

République Algérienne Démocratique et Populaire  
Ministère de l'enseignement supérieur et de la recherche scientifique  
Université 8Mai 1945 – Guelma  
Faculté des sciences et de la Technologie  
Département Electronique et Télécommunications



# Mémoire

Présenté pour obtenir le diplôme de Magister en Télécommunications

**OPTION: Télécommunications**

Par:

**GHADJATI Mohamed**

**Thème**

**Applications des réseaux de neurones aux  
systèmes de communication numériques**

**Devant le Jury :**

|                   |                            |            |                  |
|-------------------|----------------------------|------------|------------------|
| <b>Président</b>  | Pr. Hicham TEBBIKH         | Professeur | <b>U. Guelma</b> |
| <b>Rapporteur</b> | Dr. A. Krim MOUSSAOUI      | MC-A       | <b>U. Guelma</b> |
| <b>Examineurs</b> | Pr. Mohamed seghir BOUMAZA | Professeur | <b>U. Guelma</b> |
|                   | Dr. Hcène BOUALLEG         | MC-A       | <b>U. Guelma</b> |
| <b>Invité</b>     | Dr. Med Tahar TABA         | MC-B       | <b>U. Guelma</b> |

**SESSION 2012**

## Résumé

Ce mémoire traite le concept d'égalisation non linéaire des canaux de transmission par réseaux de neurones, et en particulier le perceptron multicouche. Nous avons présenté dans ce mémoire deux architectures principales, l'égaliseur MLP, et l'égaliseur DFE-MLP, qui sont les homologues des égaliseurs LTE, et DFE basés sur les filtres numériques. Nous avons montré également comment le perceptron multicouche peut être employé pour réaliser la fonction d'égalisation, et résoudre les problèmes causés par le bruit et l'interférence entre symboles, en fait, il s'agit de trouver la structure convenable qui donne les meilleurs résultats. Nous avons aussi comparé les performances des égaliseurs à base des réseaux de neurones et ceux obtenues par les égaliseurs conventionnels qui sont construits par des filtres numériques pour un canal linéaire. Les égaliseurs à base du MLP ont montré une grande capacité d'égalisation vis-à-vis aux autres. Particulièrement, l'égaliseur DFE-MLP a donné une amélioration substantielle en termes, du signal égalisé, de la valeur de l'état stable MSE et du taux d'erreur binaire BER.

**Mots clés :** Perceptron multicouche, canal de transmission, égalisation, interférence entre symboles, rapport signal sur bruit, modulation numérique.

## Summary

This thesis deals with the concept of nonlinear equalization of transmission channels by neural networks, in particular the multilayer perceptron. We presented in these paper two major architectures, MLP equalizer, and DFE-MLP equalizer, which are the counterparts of equalizers LTE and DFE based on digital filters. We have also shown how the multi-layer perceptron can be used to perform the function of equalization, and solve the problems caused by noise and inter-symbol interference; in fact, it is to find the appropriate structure that gives the best results. We also compared the performance of equalizers based on neural networks and those obtained by conventional equalizers built by digital filters for a linear channel. Equalizer based on the MLP has shown a great capacity of equalization against others. Specifically, the equalizer DFE-MLP gave a substantial improvement in terms of the equalized signal, the value of the steady state MSE and bit error rate BER.

**Keywords :** Multilayer perceptron, channel of communication, equalization, inter-symbol interference, signal to noise ratio, digital modulation.

## ملخص

هذه الأطروحة تعالج مفهوم المسويات اللاخطية لقنوات الاتصال باستعمال الشبكات العصبية الاصطناعية (Réseaux de neurones)، على وجه الخصوص البرسيبترون متعدد الطبقات (Perceptron multicouche) الذي هو احسن نوع من الشبكات العصبية يقوم بهذه المهمة. و لقد قدمنا في هذه المذكرة بنيتين اساسيتين للمسويات العصبية، هما، المسويات على اساس البرسيبترون متعدد الطبقات (égaleuseur MLP)، والمسويات الارجاعية على اساس البرسيبترون متعدد الطبقات (MLP – DFE)، والتي هي من نظائر المسويات (LTE) و(DFE) القائمة على اساس المرشحات الرقمية. لقد أظهرنا أيضا كيف يمكن استخدام البرسيبترون متعدد الطبقات متعدد الطبقات لأداء وظيفة المعادلة او التسوية، ويحل المشاكل التي يسببها الضجيج (Bruit) والتداخل بين الرموز (interférence entre symboles)، في الواقع، الامر متعلق بايجاد البنية المناسبة التي تقدم أفضل النتائج. و هو أيضا امر قائم على اساس التجربة. لقد قمنا أيضا باجراء مقارنة لأداء المسويات القائمة على الشبكات العصبية وتلك التي حصلت عليها المسويات التقليدية القائمة على المرشحات الرقمية، وهذا من اجل قناة خطية. وقد أظهرت المسويات على أساس البرسيبترون متعدد الطبقات قدرة كبيرة على التسوية مقارنة بالمسويات التقليدية. على وجه التحديد، أعطت المسويات الارجاعية MLP - DFE تحسينا كبيرا من حيث نوعية الإشارة، مستوى استقرار الخطأ MSE و معدل الخطأ بالبت BER.

**الكلمات الأساسية :** البرسيبترون متعدد الطبقات، قناة الاتصال، تسوية القناة، التداخل بين الرموز، نسبة الإشارة الى الضجيج، التضمين الرقمي.

# Remerciement

Mes remerciements vont tout premièrement à Dieu, le tout puissant, pour la volonté, la santé et la patience, qu'il nous a donnée durant toutes ces longues années.

Je voudrais, avant toute chose, exprimer ma plus grande reconnaissance à Monsieur Abdelkrim MOUSSAOUI, Maître de conférences à l'Université de Guelma, qui a su diriger ce travail avec diplomatie, efficacité et rigueur, tout en me laissant libre dans mes choix. Je lui en témoigne ici toute ma gratitude.

J'exprime mes remerciements aux membres de jury, d'avoir bien voulu consacrer du temps et de l'attention à mon travail, à commencer par Mr. Hicham TEBBIKH, Professeur à l'Université de Guelma qui me fait également l'honneur de présider le jury. J'exprime toute ma gratitude à Mr. Mohamed Seghir BOUMAAZA, Professeur à l'Université de Guelma et Mr. Hcène BOUALLEG, Maître de conférences à l'Université de Guelma qui ont accepté d'être examinateurs.

Je tiens à exprimer également ma gratitude à tous nos enseignants du Département d'électronique et de télécommunications.

Pour finir, je ne saurais exprimer à quel point je remercie ma famille pour son soutien et ses encouragements depuis toujours. Ce sont mes parents qui m'ont appris à donner le meilleur et c'est cette exigence de soi qui m'a aidée tout au long de mes études ; à mes parents je dédie ce travail. Mes amis sont également à remercier, pour leur soutien inconditionnel, ainsi que l'ensemble des personnes qui, de près ou de loin, ont contribué à la réalisation de ce travail.

# Sommaire

|  |           |
|--|-----------|
| <b>1. Introduction générale</b>  | <b>1</b>  |
| <b>2. Généralités sur les systèmes de communications numériques</b>      | <b>4</b>  |
| 2.1 Les éléments de base d'un système de communication numérique         | 5         |
| 2.1.1 Mesure de la qualité d'une transmission numérique                  | 7         |
| 2.2 Modulations numérique  | 9         |
| 2.2.1 Modulation d'amplitude ASK   | 9         |
| 2.2.2 Modulation de phase PSK  | 10        |
| 2.2.3 Démodulation cohérente d'une modulation M-QAM                      | 12        |
| 2.2.4 Modulation de fréquence FSK  | 12        |
| 2.3 Canal de transmission  | 14        |
| 2.3.1 Modèles mathématiques des canaux de transmission                   | 14        |
| 2.3.1.1 Canal à bruit additif  | 14        |
| 2.3.1.2 Canal à filtre linéaire avec bruit additif                       | 14        |
| 2.3.1.3 Canal à filtre linéaire variant dans le temps avec bruit additif | 14        |
| 2.3.2 Perturbations apportées par le canal de transmission               | 15        |
| 2.3.2.1 Le bruit blanc gaussien additif                                  | 15        |
| 2.3.2.2 L'évanouissement   | 15        |
| 2.3.2.3 L'interférence entre symboles                                    | 16        |
| 2.4 Filtres numériques   | 20        |
| 2.4.1 Les filtres à Réponse Impulsionnelle Finie (RIF)                   | 20        |
| 2.4.2 Les filtres à Réponse Impulsionnelle Infinie (RII)                 | 20        |
| 2.4.3 Comparaison entre les filtres RIF et RII                           | 22        |
| 2.4.4 Filtrage adaptatif   | 23        |
| 2.5 Conclusion   | 24        |
| <b>3. Techniques classiques d'égalisation</b>                            | <b>25</b> |
| 3.1 Fonction d'égalisation   | 26        |
| 3.2 Différents types d'égaliseurs  | 27        |
| 3.3 Egaliseur transverse   | 28        |
| 3.3.1 Critères d'optimisation  | 29        |
| 3.3.1.1 L'égaliseur par « <i>Zero Forcing</i> »                          | 29        |
| 3.3.1.2 L'égaliseur à erreur quadratique moyenne minimale                | 30        |
| 3.4 Egaliseur à retour de décision                                       | 32        |
| 3.5 Egalisation adaptative   | 33        |
| 3.6 Egalisation à maximum de vraisemblance et algorithme de Viterbi      | 34        |
| 3.6.1 Représentation du canal par un graphe de treillis                  | 35        |

|  |           |
|--|-----------|
| 3.7 Résultats de simulation  | 36        |
| 3.7.1 Canal à phase non minimale utilisé dans la simulation            | 36        |
| 3.7.2 Simulation des égaliseurs  | 37        |
| 3.8 Conclusion   | 43        |
| <b>4. Egalisation à base des réseaux de neurones</b>                   | <b>44</b> |
| 4.1 Historique   | 45        |
| 4.2 Le neurone formel  | 45        |
| 4.2.1 Différents types de neurones                                     | 46        |
| 4.3 Les réseaux de neurones formels                                    | 48        |
| 4.3.1 Définition   | 48        |
| 4.3.2 Domaines d'application des réseaux de neurones                   | 48        |
| 4.3.3 Différents types des réseaux de neurones formels                 | 49        |
| 4.3.3.1 Le perceptron multicouche (Multi Layer Perceptron)             | 49        |
| 4.3.3.2 Les réseaux de fonctions à base radiale (RBF)                  | 50        |
| 4.3.3.3 Les réseaux de neurones bouclés (récurrents)                   | 52        |
| 4.3.4 Apprentissage des réseaux de neurones                            | 53        |
| 4.3.4.1 L'algorithme de rétro-propagation (Back Propagation)           | 54        |
| 4.3.5 Généralisation et contrôle de complexité des réseaux de neurones | 56        |
| 4.3.6 Propriétés des réseaux de neurones                               | 57        |
| 4.3.6.1 La propriété d'approximation universelle                       | 57        |
| 4.3.6.2 La propriété de parcimonie                                     | 57        |
| 4.3.7 Conclusion sur les réseaux de neurones                           | 58        |
| 4.4 Réseaux de neurones et égalisation                                 | 58        |
| 4.5 Performances des égaliseurs et résultats de simulation             | 60        |
| 4.5.1 Structure de l'égaliseur MLP                                     | 60        |
| 4.5.2 Comparaison des signaux en sortie des égaliseurs                 | 61        |
| 4.5.3 Comparaison des courbes MSE                                      | 64        |
| 4.5.4 Diagramme de l'œil   | 67        |
| 4.5.5 Comparaison des performances BER                                 | 69        |
| 4.6 Conclusion   | 69        |
| <b>5. Conclusion générale</b>  | <b>70</b> |
| <b>6. Annexes</b>  | <b>72</b> |
| 6.1 Algorithmes des moindres carrés (LMS)                              | 72        |
| 6.2 Algorithmes des moindres carrés récursifs (RLS)                    | 74        |
| 6.3 Méthode de Levenberg-Marquardt                                     | 77        |
| 6.4 Apprentissage des réseaux RBF                                      | 81        |
| 6.5 Lemme d'inversion d'une matrice                                    | 85        |

# Liste des figures

## Chapitre 2

|     |  |    |
|-----|--|----|
| 2.1 | Les éléments de base d'un system de communication numérique    | 5  |
| 2.2 | Modulation 2-ASK ( <i>On Off Keying</i> )                      | 10 |
| 2.3 | Modulation 2-PSK   | 11 |
| 2.4 | Modulation 2-FSK   | 13 |
| 2.5 | Types de canaux de transmission                                | 15 |
| 2.6 | Transmission en bande de base d'un signal QAM ou PAM           | 16 |
| 2.7 | Illustration des interférences entre symboles                  | 19 |
| 2.8 | Structure directe des filtres, (a) <b>RIF</b> , (b) <b>RII</b> | 22 |

## Chapitre 3

|      |  |    |
|------|--|----|
| 3.1  | Chaîne de transmission en présence d'égalisation   | 26 |
| 3.2  | Différents types, structures, et algorithmes d'égaliseurs  | 28 |
| 3.3  | Structure d'un égaliseur transverse  | 29 |
| 3.4  | Egaliseur à retour de décision   | 33 |
| 3.5  | Egaliseur adaptatif : (1) piloté par les décisions, (2) apprentissage                                | 34 |
| 3.6  | Graphe en treillis pour une modulation à deux états et un canal de mémoire égale à deux échantillons | 35 |
| 3.7  | Caractéristiques du canal de simulation  | 37 |
| 3.8  | Courbes de convergence de l'EQM de l'égaliseur LTE   | 39 |
| 3.9  | Courbes de convergence de l'EQM des égaliseurs LTE et DFE (LMS)                                      | 39 |
| 3.10 | Diagramme de l'œil du signal BPSK transmis   | 40 |
| 3.11 | Diagramme de l'œil du signal BPSK en sortie du canal AWGN  | 40 |
| 3.12 | Diagramme de l'œil du signal en sortie de l'égaliseur LTE-LMS  | 41 |
| 3.13 | Diagramme de l'œil du signal en sortie de l'égaliseur LTE-RLS  | 41 |
| 3.14 | Diagramme de l'œil du signal en sortie de l'égaliseur DFE-LMS  | 42 |
| 3.15 | Diagramme de l'œil du signal MLSE  | 42 |
| 3.16 | Courbes BER des différents égaliseurs  | 43 |

