 17, no. 1, pp. 29-49, January 1998.
- [9] A. Jakobsson, T. Ekman and P. Stoica, “CAPON and APES spectrum estimation for real-valued signals”, *Proc. IEEE, Digital Signal Processing Workshop*, Aug. 1998.
- [10] P. Stoica, H. Li and J. Li, “A new derivation of the APES filter”, *IEEE Signal Processing Letters*, vol. 6, no. 8, pp. 205-206, August 1999.
- [11] A. Jakobsson and P. Stoica, “Combining Capon and APES for estimation of spectral lines”, *Circuits, Systems and Signal Processing*, vol. 19, no. 2, pp. 159-168, 2000.
- [12] B. R. Musicus, “Fast MLM Power Spectrum Estimation from Uniformly Spaced Correlations”, *IEEE Trans. Acoustics, Speech and Signal Processing*, vol.33, no.4, pp. 1333–1335, 1985.
- [13] Z. S. Liu, H. Li, and J. Li, “Efficient implementation of CAPON and APES for spectral estimation”, *IEEE Trans. Aerospace and Electron. Syst*, vol. 34, pp. 1314-1319, Oct. 1998.
- [14] H. Li, P. Stoica and J. Li, “Computationally efficient maximum likelihood estimation of structured covariance matrices”, *IEEE Trans. Signal Processing*, vol. 47, no. 5, pp. 1314-1323, May 1999.

- [15] H. Li, P. Stoica and J. Li, "Computationally efficient parameter estimation for harmonic sinusoidal signals", *Signal Processing*, vol. 80, no. 11, pp. 1937-1944, 2000.
- [16] T. Ekman, A. Jakobsson and P. Stoica, "On efficient implementation of the Capon algorithm", *Proc. European Signal Processing Conference*, Finland, 2000.
- [17] J. Benesty, J. Chen and Y. Huang, "Recursive and fast recursive Capon spectral estimators", *EURASIP J. Advances in Signal Processing*, 2007.
- [18] G. O. Glentis, "A Fast Algorithm for APES and Capon Spectral Estimation", *IEEE Trans. Signal Processing*, vol. 56, no. 9, pp. 4207-4220, 2008.
- [19] G. O. Glentis, "Efficient Algorithms for Adaptive Capon and APES Spectral Estimation", *IEEE Trans. Signal Processing*, vol. 58, no. 1, pp. 84-96, 2010.
- [20] S. L. Marple, M. Adeli and H. Liu, "Super-Fast Algorithm for Minimum Variance (Capon) Spectral Estimation", *Conf. Signals, Systems and Computers*, 2010.
- [21] X. S. Zhang, *Neural Networks in Optimization*. Norwell, MA, Kluwer, 2000.
- [22] J. Hertz, A. Krogh and R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison Wesley, New York, 1991.
- [23] Y. H. Hu and J. N. Hwang, *Handbook of neural network signal processing*, CRC PRESS, 2002.
- [24] U. Seiffert, "Artificial Neural Networks on Massively Parallel Computer Hardware", ESANN'2002, *Proc. European Symposium on Artificial Neural Networks*, Belgium, April 2002.
- [25] J. Misra and I. Saha, "Artificial neural networks in hardware: A survey of two decades of progress", *Neurocomputing*, vol.74, pp. 239-255, 2010.
- [26] B. Kosko, *Neural Network for Signal Processing*, Prentice Hall, New York, 1992.
- [27] A. Cichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing*, Wiley, New York, 1993.
- [28] P. Raj and S. L. Pinjare, "Design and Analog VLSI Implementation of Neural Network Architecture for Signal Processing", *European J. Scientific Research*, vol. 27, no. 2, pp. 199-216, 2009.
- [29] Y. T. Zhou, R. Chellappa, A. Vaid and B. K. Jenkins, "Image restoration using a neural network", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 1141-1151, 1988.
- [30] B. Bai and N. H. Farhat, "Radar image reconstruction based on neural net models", *Proc. IEEE APS/URSI Meeting*, pp. 774-777, Syracuse, NY, 1988.

- [31] F. Charif, N. Djedi and A. Benchabane, "On Parallel Implementation of Horn and Schunk Motion estimation Method", *Int. J. Computer and applications*. vol. 35, no.2, 2013.