## Chapitre 4

|             |  |    |
|-------------|--|----|
| <b>4.1</b>  | Neurone formel   | 46 |
| <b>4.2</b>  | Différentes fonctions d'activation   | 47 |
| <b>4.3</b>  | Perceptron multicouche   | 50 |
| <b>4.4</b>  | Réseau RBF   | 51 |
| <b>4.5</b>  | Fonction à base radiale (RBF)  | 51 |
| <b>4.6</b>  | Forme canonique d'un réseau de neurone bouclé                                  | 53 |
| <b>4.7</b>  | (a) Apprentissage supervisé, (b) Apprentissage non supervisé                   | 54 |
| <b>4.8</b>  | Classification des algorithmes d'apprentissage                                 | 54 |
| <b>4.9</b>  | Egaliseur à base du perceptron multicouche, <b>(a)</b> MLP, <b>(b)</b> DFE-MLP | 59 |
| <b>4.10</b> | Signal transmis et signal égalisé, MLP (BP)                                    | 61 |
| <b>4.11</b> | Signal transmis et signal égalisé, DFE-MLP (BP)                                | 62 |
| <b>4.12</b> | Signal transmis et signal égalisé, DFE-RBF (K-means/BP)                        | 62 |
| <b>4.13</b> | Signal transmis et signal égalisé, DFE-RBF (BP)                                | 63 |
| <b>4.14</b> | Signal transmis et signal égalisé, DFE-MLP (LM)                                | 63 |
| <b>4.15</b> | Diagramme de l'œil de l'égaliseur MLP (BP)                                     | 64 |
| <b>4.16</b> | Diagramme de l'œil de l'égaliseur DFE-MLP (BP)                                 | 65 |
| <b>4.17</b> | Diagramme de l'œil de l'égaliseur DFE-RBF (K-means/BP)                         | 65 |
| <b>4.18</b> | Diagramme de l'œil de l'égaliseur DFE-RBF (BP)                                 | 66 |
| <b>4.19</b> | Diagramme de l'œil de l'égaliseur DFE-MLP (LM)                                 | 66 |
| <b>4.20</b> | Courbes de convergence de l'EQM des égaliseurs DFE-RBF                         | 67 |
| <b>4.21</b> | Courbes de convergence de l'EQM des égaliseurs MLP (BP)                        | 68 |
| <b>4.22</b> | Courbe de convergence de l'EQM de l'égaliseur DFE-MLP (LM)                     | 68 |
| <b>4.23</b> | Courbes BER des différents égaliseurs  | 69 |

# Abréviations

|              |  |
|--------------|--|
| <b>BER</b>   | Bit Error Rate                             |
| <b>TEB</b>   | Taux d'Erreur Binaire                      |
| <b>ASK</b>   | Amplitude Shift Keying                     |
| <b>PSK</b>   | Phase Shift Keying                         |
| <b>BPSK</b>  | Binary Phase Shift Keying                  |
| <b>QAM</b>   | Quadratique Amplitude Modulation           |
| <b>FSK</b>   | Frequency Shift Keying                     |
| <b>CPFSK</b> | Continuous Phase Frequency Shift Keying    |
| <b>MSK</b>   | Minimum Shift Keying                       |
| <b>GMSK</b>  | Gaussian Minimum Shift Keying              |
| <b>OFDM</b>  | Orthogonal Frequency Division Multiplexing |
| <b>SNR</b>   | Signal to Noise Ratio                      |
| <b>PAM</b>   | Pulse Amplitude Modulation                 |
| <b>IES</b>   | Interférence Entre Symboles                |
| <b>RIF</b>   | Réponse Impulsionnelle Finie               |
| <b>RII</b>   | Réponse Impulsionnelle Infinie             |
| <b>LTE</b>   | Linear Transversal Equalizer               |
| <b>DFE</b>   | Decision Feedback Equalizer                |
| <b>MLSE</b>  | Maximum Likelihood Sequence Estimation     |
| <b>LMS</b>   | Least Mean Square                          |
| <b>RLS</b>   | Recursive Least Square                     |
| <b>MSE</b>   | Mean Square Error                          |
| <b>MLP</b>   | Multi Layer Perceptron                     |
| <b>RBF</b>   | Radial Basis Function                      |
| <b>RNN</b>   | Recurrent Neural Network                   |
| <b>EBP</b>   | Error Back Propagation algorithm           |

# Chapitre 1

## Introduction générale

---

Les télécommunications font partie des technologies qui ont révolutionné notre mode de vie au vingtième siècle. Du télégraphe à l'Internet, de la transmission sans fil au téléphone cellulaire, les progrès établis sont spectaculaires. Les informations transmises étaient tout d'abord codées en morse, puis des techniques de modulation et de codage analogiques ont permis de transmettre du son, puis des images. Ensuite la venue des techniques numériques a considérablement augmenté le débit et la qualité des informations à transmettre d'un point à un autre.

Parallèlement, le développement rapide de la microélectronique et des capacités de miniaturisation permettent aujourd'hui la mise en œuvre de techniques complexes dans des appareils de taille réduite. Cependant l'augmentation des besoins en débit se heurte à la nature des canaux eux-mêmes. En effet, dans des applications telles que la télédiffusion à grande échelle ou un réseau informatique radio à l'intérieur d'un bâtiment, le canal est de type multitrajet. Le signal est réfléchi en plusieurs endroits, et des échos apparaissent et créent des perturbations telles que l'interférence entre symboles créée par la sélectivité en fréquence dont l'influence augmente avec le débit de transmission.

Pour lutter contre la sélectivité en fréquence des canaux de transmission plusieurs techniques sont possibles parmi lesquelles on peut citer : les transmissions

multiporteuses, les techniques d'étalement de spectre et l'égalisation. Ce mémoire étant consacré à l'égalisation.

Les égaliseurs ont pour fonction principale d'inverser la réponse du canal de transmission de manière à ce que le couple "canal-égaliseur" puisse être considéré comme un canal idéal. L'opération d'égalisation peut provoquer, selon la nature de l'égaliseur utilisé des remontées plus ou moins importantes du bruit présent à son entrée.

Un récepteur optimisé selon le critère du maximum de vraisemblance, dont nous allons brièvement rappeler le principe, et un autre outil pour combattre les effets de la sélectivité en fréquence du canal.

L'égalisation est donc, une technique de traitement de l'interférence entre symboles efficace lorsque le rapport signal à bruit est suffisamment grand. Notons enfin que l'égalisation est bien adaptée pour traiter des canaux ayant une réponse impulsionnelle longue vis à vis de la durée des symboles transmis et qu'elle conduit aussi à de bons résultats en présence de modulations à grand nombre d'états. Dans ce mémoire nous étudierons les propriétés des égaliseurs classiques, transversal et à retour de décision. Ensuite les égaliseurs neuronaux feront l'objet d'une étude ultérieure.

Les égaliseurs adaptatifs les plus simples sont construits à partir de filtres transverses dont les coefficients sont généralement actualisés à partir d'un algorithme du gradient ou éventuellement, des moindres carrés récursifs. Pour favoriser la convergence d'un égaliseur on utilise une séquence d'apprentissage c'est-à-dire, une séquence préliminaire de données, connue du récepteur. Lorsque cela n'est pas possible, on est contraint d'utiliser des égaliseurs autodidactes qui basent leur traitement sur la connaissance a priori des propriétés statistiques des signaux émis, cette approche, qui fait l'objet de travaux intensifs dans la communauté scientifique, ayant un objectif double, d'une part l'absence de séquence d'apprentissage permet d'accroître l'efficacité spectrale et d'autre part dans certaines applications spécifiques (radio ou télédiffusion numérique) il est absolument impensable d'émettre périodiquement des préambules pour satisfaire les utilisateurs potentiels. La stratégie non supervisée ou autodidacte s'impose donc fortement. .

Les réseaux de neurones constituent un ensemble des techniques de traitement du signal qui s'inspirent du système nerveux humain. Ces techniques ont profité elles aussi du développement de la micro-électronique et constituent des systèmes capables d'accomplir des tâches complexes telles que la reconnaissance de visage ou l'écriture manuscrite. Elles trouvent de plus une place dans le domaine des télécommunications, en particulier dans les systèmes comportant des non-linéarités. Les travaux présentés dans ce mémoire ont donc pour but d'étudier leur emploi dans le contexte d'égalisation.

Ce mémoire est organisé en 4 chapitres.

Le premier chapitre est une introduction générale du mémoire.

Le deuxième chapitre représente un récapitulatif des outils des communications numériques, les nécessaires, pour avancer dans ce mémoire, parmi lesquels on cite : quelques types de modulation numériques, les modèles des canaux de transmission et les perturbations qui les accompagnent, telles que le bruit et l'interférence entre symboles, et les outils de mesure de la qualité de transmission (BER, diagramme de l'œil), et enfin un aperçu sur les filtres numériques.

Le troisième chapitre explique la notion d'égalisation, et présente une étude des égaliseurs conventionnels ; l'égaliseur transversal linéaire LTE (Linear Transversal Equalizer) et l'égaliseur à retour de décision DFE (Decision Feedback Equalizer), et les différents critères d'optimisation. Il expose brièvement le récepteur basé sur l'estimation de la séquence au sens du maximum de vraisemblance. Ce chapitre présente une comparaison entre les deux égaliseurs LTE et DFE en termes de performances, et montre leurs limitations face aux non linéarités du canal.

Le quatrième chapitre se concentre sur l'utilisation des réseaux de neurones pour la fonction d'égalisation, c'est-à-dire un traitement non linéaire. Il présente une étude des réseaux de neurones, leurs types, le perceptron multicouche (MLP) et les réseaux de fonction à base radiale (RBF), leurs domaines d'application, leur apprentissage, et leur généralisation.

Deux architectures d'égaliseurs à base des réseaux de neurones sont mises en œuvre pour l'égalisation du canal linéaire à phase non minimale, l'égaliseur direct, qui la version neuronale de l'égaliseur LTE et l'égaliseur DFE. Ces deux structures sont réalisées à l'aide des réseaux MLP et RBF en utilisant différents algorithmes d'apprentissage, tels que l'algorithme de rétro-propagation de l'erreur (Error Back Propagation), l'algorithme K-means, et l'algorithme de Levenberg-Marquardt.

Les simulations examinent l'efficacité de ces égaliseurs à réduire l'effet du bruit et du canal linéaire à phase non minimale utilisé, sur un signal BPSK. Les performances des égaliseurs présentés sont aussi comparées et évaluées en utilisant différents outils de mesure de performance tels que les courbes de convergence de l'erreur quadratique moyenne, le diagramme de l'œil, et les courbes BER dans le but de déterminer la structure optimale pour égaliser le canal.

La conclusion générale regroupe les résultats obtenus au cours de ce mémoire.

# Chapitre 2

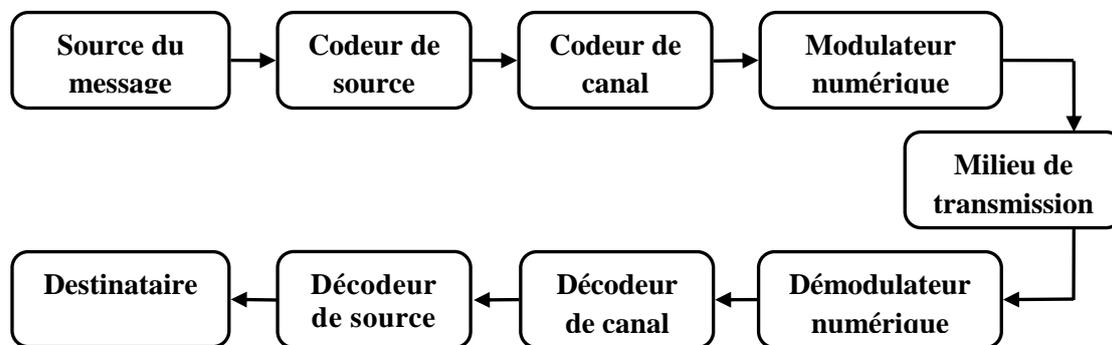
## Généralités sur les systèmes de communications numériques

---

Ce chapitre a pour but d'introduire les systèmes de communications numériques, et les notions de base qui serviront à la bonne compréhension de ce mémoire. Après un aperçu sur les éléments de base constituant un système de communication numérique partant de l'émetteur jusqu'au destinataire, les modulations numériques sont présentées, ainsi que leurs variantes. Puis on a introduit les différents modèles de canaux de transmission et les perturbations apportées par eux, telles que le bruit et la sélectivité en fréquence et notamment l'interférence entre symboles qui représente une source de dégradation importante qui affecte le signal, et qui fait l'objectif de l'opération d'égalisation. Finalement, et du fait qu'ils constituent les structures de base des égaliseurs classiques tels que le LTE (Linear Transversal Equalizer) et le DFE (Decision Feedback Equalizer), les filtres numériques sont évoqués dans ce chapitre en introduisant leur stabilité et leur adaptabilité, cette dernière qui donne une idée préalable sur les égaliseurs adaptatif.

## 2.1 Les éléments de base d'un système de communication numérique

La **figure 2.1** illustre le schéma fonctionnel et les éléments de base d'un système de communication numérique. La sortie de la source peut être soit un signal analogique, tel que le signal audio ou vidéo, ou un signal numérique, comme la sortie d'un télécopieur, qui est discret dans le temps et a un nombre fini de caractères de sortie. Dans un système de communication numérique, les messages produits par la source sont convertis en une séquence d'éléments binaires. Idéalement, on représente la sortie de la source (message) par des éléments binaires aussi peu que possible. En d'autres termes, on cherche une représentation efficace de la sortie de la source qui se traduit par peu ou pas de redondance. Le processus de conversion (efficace) de la sortie de la source, analogique ou numérique, en une séquence d'éléments binaires est appelée *encodage de la source* ou la compression de données.



**Figure 2.1** Les éléments de base d'un système de communication numérique.

La séquence d'éléments binaires à partir du codeur de source, qu'on appelle la séquence d'information, est passée à *l'encodeur de canal*. Le but du codeur canal est d'introduire, de manière contrôlée, une certaine redondance dans la séquence d'information binaire qui peut être utilisée au niveau du récepteur pour surmonter les effets du bruit et des interférences rencontrés dans la transmission du signal à travers le canal. Ainsi, la redondance ajoutée sert à augmenter la fiabilité des données reçues et améliore la fidélité du signal reçu. En effet, la redondance dans la séquence d'information aide le récepteur dans le décodage de la séquence désirée. Par exemple, une forme triviale de codage de la séquence d'information binaire est simplement de répéter chaque élément binaire (bit)  $m$  fois, où  $m$  est un entier positif. En pratique le codage canal consiste à prendre  $k$  bits d'information à la fois et les transformer en une séquence unique de  $n$  bits, appelée *mot de code*. Le coût de la redondance introduite par le codage des données de cette manière est mesuré par le rapport  $n/k$ . L'inverse de ce rapport,  $k/n$ , est appelé le *taux de code*.

La séquence binaire à la sortie du codeur canal est passée au *modulateur numérique*, qui sert d'interface au canal de communication. Puisque la plupart des canaux de communication rencontrés dans la pratique sont capables de transmettre des signaux électriques (formes d'onde), le but principal du modulateur numérique est

de transformer la séquence d'informations binaires en un signal de forme d'onde. Pour plus de précision, supposons que la séquence d'informations codée doit être transmise d'un bit à la fois à un taux uniforme de  $R.bit / s$ . le modulateur numérique peut tout simplement associer au bit 0 une forme d'onde  $S_0(t)$  et au bit 1 une forme d'onde  $S_1(t)$ . Ainsi, chaque bit provenant du codeur canal est transmis séparément. On appelle cette technique *modulation binaire*. En variante, le modulateur peut transmettre une information de  $n$  bits à la fois en utilisant  $M = 2^n$  formes d'onde distinctes  $S_i(t)$ ,  $i=0, M-1$ , une forme d'onde pour chaque séquence de  $n$  bits. On appelle cette technique *modulation M-aire* ( $M > 2$ ).

Ainsi, lorsque le débit  $R$  du canal est fixe, la durée nécessaire pour transmettre une forme d'onde parmi  $M$ , est  $n$  fois la durée nécessaire dans un système qui utilise la modulation binaire.

Le *canal de communication* est le support matériel qui est utilisé pour envoyer le signal de l'émetteur vers le récepteur. Dans la transmission sans fil, le canal peut être l'atmosphère (espace libre). D' autre part, les voies téléphoniques utilisent généralement une variété de supports physiques, y compris les lignes filaires, les fibres optiques et le sans fil (micro-ondes). Quel que soit le support physique utilisé pour la transmission de l' information, la caractéristique essentielle est que le signal transmis est corrompu d'une manière aléatoire par une variété de mécanismes possibles, tels que le bruit thermique additif généré par les appareils électroniques, bruits artificiels ( bruit d'allumage de l'automobile ), et le bruit atmosphérique, ( la foudre pendant un orage électrique ).

A la réception d'un système de communication numérique, le *démodulateur numérique* traite les signaux transmis endommagés par le canal et réduit les formes d'onde (signaux physiques) à une séquence de nombres qui représentent les estimations des symboles de données transmises (binaire ou M-aire). Cette séquence de nombres est transmis au *décodeur canal*, qui tente de reconstituer la séquence d'information originale à partir de la connaissance du code utilisé par le codeur canal et la redondance contenue dans les données reçues.

Une mesure des performances du démodulateur et du décodeur est la fréquence avec laquelle les erreurs se produisent dans la séquence décodée. Plus précisément, la probabilité moyenne des erreurs binaires à la sortie du décodeur est une mesure de la performance de la combinaison démodulateur - décodeur. En général, la probabilité d'erreur est une fonction des caractéristiques de code, les types de signaux utilisés pour transmettre les informations sur le canal, la puissance d'émission, les caractéristiques du canal (quantité de bruit), la nature des interférences, et la méthode de démodulation et de décodage.

Comme dernière étape, quand une sortie analogique est souhaitée, le *décodeur de source* reçoit la séquence de sortie du décodeur canal et, à partir de la connaissance de la méthode de codage source utilisée, tente de reconstituer le signal original de la source. le signal à la sortie du décodeur source, et en raison des erreurs de décodage

canal et d'éventuelles distorsions introduites par le codeur de source et, peut-être, le décodeur source, est une approximation du signal original de la source. La différence entre le signal original et le signal reconstruit est une mesure de la distorsion introduite par le système de communication numérique [1].

### 2.1.1 Mesure de la qualité d'une transmission numérique

La qualité d'une transmission dépend de fidélité avec laquelle les éléments binaires du message sont restitués au destinataire. Elle se mesure en général en évaluant la probabilité d'erreur par élément binaire, notée  $P_{eb}$  définie comme la probabilité de prendre une décision erronée sur un élément binaire. Cette probabilité d'erreur n'est jamais strictement nulle, mais cela ne signifie pas pour autant que la transmission est de mauvaise qualité ; en effet, il suffit qu'elle prenne une valeur suffisamment faible pour satisfaire à un certain critère de fidélité, cette valeur dépendant du type d'information transmise (parole, son, image, données,...) et du niveau de fidélité exigé : une probabilité d'erreur de  $10^{-6}$  par exemple pourrait être jugée tout à fait satisfaisante pour la transmission de la parole en téléphonie.

Désignons par  $\alpha_k$  l'élément binaire émis à l'instant  $kT_b$ . En tenant compte du fait que les éléments binaires  $\alpha_k$  issus de la source sont indépendants et identiquement distribués sur l'alphabet  $\{0,1\}$ , la probabilité d'erreur par élément binaire  $P_{eb}$  est égale à :

$$P_{eb} = P(\alpha_k = 1)P(\hat{\alpha}_k = 0/\alpha_k = 1) + P(\alpha_k = 0)P(\hat{\alpha}_k = 1/\alpha_k = 0) \quad (2.1)$$

Où  $\hat{\alpha}_k$  représente le résultat de la décision prise sur l'élément binaire  $\alpha_k$ .

Pour un canal perturbé par un bruit additif  $B(t)$  stationnaire, cette probabilité d'erreur par élément binaire ne dépend pas de l'indice  $k$  considéré.

La mesure en laboratoire de cette probabilité d'erreur est réalisée en émettant une séquence de  $N$  éléments binaires connue du destinataire, puis en évaluant en réception le rapport entre le nombre  $n$  d'éléments binaires erronés et le nombre  $N$  d'éléments binaires émis. Ce rapport, appelé *Taux d'Erreur Binaire* (*Bit Error Rate* en anglais)  $TEB$  est sous certaines conditions, une bonne estimation de la probabilité d'erreur  $P_{eb}$ .

On Peut écrire :

$$TEB = \frac{1}{N} \sum_{k=1}^N X_k \quad (2.2)$$

Où  $X_k$  est une variable aléatoire discrète qui prend la valeur 1 avec la probabilité  $P_{eb}$  si l'élément binaire  $\alpha_k$  est mal décodé et la valeur 0 avec la probabilité  $(1-P_{eb})$  dans

le cas contraire ; il s'agit donc d'une variable qui suit la loi de Bernoulli de paramètre  $P_{eb}$ .

Le taux d'erreur binaire est donc aussi une variable aléatoire que l'on peut caractériser au second ordre par sa moyenne  $m_{TEB}$  et sa variance  $\sigma_{TEB}^2$  :

$$m_{TEB} = E [TEB] = \frac{1}{N} \sum_{k=1}^N E [X_k] \quad (2.3)$$

$$\sigma_{TEB}^2 = E \left[ (TEB - E [TEB])^2 \right] \quad (2.4)$$

La grandeur  $\beta(TEB) = E [TEB] - P_{eb}$  est appelée le « biais » de l'estimateur ; elle mesure en quelque sorte l'erreur systématique, la variance de l'estimateur mesurant la précision de la mesure. Un bon estimateur a bien entendu un biais nul et une variance faible.

En tenant compte du fait que l'espérance de la variable  $X_k$  est égale à  $P_{eb}$ , et ceci quel que soit l'indice  $k$  considéré, la valeur moyenne du taux d'erreur est égale à la probabilité d'erreur.

$$m_{TEB} = E [TEB] = P_{eb} \quad (2.5)$$

L'estimateur TEB est donc sans biais. Si les erreurs de transmission sont indépendantes, les variables  $X_k$  sont aussi indépendantes et la variance est égale :

$$\sigma_{TEB}^2 = \frac{P_{eb}(1-P_{eb})}{N} \quad (2.6)$$

L'estimation de la probabilité d'erreur  $P_{eb}$  à partir de la mesure du TEB est donc entachée d'une erreur de mesure qui est fonction du nombre  $N$  d'éléments binaires utilisés et de la probabilité d'erreur considérée. Cette erreur peut être évaluée à partir de l'erreur quadratique relative  $\varepsilon^2$ , égale à :

$$\varepsilon^2 = \frac{\sigma_{TEB}^2}{m_{TEB}^2} = \frac{1-P_{eb}}{NP_{eb}} \approx \frac{1}{NP_{eb}} \quad \text{si } P_{eb} \ll 1 \quad (2.7)$$

L'expression ci-dessus permet de déterminer le nombre  $N$  d'éléments binaires (et donc la durée  $N/D$  pour un débit  $D$ ) nécessaire pour mesurer un TEB avec une erreur de mesure  $\varepsilon$  donnée. Le temps nécessaire à la mesure d'un TEB petit peut donc devenir rapidement important, pour une transmission à faible débit. Ainsi pour estimer une probabilité d'erreur  $P_{eb}=10^{-6}$  avec une erreur de mesure inférieure à 10 % ( $\varepsilon^2 < 10^{-2}$ ), il faudra mesurer le TEB à partir d'un échantillon de  $N=10^8$  éléments binaires, ce qui nécessitera une seconde pour un débit de 100 Mbit/s, mais  $10^5$  seconde soit plus d'une journée, pour un débit de 1 kbit/s !

On notera que enfin que, d'après (2.5) et (2.7), on peut écrire :

$$E [n] = \frac{1}{\varepsilon^2} \quad (2.8)$$

Ceci signifie que, si l'on veut une erreur de mesure inférieure à 10 %, il faut compter, en moyenne,  $n=100$  erreurs. Cette valeur est intéressante à connaître quand on fait des mesures de TEB par simulation.

Il importe de souligner, que les résultats établis ci-dessus ne sont valables que lorsque les erreurs successives soient indépendantes [2].

## 2.2 Modulations numériques :

Pour transmettre des données sur un canal physique, il faut leur associer un signal physique qui porte l'information relative à ces données et qui est bien adapté au canal.

On distingue en pratique les transmissions dites en bande de base, qui s'effectuent dans une bande de fréquences incluant la fréquence nulle et ne mettent pas explicitement en œuvre de porteuse, et les transmissions sur fréquence porteuse, qui effectue une transmission dans une bande de fréquence centrée autour de la fréquence porteuse non nulle. Dans ce cas, la séquence de symboles à émettre module une porteuse en modifiant l'un ou plusieurs de ses paramètres, d'amplitude, de phase ou de fréquence, éventuellement de polarisation. Dans certains cas, la transmission peut s'effectuer en modulant simultanément plusieurs porteuses.

Selon le type de modulation considéré, le signal modulé a les expressions suivantes :

### 2.2.1 Modulation d'amplitude (ASK: *Amplitude Shift Keying*)

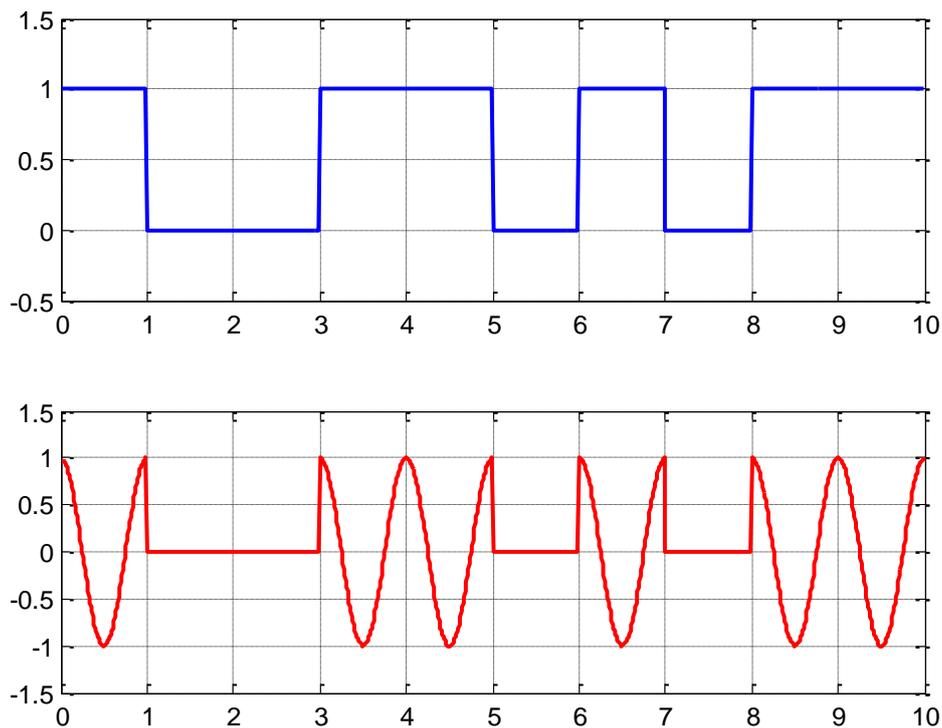
Dans une modulation ASK, à chaque symbole  $s_i$  est associée une amplitude  $A_i$ . Ainsi le signal modulé ASK s'écrit :

$$u(t) = \sum_k \alpha_k \cos(2\pi f_c t) r(t - kT) \quad (2.9)$$

Où  $r(t)$  est une impulsion d'énergie finie. Le cas le plus simple correspond à une impulsion rectangulaire de durée  $T$ , et  $\alpha_k$  représente l'amplitude  $A_i$  émise dans l'intervalle  $kT$ . On peut écrire :

$$u(t) = z_i(t) \cos(2\pi f_c t) \quad (2.10)$$

Où  $z_i(t)$  est un signal PAM NRZ à M niveaux, ainsi le signal modulé ASK s'obtient par une modulation d'amplitude analogique de la porteuse par un signal NRZ à M niveaux.