- [32] G. Martinelli and R. Perfetti, "Neural network for real time spectral estimation", *Int. Conf. Neural Networks*, Paris, France, July 1990.
- [33] D. A. Karras and S. J. Varoufakis, "A neural network for estimating the parameters of multiple sinusoids", *Int. Conf. Neural Networks*, pp.1005-1010, July 1992.
- [34] G. Martinelli and R. Perfetti, "Neural network approach to spectral estimation of harmonic processes", *IEE Proc.* vol. 140, no. 2, April 1993.
- [35] Y. Xia and M .S. Kamel, "A Cooperative Recurrent Neural Network Algorithm For Parameter Estimation of Autoregressive Signals", *Int. Conf. Neural Networks, Canada*, pp.2516- 2522, July 2006.
- [36] Y. Xia, M. S. Kamel, "Cooperative Recurrent Neural Networks for the Constrained L1 Estimator", *IEEE Trans. Signal processing*, vol.55, no. 7, pp. 3192- 3206, July 2007.
- [37] D. Liu, N. Zhang, H. Liu and Y. Zhu, "A New Frequency Spectrum Estimation Method based on Neural Network Techniques", *Proc. Int. Sympo. Web Information Systems and Applications, Nanchang, China*, pp. 383-386, May 2009.
- [38] L. Han, "Neural Networks for Sinusoidal Frequency Estimation", *J. Franklin Inst.* vol. 334B, no.1, pp. 1- 18, 1997.
- [39] H. El-Bakry, "A Novel Hopfield Neural Network for Perfect Calculation of Magnetic Resonance Spectroscopy", *Int. J. Biological and Life Sciences*, vol. 3, no.1, 2007.
- [40] S. K. Park, "Hopfield Neural Network for AR Spectral Estimator", *Proc. IEEE'1990*, pp. 487-490, 1990.
- [41] A. Benchabane, A. Bennia and F. Charif, "Complex-Valued Hopfield Neural Network for Amplitude Estimation of Sinusoidal Signals", *GESJ: Computer Science and telecommunication*, vol. 30, no.1. 2011.
- [42] A. Benchabane, A. Bennia, F. Charif and N. Djedi, "Estimation spectrale d'un signal sinusoidal multi-dimensionnel par le réseau de Hopfield", *Int. Conf. electronics and oil, ICEO11, Ouargla, Algeria* , 1-2 Mars 2011.
- [43] A. Benchabane, A. Bennia and F. Charif, "Real / Complex Valued Hopfield Neural Network Models for Amplitude Estimation of Sinusoidal Signals", *Int. Congr. models, optimisation & security of systems*, Tiaret, Algeria ,Mai 2010.

- [44] A. Benchabane, A. Bennia and F. Charif, “ Continuous-time Zhang Neural Networks for AR Spectral Estimator”, *Int. Conf. electronics and oil*, ICEO13, Ouargla, Algeria , 5-6 Mars 2013.
- [45] A. Benchabane, A. Bennia, F. Charif and A. Ahmed-Taleb, “Multi-Dimensional Capon Spectral Estimation Using Zhang Neural Networks”, *Multidimensional systems and signals processing*. vol. 24, pp. 583-598. Springer 2013.
- [46] S. R. DeGraaf, “SAR imaging via modern 2-D spectral estimation methods”, *IEEE Trans. Image Processing*, vol. 7, pp. 729-761, May.1998.
- [47] M. R. Palsetia, and J. Li, “Using APES for interferometric SAR imaging”, *IEEE Trans. Image Processing*, vol. 7, pp. 1340-1353, Sept. 1998.
- [48] E. G. Larsson, G. Liu, P. Stoica and J. Li “ High-resolution SAR imaging with angular diversity”, *IEEE Trans. Aerospace and Electronic Systems*, vol. 37,pp. 1359-1372, Oct. 2001.
- [49] F. Gini, and F. Lombardini, “Multi look APES for multi baseline SAR interferometry”, *IEEE Trans. Signal Processing*, vol. 50, no. 7, pp. 1800-1803, 2002.
- [50] F. Lombardini, F. Cai and M. Pardini, “Parametric Differential SAR Tomography of Decorrelating Volume Scatterers”, *Proc. European Radar Conference*, 2009.