**Figure 2.2** Modulation 2-ASK (*On Off Keying*)

### 2.2.2 Modulation de phase (PSK: *Phase Shift Keying*)

Dans une modulation PSK, à chaque symbole  $s_i$  est associée une phase  $\varphi_i$ . Le signal modulé PSK s'écrit :

$$u(t) = \sum_k \cos(2\pi f_c t + \varphi_k) r(t - kT) \quad (2.11)$$

Où  $\varphi_k$  représente la phase émise dans l'intervalle  $k$ . Cette expression peut se réécrire sous la forme :

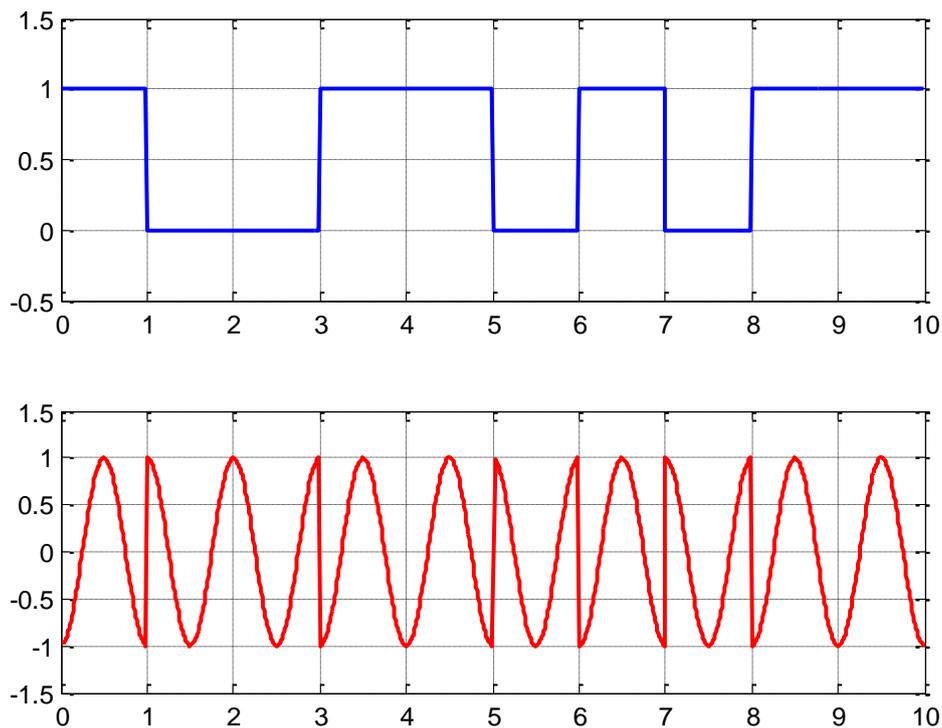
$$u(t) = z_I(t) \cos(2\pi f_c t) - z_Q(t) \sin(2\pi f_c t) \quad (2.12)$$

Avec :

$$z_I(t) = \sum_k \cos(\varphi_k) r(t - kT) \quad (2.13)$$

$$z_Q(t) = \sum_k \sin(\varphi_k) r(t - kT) \quad (2.14)$$

Le signal  $u(t)$  modulé PSK est obtenu en combinant linéairement deux modulations d'amplitude en quadrature : le signal  $z_I(t)$ , modulant la porteuse  $\cos(2\pi f_c t)$ , et le signal  $z_Q(t)$ , modulant la porteuse en quadrature (déphasée de  $\pi/2$ )  $\sin(2\pi f_c t)$ .



**Figure 2.3** Modulation 2-PSK

Les deux modulations précédentes sont combinées, et produisent une modulation M-aire de phase et d'amplitude s'appelle la modulation **M-QAM** (*Quadratique Amplitude Modulation à M niveaux*), on associe ici à chaque symbole  $s_i$  une amplitude  $A_i$  et une phase  $\varphi_i$ . le signal modulé à M niveaux d'amplitude et de phase, souvent appelé M-QAM s'écrit :

$$u(t) = \sum_k \alpha_k \cos(2\pi f_c t + \varphi_k) r(t - kT) \quad (2.15)$$

$u(t)$  peut se réécrire sous la forme de l'équation (2.12) avec :

$$z_I(t) = \sum_k \alpha_k \cos(\varphi_k) r(t - kT) \quad (2.16)$$

$$z_Q(t) = \sum_k \alpha_k \sin(\varphi_k) r(t - kT) \quad (2.17)$$

Où  $\alpha_k$  et  $\varphi_k$  représentent l'amplitude  $A_i$  et la phase  $\varphi_i$  émises dans l'intervalle  $k$ . Il est à noter que les modulations QAM sont des modulations par saut d'amplitude ou de phase ou d'amplitude et de phase combinées. Quand seule l'amplitude de la porteuse porte l'information, il s'agit d'une ASK. Quand seule la phase de la porteuse porte l'information, il s'agit d'une PSK. Lorsque l'information est transmise par des sauts de phase et d'amplitude, combinées, on l'appelle modulation QAM.

### 2.2.3 Démodulation cohérente d'une modulation M-QAM

On appellera démodulation, l'opération consistant à ramener en bande de base le signal reçu, c'est-à-dire récupérer les signaux  $Z_I(t)$  et  $Z_Q(t)$ . On n'inclura pas dans le terme démodulation, la notion de détection, c'est-à-dire la décision sur le symbole reçu.

On appelle démodulation cohérente, une modulation effectuée en utilisant la connaissance parfaite de la porteuse (en amplitude, fréquence et en phase).

Pour démoduler de façon cohérente, on peut opérer de deux manières : soit par translation en fréquence puis filtrage passe-bas, soit par filtrage de Hilbert puis translation en fréquence.

### 2.2.4 Modulation de fréquence (FSK: *Frequency Shift Keying*)

Dans ce cas c'est la fréquence instantanée du signal modulé qui peut prendre un certain nombre de valeurs associées aux états possibles de l'information à transmettre. Le signal modulé en 2-FSK s'écrira donc sous la forme :

$$u(t) = A \cos(2\pi f_0 t + \varphi(t)) \quad (2.18)$$

$\varphi(t)$  est la phase à l'instant  $t$ . La fréquence instantanée  $f(t)$  est obtenue par dérivation de la phase par rapport au temps et division par  $2\pi$ , soit :

$$f(t) = f_0 + \frac{1}{2\pi} \frac{d\varphi}{dt} \quad (2.19)$$

Le second terme du second membre de (2.11) est la déviation de la fréquence par rapport à sa valeur moyenne, égale à  $f_0$ . Compte tenu que cette déviation prend les valeurs  $\pm \Delta f / 2$  selon l'élément binaire transmis, on pourra encore écrire :

$$\frac{1}{2\pi} \frac{d\varphi}{dt} = \frac{\Delta f}{2} \sum_k \alpha_k r(t - kT) \quad (2.20)$$

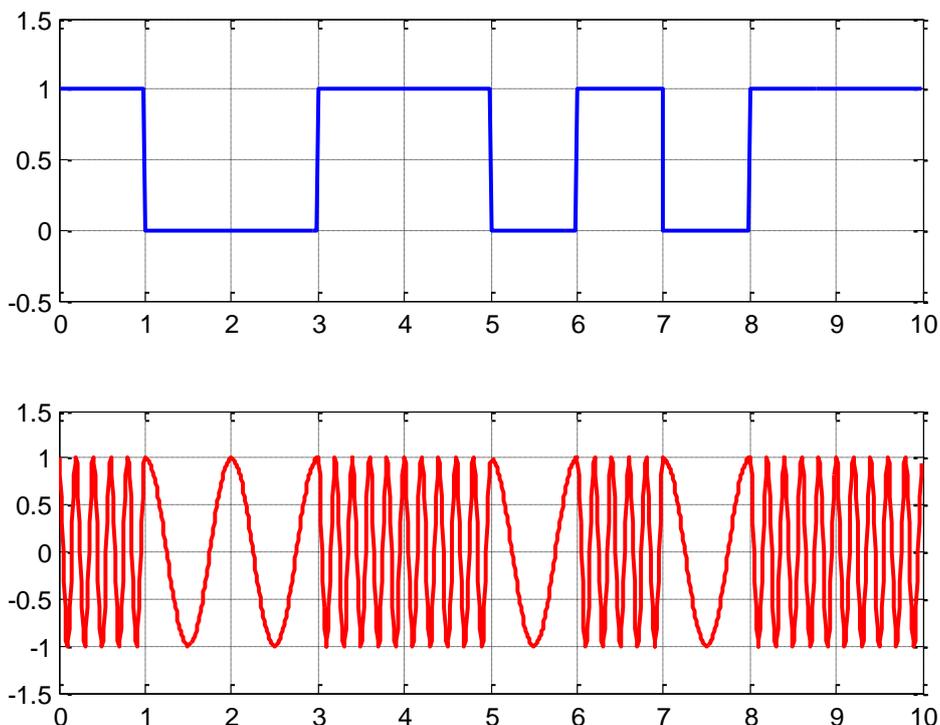
$\alpha_k$  prenant les valeurs  $\pm 1$ , soit encore, introduisant l'indice de modulation  $m$  :

$$\frac{d\varphi}{dt} = \pi \cdot \Delta f \sum_k \alpha_k r(t - kT) = \frac{m\pi}{T} \sum_k \alpha_k r(t - kT) \quad (2.21)$$

Puisque la phase est la primitive de la fréquence, elle n'est définie qu'à une constante d'intégration près. Partant de l'équation (2.21), on obtient par intégration :

$$\varphi(t) = \pi \cdot \Delta f \cdot \alpha_k (t - kT) + \theta_k = \frac{m\pi}{T} \alpha_k (t - kT) + \theta_k \quad (2.22)$$

Avec  $t \in [kT, (k+1)T[$ , l'équation (2.22) montre donc que la phase varie linéairement sur l'intervalle  $t \in [kT, (k+1)T[$  et que la variation est de  $\pm m\pi$ , selon la valeur de l'élément binaire transmis. La constante  $\theta_k$  est déterminée par les conditions initiales, c'est-à-dire qu'elle est égale à  $\varphi(kT)$ .



**Figure 2.4** Modulation 2-FSK

On sera amené à distinguer deux cas importants, *les modulations à phase discontinue* et *les modulations à phase continue*, telles que, la modulation *CPFSK* (*Continuous Phase Frequency Shift Keying*), la modulation *MSK* (*Minimum Shift Keying*) qui est une modulation *CPFSK* d'indice de modulation  $m=0.5$ . Son nom provient de l'utilisation de l'écart ( $\Delta f$ ) minimum permettant d'obtenir des fréquences orthogonales, et la modulation *GMSK* (*Gaussian Minimum Shift Keying*) qui est une modulation *MSK* à laquelle on a ajouté un filtre passe-bas gaussien dans le but de diminuer l'occupation spectrale du signal modulé.

Il y a aussi les modulations multiporteuses. Leur nom provient de l'utilisation de plusieurs porteuses ensemble, on cite entre autres la modulation *OFDM*.

## 2.3 Canal de transmission

### 2.3.1 Modèles mathématiques des canaux de transmission

2.3.1.1 *Canal à bruit additif* : Le modèle mathématique le plus simple pour un canal de communication, **figure. 2.5.a**. Dans ce modèle, le signal transmis est affecté par un bruit aléatoire additif. Physiquement, le bruit additif peut résulter de composants électroniques et des amplificateurs au niveau du récepteur, dans ce cas le bruit peut être caractérisé comme un bruit thermique, statistiquement, caractérisé comme un bruit gaussien, par conséquent le modèle mathématique résultant est souvent appelé, canal à bruit additif gaussien, ou des interférences rencontrées dans la transmission, comme dans le cas de transmission du signal radio. Le signal à la sortie du canal est donné par :

$$y(t) = x(t) + n(t) \quad (2.23)$$

2.3.1.2 *Canal à filtre linéaire avec bruit additif* : Dans certains canaux physiques, tels que lignes filaires téléphoniques, les filtres sont utilisés pour s'assurer que les signaux transmis ne dépassent pas les limites d'une bande passante spécifiée et afin de ne pas interférer avec des autres signaux. Ces canaux sont généralement caractérisés mathématiquement comme des canaux à filtre linéaire avec bruit additif, **figure. 2.5.b**. Ainsi,

$$y(t) = x(t) * h(t) + n(t) \quad (2.24)$$

2.3.1.3 *Canal à filtre linéaire variant dans le temps avec bruit additif* : les canaux physiques qui résultent d'une propagation à trajets multiples du signal émis, peuvent être mathématiquement caractérisés par un filtre linéaire variant dans le temps de réponse impulsionnelle  $h(\tau, t)$ , où  $h(\tau, t)$  est la réponse du canal à l'instant  $t$  de l'impulsion lui appliquée à l'instant  $t - \tau$ , le canal à filtre linéaire variant dans le temps avec bruit additif est illustré sur la **figure 2.5.c**, la sortie de tel canal pour une entrée  $x(t)$  est :

$$y(t) = x(t) * h(\tau, t) + n(t) \quad (2.25)$$

Une bonne représentation d'une propagation à trajets multiples du signal à travers un canal physique est le cas où :

$$h(t, \tau) = \sum_{k=1}^L a_k(t) \delta(t - \tau_k) \quad (2.26)$$

Avec  $a_k(t)$  représente la variation dans le temps du facteur d'atténuation sur les  $L$  trajets de propagation, et  $\tau_k$  sont les retards correspondants.

Les modèles mathématiques décrits ci-dessus permettant de caractériser la grande majorité des canaux physiques rencontrés dans la pratique.

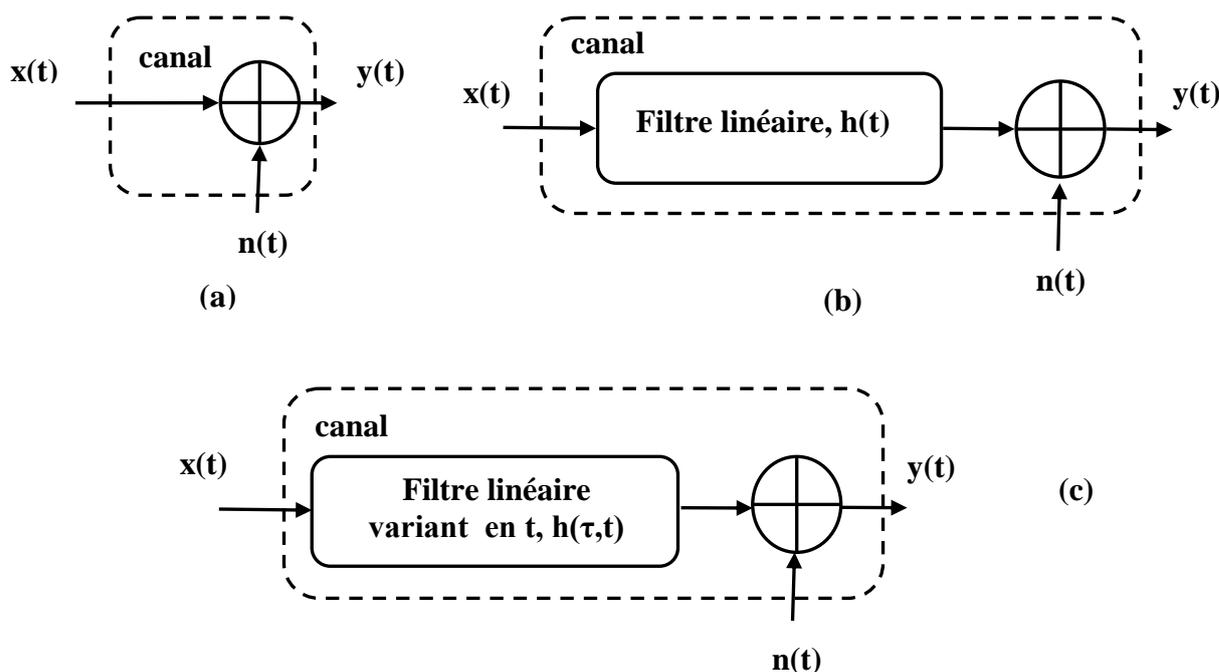


Figure 2.5 Types de canaux de transmission.