- [51] Y. Li, J. Razavilar and K. Liu, “A High Resolution Technique for Multi-Dimensional NMR Spectroscopy”, *IEEE Trans. Biomedical engineering*. Vol.45, no.1, pp.78-86, 1998.
- [52] F. Lombardini, M. Pardini and L.Verrazzani, “A Robust Multibaseline Sector Interpolator for 3D SAR Imaging”, *Proc. EUSAR 2008*, Germany ,2008.
- [53] E. O. Brigham, *The fast fourier transform and its applications*, Prentice Hall, 1988.
- [54] P. D. Welch, “The use of fast Fourier transform for estimation of power spectra: A method based on time averaging over short, modified periodograms “, *IEEE Trans. Audio and Electroacoustics*, vol. 15, pp. 70-73, June 1967.
- [55] R. B. Blackman and J. W. Tukey, *The Measurement of Power Spectra from the Point of View of Communications Engineering*, Dover Publications, New York, 1958.
- [56] B. Raton, *Advanced Signal Processing Handbook*, CRC Press LLC, 2001.
- [57] V. F. Pisarenko “The retrieval of harmonics from a covariance function”, *Geophysical J. Royal Astron. Soc*, vol.33, pp. 347–366, 1973.
- [58] P. Stoica and R. Moses, *Spectral Analysis of Signals*, Prentice hall.2005.
- [59] E. G. Larsson, J. Li and P. Stoica, *In High-resolution and robust signal processing*, Kindle Edition, 2003.

-
- [60] J. Li and P. Stoica, “An adaptive filtering approach to spectral estimation and SAR imaging”, *IEEE Trans. Signal Processing*, vol. 44, pp. 1469-1484, June 1996.
- [61] P. Stoica, A. Jakobsson and J. Li, “Matched-filter bank interpretation of some spectral estimators”, *Signal Process.* vol. 66, no.1, pp. 45–59, April 1998.
- [62] H. Li, J. Li and P. Stoica, “Performance analysis of forward-backward matched-filter bank spectral estimators”, *IEEE Trans. Signal Processing*, vol.46, no.7, pp. 1954-1966, July 1998.
- [63] E. G. Larsson, P. Stoica, and J. Li, “Spectral estimation via adaptive filter bank methods: A unified analysis and a new algorithm”, *Signal Processing*, vol. 82, pp. 1991-2001, Dec. 2002.
- [64] P. Stoica, H.Li and J. Li, “Amplitude estimation of sinusoidal signals: survey, new results, and an application”, *IEEE Trans. Signal Processing*, vol.48, no.2, pp. 338-352, February 2000.
- [65] D. Hush and B. Horne, “Progress in supervised neural networks”, *IEEE Signal processing magazine*, pp:8-39, 1993.
- [66] J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities”, *Proc. Natl. Acad. Sci. USA*, vol.79, pp. 2554-2558, 1982.
- [67] J. Hopfield, “Neurons with graded response have collective computational properties like those of two-state neurons”, *Proc. Natl. Acad. Sci. USA*, vol.81, pp. 3088-3092, 1984.
- [68] J. Hofield and D. Tank, “Neural computation of decisions in optimization problems”, *Biol. Cybernet*, vol. 52, pp. 141-152, 1985.
- [69] D. Tank and J. Hopfield, “Simple Neural Optimization Networks: An A/D Converter, Signal Decision Circuit and Linear Programming Circuits”, *IEEE Trans. Circuits and Systems*, CAS- 33, pp. 533-541, 1986.
- [70] A. Hirose, “Dynamics of fully complex-valued neural networks”, *Electr. Lett.* vol. 28, no.16, July 1992.
- [71] T. Nitta, *Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters*. Hershey, PA, USA: Information Science Reference-Imprint of IGI Publishing, 2009.
- [72] A. Hirose, “Complex-valued neural networks: The merits and their origins”, *Proc. IJCNN'09*, pp. 1237–1244, 2009.
- [73] D. Mandic and V. S. L. Goh, *Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models*. NY, USA, Wiley, 2009.
- [74] A. Hirose, *Complex-Valued Neural Networks*. Berlin, Germany, Springer, 2006.

- [75] Y. Zhang, K. Chen, W. Ma, "MATLAB simulation and comparison of Zhang neural network and gradient neural network for online solution of linear time-varying equations", *Proc. Int.l Conf. Life System Modeling and Simulation*, Shanghai, China, pp. 450-454, September 2007.