Un autre type de canaux de transmission, s'appelle les canaux non linéaires. La non linéarité est due essentiellement aux amplificateurs de puissance qui représentent en général les éléments clé d'un système embarqué (téléphones mobiles, satellites ...), car ce sont les principaux consommateurs d'énergie. Les meilleurs rendements pour un amplificateur de puissance sont obtenus en zone saturée, c'est à dire lorsque l'amplificateur est proche de son niveau de puissance maximum. Cependant dans de telles conditions, la non linéarité introduite par l'amplificateur dégrade l'information transportée par le signal hyperfréquence.

### 2.3.2 Perturbations apportées par le canal de transmission

2.3.2.1 *Le bruit blanc gaussien additif* : qui s'ajoute au signal utile, est la première source de dégradation. Le terme « blanc » signifie que toutes les fréquences sont égales en puissance, le terme « gaussien » décrit la distribution de l'amplitude. Ces deux concepts sont indépendants, en effet, on peut avoir l'un sans l'autre. Le bruit est omniprésent et peut provenir soit du milieu de transmission lui même (interférences venant d'autres systèmes radios, perturbations de lignes à haute tension, etc.) ou bien des équipements électroniques de réception. La robustesse de la transmission face au bruit est caractérisée par le rapport signal à bruit [Signal to Noise Ratio (SNR)],  $E_b/N_0$   $E_b$  : l'énergie reçue par élément binaire,  $N_0$  : la densité spectrale de mono latérale du bruit.

2.3.2.2 *L'évanouissement (fading)* : La façon dont se propage le signal dans le milieu est également source de distorsions. Le signal émis subit des réflexions autour des obstacles se trouvant dans l'environnement de propagation entre la source et la destination, ces réflexions engendrent des évanouissements profonds de la puissance du signal reçu, diminuant le SNR et affaiblissant la résistance au bruit. Ces

atténuations peuvent être contrées par des techniques dites de diversité [3]. Par ailleurs, la propagation par trajets multiples introduit une sélectivité en fréquence du canal, certaines fréquences du signal étant atténuées par rapport à d'autres. Dans le domaine temporel, ceci se traduit par un brouillage du message dû à la réception de plusieurs copies du signal utile différemment retardées.

2.3.2.3 *L'interférence entre symboles* : soit le système de transmission suivant, avec une modulation de type QAM (liaison modélisée en bande de base), ou PAM (*Pulse Amplitude Modulation*),

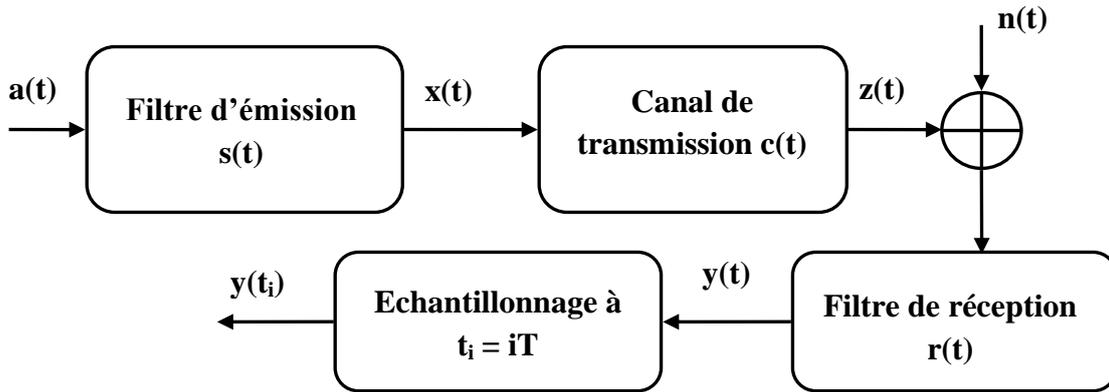


Figure 2.6 Transmission en bande de base d'un signal QAM ou PAM.

La sortie du filtre de réception s'écrit :

$$y(t) = \sum_i \alpha_i p(t - iT) + w(t) \quad (2.27)$$

Avec :

$$a(t) = \sum_i \alpha_i \delta(t - iT) \quad (2.28)$$

$$p(t) = s(t) * c(t) * r(t) \text{ et } w(t) = n(t) * r(t) \quad (2.29)$$

La sortie  $y(t)$  est échantillonnée aux instants  $t_i = iT_i$  ce qui donne :

$$y(t_i) = p(0)\alpha_i + \sum_{k \neq i}^{+\infty} \alpha_k p((i - k)T) + w(t_i) \quad (2.30)$$

Dans cette expression :

- Le terme  $p(0)\alpha_i$  représente la contribution du  $i^{\text{ème}}$  symbole transmis.
- Le second terme représente l'effet résiduel de tous les autres symboles transmis sur le décodage du  $i^{\text{ème}}$  symbole. Cet effet résiduel dû aux impulsions arrivant avant et après l'instant  $t_i$  est appelé *Interférence Entre Symboles* (IES).

- Le dernier terme  $w(t_i)$  représente le bruit à l'instant  $t_i$ .

En absence du bruit et d'IES,  $y(t_i) = p(0)\alpha_i$ , ce qui montre que dans les conditions idéales, le  $i^{\text{ème}}$  symbole est décodé correctement ( si  $p(0) \neq 0$  ) [4].

#### Caractérisation de l'IES : Diagramme de l'œil

La quantité d'IES affectant une communication peut être visualisée en observant le faisceau de tracés du signal reçu sur un multiple de la durée symbole, c'est-à-dire la superposition des intervalles  $[y(iT), y((i+k)T)]$ .

Le résultat visuel s'appelle *diagramme de l'œil*, en raison de sa ressemblance avec un œil humain, pour la transmission de symboles binaires.

Ce diagramme de l'œil fournit des informations intéressantes sur les performances de la chaîne de transmission.

La hauteur de l'œil (ouverture verticale) définit la marge dont on dispose en termes de bruit additif. En effet, la présence de bruit causera une erreur de décision, avec une probabilité d'autant plus grande que cette hauteur sera petite. L'instant d'échantillonnage correspond à l'abscisse de plus grande ouverture.

La largeur de l'œil (ouverture horizontale) donne la marge dont on dispose en écart de temps entre l'instant d'échantillonnage idéal et un autre instant.

La pente de la fermeture (ou ouverture) de l'œil détermine la sensibilité à une erreur par rapport à cet instant optimal.

Lorsque l'effet de l'IES est très grave, l'œil est complètement fermé. Il faut alors trouver une technique pour corriger cet effet, cette technique s'appelle *l'égalisation*.

#### Condition d'absence d'IES – Critère de Nyquist

A partir de l'équation (2.30), il est facile de déterminer la condition sur  $p(t)$  pour laquelle il n'y aura pas d'IES (en absence de bruit). Cette condition est appelée *critère de Nyquist en temps*.

$$p((i-k)T) = \begin{cases} p(0) & \text{pour } i = k \\ 0 & \text{pour } i \neq k \end{cases} \quad (2.31)$$

Si cette condition est remplie, on aura  $y(t_i) = p(0)\alpha_i$ , et il n'y a pas d'IES. Le filtre  $p(t)$  représente le canal total (depuis la source des symboles  $\alpha_i$  jusqu'à l'entrée de l'organe de décision du récepteur) est dit canal de Nyquist s'il vérifie la condition (2.31).

Considérons le signal  $p_e(t)$  représentant la réponse impulsionnelle du canal  $p(t)$  échantillonnée tous les  $T$  :

$$p_e(t) = \sum_{n=-\infty}^{+\infty} p(nT) \delta(t - nT) \quad (2.32)$$

La transformée de Fourier de ce signal  $p_e(t)$  s'écrit :

$$P_e(f) = \frac{1}{T} \sum_{k=-\infty}^{+\infty} P(f - \frac{k}{T}) = \int_{-\infty}^{+\infty} \left( \sum_{n=-\infty}^{+\infty} p(nT) \delta(t - nT) \right) e^{-j2\pi ft} dt \quad (2.33)$$

Le critère de Nyquist en temps implique :

$$P_e(f) = \int p(0) \delta(t) e^{-2j\pi ft} dt = p(0) \quad (2.34)$$

Soit la nouvelle condition :

$$\sum_{k=-\infty}^{+\infty} P(f - \frac{k}{T}) = p(0) T \quad (2.35)$$

La réponse fréquentielle  $P(f)$  élimine l'IES sur des échantillons pris à intervalles constants  $T$  si elle vérifie la condition (2.35) appelée *critère de Nyquist en fréquence*. La réponse repliée du canal correspondant à l'échantillonnage en temps au rythme symbole doit être constante. Pour l'interprétation du critère de Nyquist pour un canal de largeur de bande  $B$  ( $H(f)=0$  pour  $|f| > B$ ). On distingue trois cas [4] :

- $\frac{1}{T} > 2B$ . Le spectre consiste en répliques de  $P(f)$  qui ne se recouvrent pas. On ne peut pas concevoir  $P(f)$ , et donc  $S(f)$  et  $R(f)$ , tels qu'il n'y ait pas d'IES. Autrement dit si le rythme symbole est supérieur à deux fois la largeur de bande, il y a nécessairement de l'IES et il faut avoir recours à des techniques d'égalisation pour compenser ces effets.
- $\frac{1}{T} = 2B$ . Le critère de Nyquist n'est vérifié que pour  $P(f) = p(0) T$ .  $1/T$  est le débit maximum (appelé débit de Nyquist) que l'on peut espérer transmettre sans IES sur un canal de largeur  $B$ . Inversement  $B=1/2T$  (largeur de bande de Nyquist) est la largeur de bande minimale pour pouvoir transmettre sans IES une information numérique au débit  $1/T$ . Bien que ce canal de Nyquist permette d'éliminer l'IES avec une largeur de bande minimale, il y a deux difficultés pratiques qui le rendent non utilisable dans la conception des systèmes :
  - Le gabarit parfaitement rectangulaire de  $P(f)$  n'est pas réalisable ;

- La fonction  $p(t)$  décroît en  $1/|t|$  pour les grandes valeurs de  $|t|$ , ce qui entraîne une erreur importante si on commet une petite erreur d'échantillonnage.
- $\frac{1}{T} < 2B$ . Le spectre consiste en répliques de  $P(f)$  avec recouvrement et il existe de nombreux choix possibles sur  $P(f)$  pour assurer le critère de Nyquist en fréquence. Généralement, on peut considérer que  $P(f)=0$  pour  $|f| > 1/T$  (c'est-à-dire  $B < 1/T$ ). la condition (2.35) peut alors d'écrire :

$$P(f) + P(f - \frac{1}{T}) = p(0) T \quad (2.36)$$

Une classe de fonctions  $P(f)$ , appelées fonctions en cosinus surélevé, sont largement utilisés en pratique dans les filtres d'émission et de réception des systèmes de communication numérique [4].

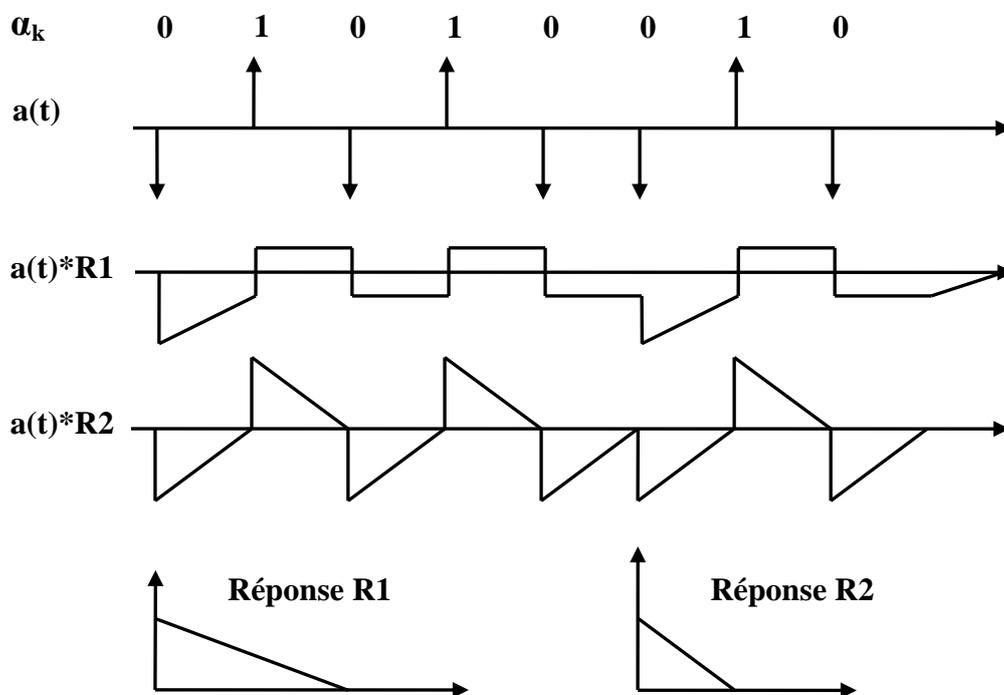


Figure 2.7 Illustration des interférences entre symboles

La figure 2.7 illustre schématiquement ce phénomène d'interférence entre symboles. La suite des valeurs émises est  $\{0, 1, 0, 1, 0, 0, 1, 0\}$ , la figure représente l'entrée  $a(t)$  du canal ainsi que sa sortie pour les deux réponses impulsionnelles  $R1$  et  $R2$ . La réponse  $R1$  s'étale sur exactement deux fois la durée symbole, elle induit par conséquent des interférences (il n'est plus possible de distinguer les différents symboles en sortie d'un canal de réponse  $R1$ ). La réponse  $R2$  quant à elle est juste inférieure à la durée maximale qui permet une transmission sans interférence : les symboles ne se recouvrent pas en sortie d'un filtre de réponse impulsionnelle  $R2$  [5].

## 2.4 Filtres numériques

Les filtres numériques sont utilisés pour deux raisons principales, la première est la séparation des signaux qui ont été combinés, la séparation est nécessaire quand un signal a été affecté par des interférences, de bruit, ou d'autres signaux, et la seconde raison est la restauration des signaux qui ont été déformés. Les filtres analogiques peuvent être utilisés pour ces mêmes tâches, mais, les filtres numériques aboutissent des résultats bien supérieurs. Selon l'implémentation, deux types sont distingués, les filtres récurrents, dits filtres à **Réponse Impulsionnelle Infinie RII**, et les filtres non récurrents, dits filtres à **Réponse Impulsionnelle Finie RIF** [6].

### 2.4.1 Les filtres à Réponse Impulsionnelle Finie (RIF)

La sortie d'un filtre numérique à réponse impulsionnelle finie peut s'exprimer comme une combinaison linéaire des échantillons présents à son entrée. Les coefficients de la combinaison linéaire constituent la réponse impulsionnelle du filtre. Un filtre RIF peut être décrit par l'équation suivante :

$$y(n) = \sum_{i=0}^N h(i)x(n-i) \quad (2.37)$$

Sa fonction de transfert est de la forme :

$$H(z) = \sum_{i=0}^N h(i) z^{-i} \quad (2.38)$$

La fonction de transfert  $H(z)$  possède  $N$  racines  $z_i$  dans le plan complexe, appelées zéros de la fonction de transfert, et peut s'écrire sous la forme d'un produit de facteurs :

$$H(z) = h(0) \prod_{i=1}^N (1 - z_i z^{-1}) \quad (2.39)$$

Un filtre RIF à réponse impulsionnelle de durée  $N$  est toujours stable, pour autant que toutes les valeurs  $h(i)$  de sa réponse impulsionnelle soient bornées. Pour cette raison les filtres RIF sont très souvent utilisés dans les systèmes d'égalisation. Un autre avantage des filtres RIF est que l'on peut les rendre causaux, donc physiquement réalisables, moyennant l'introduction d'un retard [7].

### 2.4.2 Les filtres à Réponse Impulsionnelle Infinie (RII)

Les filtres numériques à réponse impulsionnelle infinie RII, conservent une trace des échantillons qui leur ont été appliqués pendant une durée infinie, ils sont donc à mémoire infinie, d'où la dénomination courante de *filtre récurrent*. Un filtre RII est décrit par l'équation suivante :

$$y(n) = \sum_{l=0}^L a(l)x(n-l) - \sum_{m=1}^M b(m)y(n-m) \quad (2.40)$$

Chaque échantillon de sortie est égal à une combinaison linéaire des échantillons présents à l'entrée du filtre et des échantillons précédemment déterminés en sortie du filtre. La fonction de transfert d'un filtre RII se présente sous la forme suivante [7]:

$$H(z) = \frac{\sum_{l=0}^L a(l) z^{-l}}{1 + \sum_{m=1}^M b(m) z^{-m}} \quad (2.41)$$

Les racines du numérateur et du dénominateur sont respectivement appelées les zéros et les pôles de la fonction de transfert, ainsi la fonction de transfert peut s'écrire :

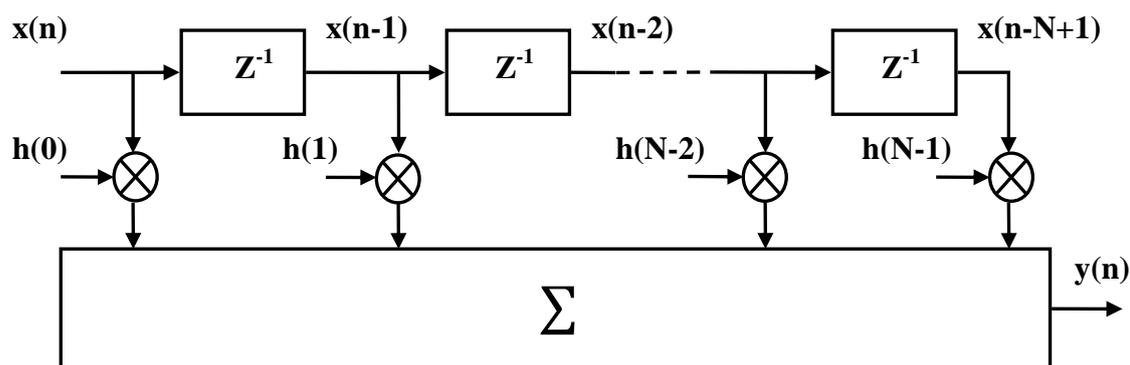
$$H(z) = a_0 \frac{\prod_{l=0}^L (1 - Z_l Z^{-1})}{\prod_{m=1}^M (1 - P_m Z^{-1})} \quad (2.42)$$

$Z_l$  et  $P_m$  dénotent les zéros et les pôles respectivement,  $a_0$  représente le facteur d'échelle qui définit le gain [8].

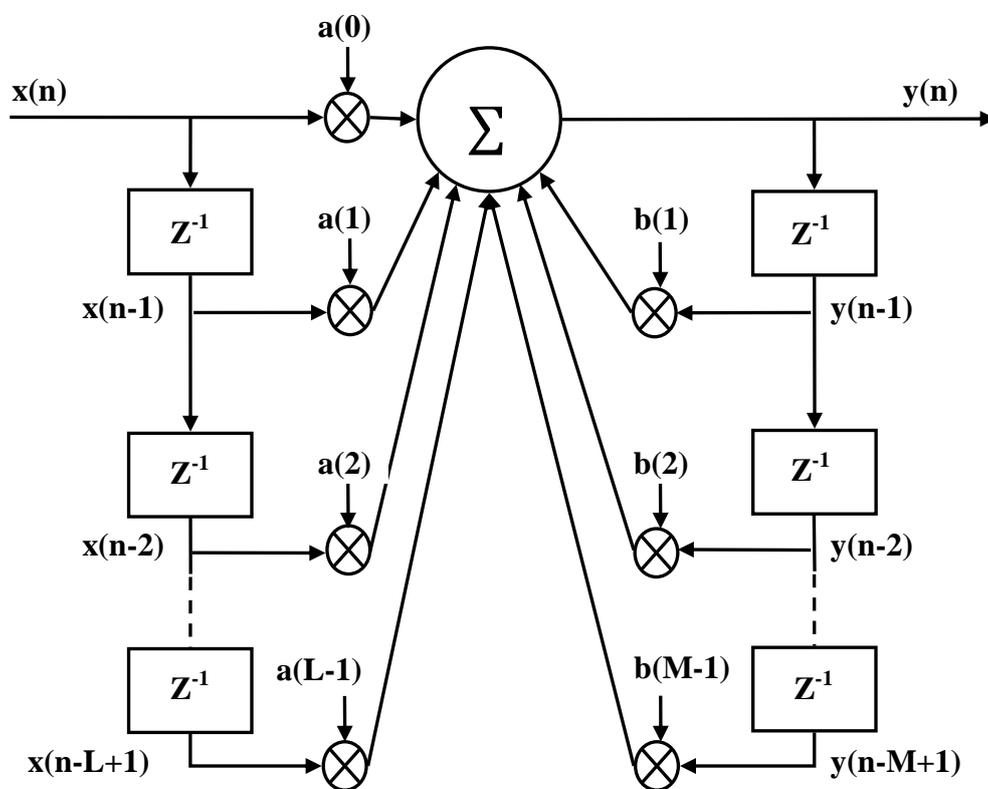
Pour qu'un filtre à réponse impulsionnelle infinie soit réalisable, il est nécessaire qu'il soit causal et stable. Un filtre causal est stable si les pôles de sa fonction de transfert en  $z$  sont à l'intérieur du cercle unité.

On notera qu'un filtre possédant tous ses pôles et ses zéros à l'intérieur du cercle unité est appelé *filtre à phase minimale*, tandis qu'un filtre possédant tous ses zéros à l'extérieur du cercle unité est appelé *filtre à phase maximale*.

La structure d'un filtre numérique est choisit en se basant sur plusieurs critères, dans certains cas, tels que pour les filtres RII, la stabilité dépend du choix convenable de cette structure [9]. La **figure 2.8** montre les filtres RIF et RII en structure directe.



(a)



(b)

Figure 2.8 Structure directe des filtres, (a) RIF, (b) RII

### 2.4.3 Comparaison entre les filtres RIF et RII

- Un filtre RIF peut atteindre une phase linéaire exacte, il n'introduit pas des distorsions sur la phase du signal. La réponse à phase linéaire est importante dans les applications telles que la transmission des données, et le traitement d'image. Par

contre un filtre RII possède généralement une réponse à phase non linéaire en particulier sur les limites de bande.

- La stabilité des filtres RIF est garantie alors qu'un tel garant n'est pas disponible pour les filtres RII.
- Les filtres RII sont plus souhaités lorsqu'un seuil de coupure est exigé. Un tel cas nécessite un filtre RIF d'ordre très élevé, ce qui implique un grand retard. Or, un filtre d'ordre supérieur possède plus de coefficients et par conséquent plus de mémoire et plus de coût en termes de calcul.
- Il est relativement facile de concevoir un filtre RIF avec une fonction de transfert arbitraire. Cependant, les filtres analogiques peuvent aisément se convertir en un filtre RII équivalent de mêmes spécifications. Ceci est un avantage pour les concepteurs qui veulent convertir des applications analogiques en numériques. Un filtre RIF n'a pas une contrepartie analogique.
- Les filtres RII sont plus sensibles aux erreurs d'arrondi et des erreurs de quantification que les filtres RIF.

En conclusion, si un seuil de coupure avec un haut débit (retard faible) sont demandés, les filtres RII sont les plus souhaitables. D'autre part, si la phase linéaire exacte est importante, les filtres RIF sont les plus conseillés. Par conséquent les filtres RIF sont le choix le plus couramment utilisé si le nombre de coefficients n'est pas trop grand en raison de leurs propriétés de calcul supérieures [9].

#### **2.4.4 Filtrage adaptatif**

Dans l'étude de l'égalisation, il semble pertinent d'avoir une idée au sens large sur le « filtrage adaptatif ». Le terme « adaptatif » peut être compris en considérant un système qui tente de lui ajuster pour répondre à certains phénomènes qui avaient lieu autour de lui ; en d'autres mots, le système tente d'ajuster leurs paramètres dans le but de répondre à certains objectifs bien définis, qui dépendent de l'état du système ainsi que son environnement. De plus, il est nécessaire d'avoir un ensemble des étapes ou une certaine procédure avec laquelle l'adaptation est accomplie. Le système qui réalise et subit la procédure d'adaptation est appelé « filtre ».

Evidemment, selon le temps nécessaire pour atteindre l'objectif final de l'adaptation, qui s'appelle le temps de convergence, la complexité, et les ressources disponibles pour accomplir l'adaptation, nous pouvons avoir une variété des algorithmes d'adaptation et des structures des filtres.

Les filtres adaptatifs, par leur nature même, sont des systèmes auto-concepteurs qui peuvent s'adapter à différents environnements. Par conséquent, ils trouvent des applications dans des domaines aussi divers que le contrôle, les communications, le traitement du signal radar et sonar, l'annulation des interférences, le contrôle actif du bruit, le génie biomédical, etc. La caractéristique commune de ces applications qui les regroupent sous la même formule de base de filtrage adaptatif est qu'elles impliquent toutes un processus de filtrage d'un signal d'entrée pour atteindre une réponse souhaitée [10].

les paramètres du filtre sont mis à jour en faisant un ensemble de mesures des signaux sous-jacents et l'application de cet ensemble à l'algorithme de filtrage adaptatif de telle sorte que l'écart entre la sortie du filtre et la réponse désirée est réduit au minimum soit dans les algorithmes statistiques ou déterministes. Dans ce contexte, quatre classes de base des applications du filtrage adaptatif sont reconnus. à savoir, la modélisation, la modélisation inverse, la prédiction linéaire, et l'annulation d'interférence [10].

## **2.5 Conclusion**

Dans ce chapitre nous avons présenté les notions de base de la communication numérique, en partant des éléments d'un système de transmission. Nous avons également examiné les techniques de base de modulation numérique telles qu'ASK PSK et FSK, en donnant les courbes de chaque modulation en deux états, et un petit récapitulatif sur ses variantes telles que la QAM, la MSK, et la GMSK. Les différents modèles mathématiques des canaux de transmission les plus rencontrés en pratique, sont également évoqués, avec les distorsions introduites par ces canaux, notamment l'interférence entre symboles qui a été bien détaillé en considérant qu'il constitue un problème dont son élimination fait l'objectif principal de l'égalisation. On a présenté aussi un aperçu sur les filtres numériques, leurs structures, leur stabilité, et leurs équations, le filtrage adaptatif est une notion très importante qu'on a évoqué aussi, tous les éléments cités précédemment, constitue les éléments de base qui vont nous aider à avoir une idée précise des égaliseurs adaptatifs classiques, l'objet du chapitre prochain.

# Chapitre 3

## Techniques classiques d'égalisation

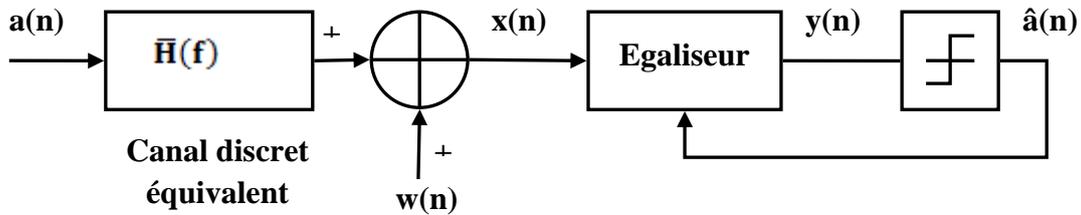
---

Dans ce chapitre nous présentons une étude des techniques classiques d'égalisation permettant de reconstituer la séquence de symboles transmise en éliminant les effets de distorsion qui peuvent altérer l'information utile. Nous montrons les caractéristiques des canaux utilisés dans les simulations, les principales structures d'égaliseurs conventionnels tels que l'égaliseur transversal linéaire LTE (Linear Transversal Equalizer) et l'égaliseur à retour de décision DFE (Decision Feedback Equalizer), ainsi que la solution d'égalisation à maximum de vraisemblance. Enfin nous comparons leurs performances respectives en termes de signal égalisé, de courbes de convergence de l'erreur quadratique moyenne MSE, de courbes de BER, et finalement, en comparant les diagrammes de l'œil.

### 3.1 Fonction d'égalisation

L'objet de la fonction d'égalisation est de permettre de retrouver à partir de la séquence reçue  $x(n)$ , présentant de l'interférence entre symboles (*IES*) introduite par la sélectivité du canal, la séquence émise  $a(n)$ .

Une chaîne de transmission numérique, en présence d'égalisation, peut être représentée par le schéma de principe de la **figure 3.1**.



**Figure 3.1** Chaîne de transmission en présence d'égalisation

L'ensemble modulateur, milieu de transmission, démodulateur est modélisé par un **canal discret équivalent de réponse  $H(z)$**  [6]. Cette modélisation correspond au cas où les données  $d[n]$  sont émises tous les  $T$  secondes et le signal reçu en sortie du démodulateur est échantillonné à la fréquence  $1/T$ . La quantité  $T$  est appelée la **durée symbole** et  $1/T$  représente la rapidité de modulation.