- [76] Y. Zhang, Z. Li, K.Chen and B. Cai, "Common nature of learning exemplified by BP and Hopfield neural networks for solving online a system of linear equations", *IEEE int. Conf. networking, sensing and control*, pp. 832–836, 2008.
- [77] D. Guo, C. Yi and Y. Zhang, "Zhang neural network versus gradient based neural network for time-varying linear matrix equation solving", *Neurocomputing*, vol. 74, no. 17, pp. 3708–3712, Oct. 2011.
- [78] C. Yi and Y. Zhang, "Analogue recurrent neural network for linear algebraic equation solving", *Electr. Lett.*, vol. 44, no. 18, pp. 1078-1079, August 2008.
- [79] Y. Zhang, Y. Shi, and Y. Yang, "Convergence Analysis of Zhang Neural Networks Solving Time-Varying Linear Equations but without Using Time-Derivative Information", *2010 8th IEEE Int. Conf. Control and Automation Xiamen*, China, pp.1215-1220, June 2010.
- [80] Z. Li and Y. Zhang, "Improved Zhang neural network model and its solution of time-varying generalized linear matrix equations", *Expert Syst. Appl.*, vol. 37, no. 10, pp. 7213–7218, Oct. 2010.
- [81] Y. Zhang, B. Mu and H. Zheng, "Discrete-Time Zhang Neural Network and Numerical Algorithm for Time-Varying Linear Equations Solving", *IEEE Proc. Int. Conf. Mechatronics and Automation*, Beijing, China, pp.934-943, August 2011.
- [82] I. Gohberg and V. Olshevsky, "Complexity of multiplication with vectors for structured matrices", *Linear Algebra and its Applications*, vol.202, pp. 163-192, 1994.
- [83] T. Kailath, S.Y. Kung and M. Morf, "Displacement ranks of matrices and linear equations", *J. Mathematical Analysis and Applications*, vol.68, no.2, pp.395-407, 1979.
- [84] T. Kailath and A.H.Sayed, "Displacement structure: Theory and applications", *SIAM Review*, vol.37, no.3, pp.297-386, 1995.
- [85] T. Kailath and A.H.Sayed, *Fast Reliable Algorithms for Matrices with Structure*, SIAM Publications, Philadelphia, 1999.
- [86] J. Mc. Clellan, "Multidimensional Spectral Estimation", *Proc. IEEE*, vol.70, pp. 1029-1037, September 1982.

- [87] D. Dudgeon and R. Mersereau, *Multidimensional Digital Signal Processing*, Prentice Hall, Inc., Englewood Cliffs, N.J., 1984.
- [88] J. Li, P. Stoica and D. Zheng, “An efficient algorithm for two-dimensional frequency estimation”, *Multidimensional Systems and Signal Processing*, vol.7, pp.151-178, April 1996.
- [89] E. Larsson and P.Stoica, “Fast implementation of two-dimensional APES and Capon spectral estimators”, *Multidimensional Systems and Signal Processing*, vol. 13, no.1, pp. 35–54, 2002.
- [90] A. Jakobsson, S. L. Marple and P. Stoica, “Two-dimensional CAPON spectral analysis”, *IEEE Trans. Signal Processing*, vol. 48, pp. 2651-2661, Sept. 2000.
- [91] S. Kay and L.Pakula, “Convergence of the Multidimensional Minimum Variance Spectral Estimator for Continuous and Mixed Spectra”, *IEEE Signal Proc. Lett.* vol.17, no.1, pp. 1-4, 2010.
- [92] Y. Zhang, “Revisit the analog computer and gradient-based neural system for matrix inversion”, *In IEEE int. sympo. Intelligent control*, pp.1411–1416, 2005.
- [93] Y. Zhang, W. Ma, and C. Yi, “The link between Newton iteration for matrix inversion and Zhang neural network (ZNN)”, *Proc. of IEEE Int. Conf. Industrial Technology*, Chengdu, China. 2008.
- [94] Y. Zhang, Z. Li and K. Li, “Complex-valued Zhang neural network for online complex-valued time-varying matrix inversion”, *Appl. Math.Comput.* vol. 217, no. 24, pp. 10066–10073, 2011.