En principe, si le canal est parfaitement connu, il est possible de rendre l'interférence entre symboles arbitrairement faible, ou même de l'éliminer complètement, en utilisant une paire de filtres d'émission et de réception tels que :

$$\sum_n G(f + n/T_s) C(n + n/T_s) H(f + n/T_s) = T_s \quad (3.1)$$

c'est-à-dire que la chaîne complète vérifie le critère de **Nyquist** (le filtre global  $G(f)C(f)H(f)$  est un filtre de **Nyquist**). En pratique, cependant, on ne connaît que très rarement les caractéristiques exactes du canal, et tout au plus des valeurs moyennes. Par ailleurs, il subsiste des erreurs dans la correction de l'interférence entre symboles, en raison des imperfections sur l'implantation des filtres  $g$  et  $h$ . Enfin, le canal peut ne pas être stationnaire, c'est-à-dire que ses caractéristiques varient au cours du temps. L'effet de ces différents facteurs est une interférence entre symboles (éventuellement résiduelle), qu'il faut compenser, à l'aide d'un dispositif appelé *égaliseur*, ce dispositif pouvant également être variable dans le temps.

En bande de base, si les filtres d'émission et de réception sont fixés (et forment ensemble un filtre de Nyquist), le rôle de l'égaliseur est simplement de compenser la réponse du canal. En notant ainsi  $e(t)$  la réponse impulsionnelle de l'égaliseur et  $E(f)$  sa fonction de transfert, il faut obtenir :

$$E(f) = \frac{1}{C(f)}, \text{ sur } [-B, B] \quad (3.2)$$

Comme la paire des filtres d'émission et de réception forment un filtre de **Nyquist**, on peut parfaitement reconstituer la séquence d'entrée  $a(n)$ . Du point de vue spectral, on a « égalisé » la réponse du canal [4].

### 3.2 Différents types d'égaliseurs

Les techniques d'égalisation se divisent en deux grandes catégories: linéaires et non linéaires. Les techniques linéaires sont généralement les plus simples à mettre en œuvre. Cependant, les techniques d'égalisation linéaire souffrent généralement de l'amplification du bruit plus que les égaliseurs non linéaires, et ne sont donc pas utilisées dans la plupart des applications sans fil. Parmi les techniques d'égalisation non linéaire, l'égaliseur à retour de décision (DFE) est le plus populaire, car il est simple à mettre en œuvre et présente généralement des bons résultats. Toutefois, sur les canaux à faible SNR, le DFE souffre de la propagation d'erreur lorsque les bits sont décodés faussement, conduisant à des performances médiocres. La technique d'égalisation optimale est l'estimation de séquence au sens du maximum de vraisemblance (MLSE, pour Maximum Likelihood Sequence Estimation). Malheureusement, la complexité de cette technique se développe de façon exponentielle avec la longueur du canal, elle est donc inutilisée sur la plupart des canaux d'intérêt. Toutefois, la performance du MLSE est souvent utilisée comme une borne supérieure pour les autres techniques d'égalisation. La **figure 3.2** résume les différents types d'égaliseurs, avec leurs structures correspondantes et les algorithmes d'adaptation.

Les égaliseurs peuvent aussi être classés comme détecteurs symbole par le symbole (SBS, pour Symbol-By-Symbol) ou estimateurs de séquence (SE, pour Sequence Estimator). Les égaliseurs SBS retirent l'IES de chaque symbole et ensuite détectent chaque symbole individuellement. Tous les égaliseurs linéaires dans la **figure 3.2** ainsi que le DFE sont égaliseurs SBS. Les égaliseurs SE détectent toute la séquence de symboles, donc l'effet de l'IES fait partie du processus d'estimation. L'estimation de séquence au sens du maximum de vraisemblance (MLSE) est la forme optimale de la détection de séquence, mais elle est très complexe.

Les égaliseurs linéaires et non linéaires sont typiquement implémentés en utilisant une structure transversale ou en treillis. La structure transversale est un filtre à  $N - 1$  éléments de retard et  $N$  coefficients complexes accordable. Le filtre en treillis utilise une structure récursive plus complexe. En échange de cette complexité accrue par rapport aux structures transversales, les structures en treillis sont souvent meilleures en termes de stabilité numérique et en termes de convergence, et elles disposent d'une grande flexibilité pour modifier leur longueur. Dans ce chapitre on se concentre sur les structures transversales.

En plus du type d'égaliseur et de sa structure, les égaliseurs adaptatifs nécessitent des algorithmes d'adaptation des coefficients de filtre pendant l'apprentissage et le mode opérationnel. De nombreux algorithmes ont été développés au cours des années à cet effet. Ces algorithmes impliquent généralement des compromis entre la complexité, le taux de convergence et la stabilité numérique [10].

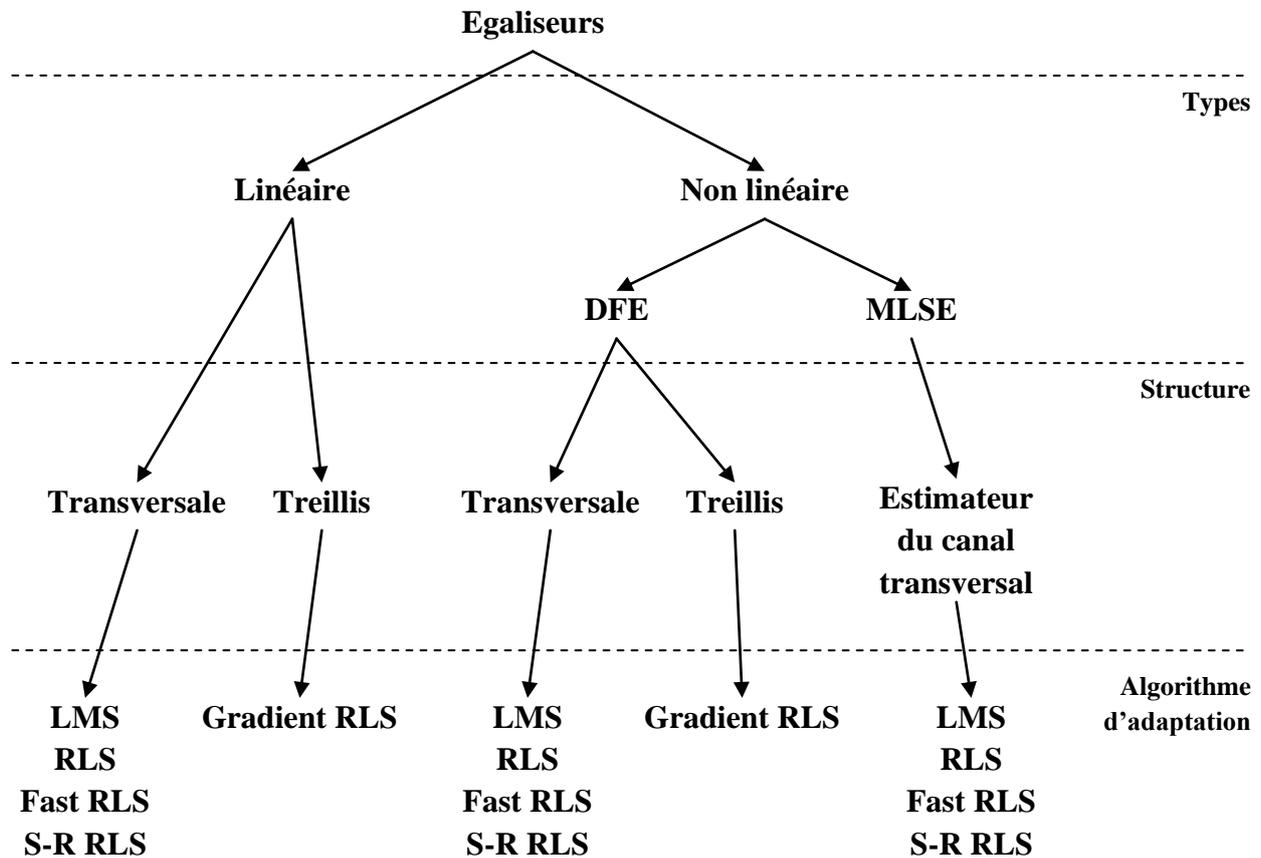


Figure 3.2 Différents types, structures, et algorithmes d'égaliseurs.

### 3.3 Egaliseur transverse

Les égaliseurs transverses sont les plus simples à mettre en œuvre. En effet, il s'agit simplement d'utiliser un filtre numérique à réponse impulsionnelle finie. La structure du filtre est donnée sur la **figure 3.3** et correspond à la relation :

$$y(n) = \sum_{i=0}^{M-1} e(i)x(n-i) \quad (3.3)$$

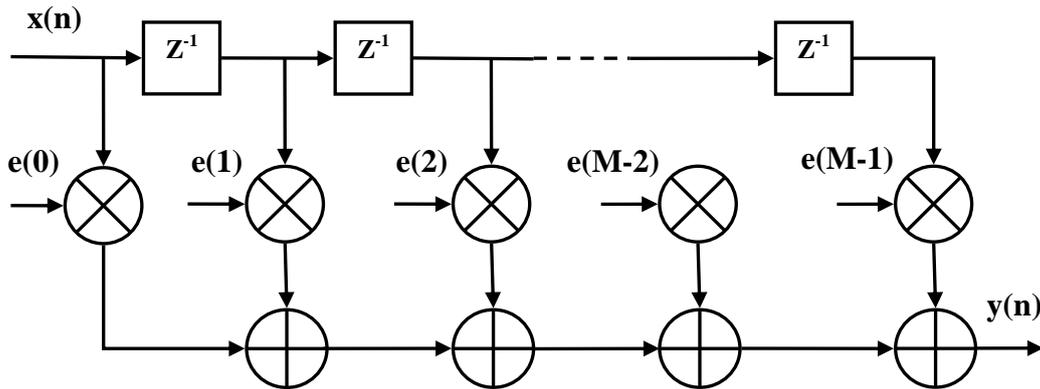


Figure 3.3 Structure d'un égaliseur transverse

où  $e(n)$  est la réponse impulsionnelle de l'égaliseur, de longueur  $M$ ,  $x(n)$  est la séquence d'observations, et  $y(n)$  la sortie de l'égaliseur.

### 3.3.1 Critères d'optimisation

#### 3.3.1.1 L'égaliseur « Zero Forcing »

La solution *Zero Forcing* tente d'inverser exactement la fonction de transfert du canal, ce qui est *a priori* le but recherché par l'égalisation. Ainsi, l'interférence entre symboles est exactement compensée (pourvu que l'ensemble des filtres d'émission et de réception soit **Nyquist**), et l'on dit que l'interférence entre symboles est forcée à zéro. On alors :

$$E(z) = \frac{1}{C(z)} \quad (3.4)$$

Où  $E(z)$  est la fonction de transfert du filtre *RIF* égaliseur, et  $C(z)$  la fonction de transfert du canal.

Dès maintenant, on peut s'apercevoir que cette démarche souffre de deux défauts : d'abord,  $C(z)$  peut posséder des zéros de module supérieur à 1, ce qui induit des pôles instables pour  $E(z)$ , si celui-ci doit être causal ; d'autre part, si  $c(n)$  la réponse impulsionnelle du canal est finie, alors  $e(n)$  la réponse impulsionnelle de l'égaliseur est infinie. On peut tourner partiellement la première difficulté en introduisant un retard  $R$  lors de la résolution ce qui permet de prendre en compte une éventuelle partie non causale (mais retardée), et tient également compte du nécessaire retard lié à la mise en œuvre des filtres. Le choix de ce retard est à la fois important et difficile. Globalement, on peut dire que le retard engendré par les deux filtres est égal à la moitié de la longueur du filtre équivalent, longueur qui vaut ici  $N + M - 1$ , où  $N$  la longueur du canal, et  $M$  la longueur de l'égaliseur.

D'autre part, on choisit de prendre un ordre  $M$  assez grand pour que l'éventuelle erreur de troncature de la réponse impulsionnelle soit peu sensible. Dans ces

conditions, on peut écrire l'équation de convolution correspondant à l'inversion du canal :

$$\sum_{i=0}^{M-1} e(i)c(n-i) = \begin{cases} 1, n = R \\ 0, n \neq R \end{cases}, R = \frac{N + M - 1}{2} \quad (3.5)$$

Cette relation de convolution étant valable pour tout  $n$ , on peut se donner  $M$  équations, par exemple pour  $n = 0..M - 1$ , et résoudre exactement le système linéaire correspondant pour obtenir les coefficients de l'égaliseur  $e(i)$ ,  $i = 0..M - 1$ . À la sortie de l'égaliseur, on obtient alors

$$y(n) = a(n - R) + b'(n) \quad (3.6)$$

Où  $a(n-R)$  la séquence de symboles est stationnaire, centrée, et blanche, et  $b'(n)$  est le bruit d'observation filtré par l'égaliseur. L'annulation des interférences entre symboles se fait généralement au prix d'une augmentation sensible du niveau de bruit. En effet, la fonction de transfert du canal est en général de type passe-bas, et son inverse est de type passe haut. Lorsque le bruit est large bande, il s'en suit une forte augmentation du bruit en haute fréquence et une dégradation du rapport signal-à-bruit. Ainsi, en dehors du cas où l'on est assuré d'un faible niveau de bruit d'observation, cette solution n'est pas à retenir. On notera en outre que le canal est supposé parfaitement connu ; dans cette méthode *supervisée*, on devra alors passer par une estimation préalable de la réponse impulsionnelle du canal. Les erreurs d'estimation de la réponse impulsionnelle se répercuteront alors sur les coefficients de l'égaliseur et entraîneront une dégradation des performances [4].

### 3.3.1.2 L'égaliseur à erreur quadratique moyenne minimale

Alors que l'égaliseur à *Zero Forcing* résout le problème en faisant abstraction du bruit d'observation, l'idée de l'égaliseur à erreur quadratique minimale, EQM (en anglais, MSE, pour Mean Square Error) est de minimiser l'erreur quadratique entre la séquence d'entrée (symboles) et la sortie de l'égaliseur. Le bruit est ainsi pris en compte dans le critère. On cherche ainsi à minimiser l'erreur quadratique moyenne :

$$J(e) = E \left[ |y(n) - a(n - R)|^2 \right] \quad (3.7)$$

Avec :

$$y(n) = \sum_{i=0}^{M-1} e(i)x(n-i) = e^T x(n) \quad (3.8)$$

Où :

$$e^T = [e(0)..e(M-1)] \quad (3.9)$$

Et :

$$x(n)^T = [x(n) \dots x(n-M+1)] \quad (3.10)$$

On a :

$$J(e) = E \left[ \left| e^T x(n) - a(n-R) \right|^2 \right] \quad (3.11)$$

Ainsi, la minimisation de l'erreur quadratique moyenne est obtenue pour :

$$\frac{\partial J(e)}{\partial e} = 2E \left[ x(n) \left( e^T x(n) - a(n-R) \right) \right] = 0 \quad (3.12)$$

Soit :

$$E \left[ x(n)x(n)^T \right] e = E \left[ x(n)a(n-R) \right] \quad (3.13)$$

On tire de cela :

$$R_{xx}e = R_{xa}(R) \quad (3.14)$$

Et :

$$e = R_{xx}^{-1}R_{xa}(R) \quad (3.15)$$

Où  $R_{yy}$  est la matrice de corrélation de  $x(n)$  et  $R_{xa}(R)$  est le vecteur d'intercorrélation entre  $x(n)$  et  $a(n-R)$ . Le nombre de coefficients du filtre est nécessairement limité. Si l'égalisation obtenue est clairement de meilleure qualité que celle fournie par un *Zero Forcing*, en raison de la prise en compte effective du bruit, elle reste souvent de qualité médiocre, en particulier en présence d'évanouissements sélectifs (non stationnarités). Ceci est également lié à la structure transverse (pas de pôles) qui limite la capacité de représentation d'une réponse quelconque. Par ailleurs, pour la mise en œuvre pratique, il est nécessaire de connaître  $a(n-R)$ . Pour ce faire, on utilise une séquence connue du récepteur, une *séquence d'apprentissage*, pour calculer les coefficients du filtre. La nécessité d'inclure dans l'émission une séquence d'apprentissage, éventuellement répétée périodiquement si le système est non stationnaire, limite en outre le débit en données utiles [4].

Le critère d'optimisation le plus utilisé est celui de l'erreur quadratique moyenne minimale. Les avantages de ce critère d'optimisation sont les suivants [5] :

- Lorsque le canal est bruité, ce critère réalise un compromis entre deux objectifs contradictoires : éliminer les interférences entre symboles, et réduire le bruit.

- Il conduit à des calculs très simples et à des structures faciles à mettre en œuvre.
- Il coïncide avec le critère Zero Forcing dans le cas d'un canal non bruité.

En résumé, deux critères sous-optimaux sont classiquement utilisés pour calculer l'égaliseur : le critère naturel de forçage à zéro (Zero Forcing), qui vise à annuler les interférences entre symboles sans tenir compte du bruit, et le critère de l'EQM minimale, qui réalise un compromis entre interférences résiduelles minimales et atténuation du bruit maximale [5].

### 3.4 Egaliseur à retour de décision

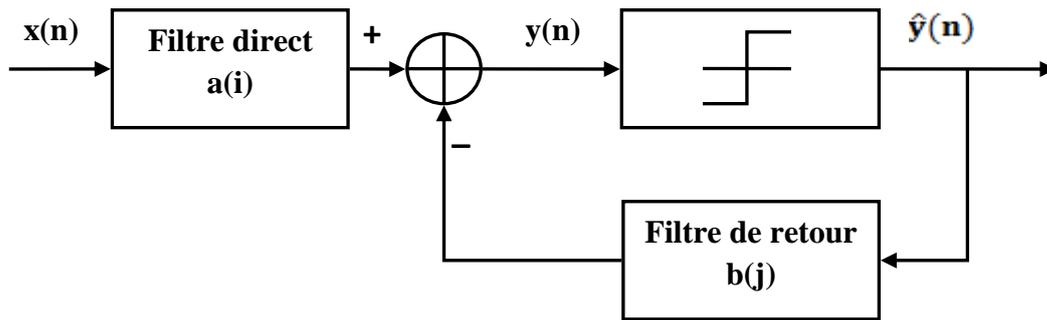
L'égaliseur à retour de décision (DFE), représenté sur la **figure 3.4** est constitué de deux filtres, un filtre direct et un filtre de retour [1][11][12]. L'entrée du filtre direct est la séquence des symboles reçus  $x(k)$ , ainsi, ce filtre est identique à un égaliseur transversal. Le filtre de retour (un second filtre transversal [1]) a comme entrée la séquence des décisions sur les symboles préalablement détectés. L'égaliseur DFE est très utile dans le cas des canaux présentant des distorsions sévères en amplitude.

La sortie de l'égaliseur DFE est donnée par :

$$y(n) = \sum_{i=0}^{N-1} a(i)x(n-i) + \sum_{j=0}^{M-1} b(j)\hat{y}(n-j) \quad (3.16)$$

L'idée de base de l'égalisation à retour de décision est que, si les valeurs des symboles détectées sont connues (c.-à-d. les décisions passées sont supposées correctes), alors l'IES contribué par ces symboles peut être complètement éliminée, en retranchant les valeurs des symboles passés de la sortie de l'égaliseur par les coefficients appropriés du filtre de retour [7]. Notant que l'inclusion de l'organe de décision dans la boucle de retour rend l'égaliseur non linéaire [12].

Lorsqu'une décision incorrecte particulière est faite, la sortie du DFE reflète cette erreur durant quelques symboles prochains pendant que la décision incorrecte traverse le filtre de retour. Ainsi, il y a une plus grande probabilité d'autres fausses décisions suivant la première, à savoir, la propagation d'erreur. Heureusement, la propagation d'erreur dans le DFE n'est pas catastrophique. Sur les canaux typiques, les erreurs se produisent dans des courtes durées qui ne dégradent les performances que légèrement [1].



**Figure 3.4** Egaliseur à retour de décision

Le critère d'erreur quadratique moyenne peut être utilisé pour obtenir une optimisation mathématique docile d'un égaliseur à retour de décision. En effet, l'algorithme LMS<sup>(1)</sup> peut être utilisé pour adapter à la fois les coefficients du filtre direct et celui du retour en se basant sur un signal d'erreur commun.

### 3.5 Egalisation adaptative

Lorsque le canal n'est plus invariant dans le temps, les techniques adaptatives de traitements permettent de concevoir des algorithmes estimant en permanence les meilleurs paramètres pour l'égaliseur. On parle alors d'égalisation adaptative.

Lors de la construction d'un algorithme de filtrage adaptatif des données, il est nécessaire de bien dissocier deux points :

- La structure du filtre utilisé d'une part (structure transverse à réponse impulsionnelle finie, structure réursive avec ou sans décision dans la boucle de retour, ...).
- Le type de l'algorithme utilisé pour adapter les paramètres de cette structure d'autre part.

Nous considérons les deux algorithmes de minimisation les plus courants : l'algorithme du gradient stochastique (*LMS*, pour *Least Mean Square*) (**Annexe 1**), et l'algorithme de Newton stochastique (*RLS*, pour *Recursive Least Square*) (**Annexe 2**). Le premier est connu pour sa simplicité de mise en œuvre, le second pour sa bonne rapidité de convergence initiale.

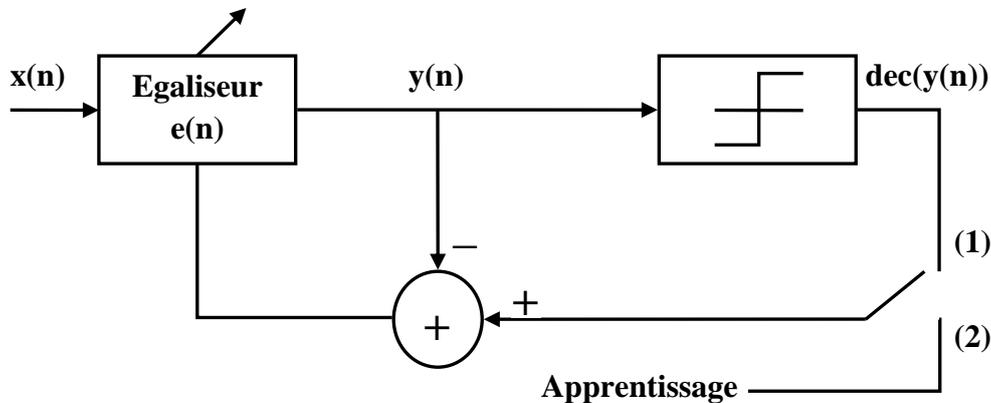
Les algorithmes de minimisation sont utilisés pour adapter au cours du temps les coefficients de l'égaliseur. Les combinaisons structure-algorithme les plus courants sont présentées :

- Filtre transverse (à réponse impulsionnelle finie) couplé à l'algorithme du gradient stochastique (*LMS-FIR*), ou l'algorithme de Newton (*RLS-FIR*).

- Filtre doté d'une partie récurrente avec ou sans décisions dans la boucle de retour, nous considérant le cas le plus utile dans les problèmes de communication : la structure DFE adaptative.

La mise en œuvre de l'algorithme se fait suivant deux modes opératoires, **figure 3.5** :

- Mode supervisé, ou la séquence  $a(n)$  est connue (apprentissage). Le calcul de  $z(n)$  ne sert alors qu'à adapter le filtre, jusqu'à convergence. Au bout de  $K$  itérations, on considère que  $e(K)$  a convergé vers la solution, et
- Mode opérationnel. La sortie de l'égaliseur  $z(n)$  sert alors à estimer  $a(n-R)$  :  $\hat{a}(n-R) = \text{dec}(z(n))$ , où « dec » indique que l'on prend la décision sur  $z(n)$ . L'erreur est alors maintenant calculée à partir des décisions :  $\varepsilon(n) = \text{dec}(z(n)) - z(n)$ . En phase opérationnelle, l'algorithme est dit « piloté par les décisions » (*decision directed*).



**Figure 3.5** Egaliseur adaptatif : (1) piloté par les décisions, (2) apprentissage

### 3.6 Egalisation à maximum de vraisemblance et algorithme de Viterbi

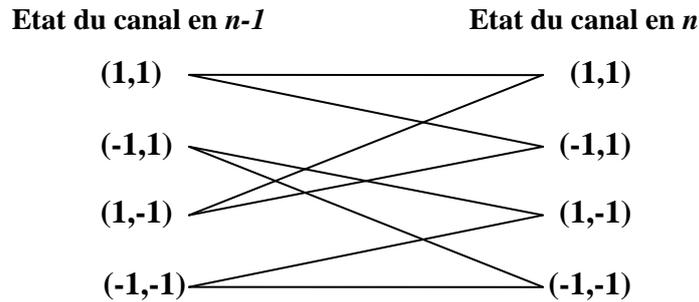
La présence d'interférences entre symboles, liée à un canal imparfait, est caractéristique d'une *mémoire* dans le signal. On peut alors chercher à reconstituer la *séquence de symboles* au sens du maximum de vraisemblance, en exploitant l'interdépendance des données reçues, et en maximisant la vraisemblance à l'aide de l'algorithme de Viterbi. Rappelons que celui-ci permet de sélectionner dans un treillis le chemin de métrique la plus faible, où la métrique est une mesure cumulée, mise à jour de nœud en nœud. Ceci signifie en fait que la densité de probabilité conjointe des observations pour une séquence particulière (la vraisemblance), peut s'écrire comme le produit des vraisemblances élémentaires à chaque instant. Il est important de noter alors que ceci n'est vrai que si *le bruit d'observation, après échantillonnage, est blanc*. Ce modèle garantissant l'optimalité des procédés de réception, et en particulier la mise en œuvre de l'algorithme de Viterbi.

### 3.6.1 Représentation du canal par un graphe de treillis

La durée de la réponse impulsionnelle du canal étant supposée finie (filtre RIF) ; sa sortie, échantillonnée de façon synchrone, s'écrit :

$$y(k) = \sum_{i=0}^L h(k-i)a(k-i) + b(k) \quad (3.17)$$

Cette formule traduit le fait que le canal possède une mémoire dont le contenu caractérise son état à un instant donné ; on note cet état  $e(k) = a(k-1), \dots, a(k-L)$  l'arrivée d'un nouveau symbole  $a(k+1)$  occasionne un changement de l'état vers  $e(k) = a(k), \dots, a(k-L+1)$ . L'évolution de l'état du canal peut être représentée par un treillis dont chaque nœud correspond à un état possible et chaque branche à une transition possible. La **figure 3.6** montre le cas d'un canal à deux coefficients ( $L=1$ ) dont l'entrée se compose de  $M=2$  niveaux ( $\pm 1$ ). A chaque instant, il y a  $M^{L+1} = 4$  état possible. Pour  $k$  variant de  $0$  à  $N-1$  (séquence de longueur  $N$ ), il existe  $M^N$  chemins possible sur les branches du treillis. Déterminer la séquence la plus probablement émise revient à déterminer le chemin le plus probable sur cette représentation.



**Figure 3.6** Graphe en treillis pour une modulation à deux états et un canal de mémoire égale à deux échantillons

Si on considère maintenant une séquence de symboles de longueur  $K$ , et que l'on note  $s^{(n)}$  la  $n^e$  parmi les  $M^K$  séquences possibles, alors, sous l'hypothèse d'un bruit additif blanc gaussien, la densité de probabilité conjointe de la séquence d'observation  $y_1, y_2, \dots, y_K$  si la séquence  $s(n)$  a été émise, est :

$$p(y_1, y_2, \dots, y_K / s^{(n)}) = \prod_{k=1}^K \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp \left\{ -\frac{y_k - s_k^{(n)2}}{2\sigma_b^2} \right\} \quad (3.18)$$

$$= \left( \frac{1}{\sqrt{2\pi\sigma_b^2}} \right)^2 \exp \left\{ \frac{-1}{2\sigma_b^2} \sum_{k=1}^K y_k - s_k^{(n)} \right\}^2 \quad (3.19)$$

où  $s^{(n)}(k) = \sum_{i=0}^{L-1} v(i)s^{(n)}(k-i)$  est la sortie à l'instant  $k$  qui correspondrait à l'entrée  $s(n)$ . on peut alors rechercher la séquence  $s(n)$  qui maximise la vraisemblance, ou qui minimise la métrique cumulée :

$$D(k) = \sum_{k=1}^K y_k - s_k^{(n)} \quad (3.20)$$

À tout instant  $i$ , on observe que  $D(i) = D(i-1) + (y_i - s_i^{(n)})^2$ , ce qui signifie que l'on peut mettre en œuvre l'algorithme itérativement. Ceci étant posé, l'algorithme de Viterbi procède exactement comme nous l'avons indiqué dans le chapitre sur le récepteur optimal : parmi tous les chemins entrants sur un nœud, on ne conserve que celui de plus faible métrique cumulée, et seul ce chemin est prolongé. La complexité calculatoire passe alors de  $M^K$  à  $KM^L$  chemins à évaluer. On répète ces opérations au cours du temps, et on peut prendre les décisions avec un retard de  $5L$ .

Cet égaliseur est l'égaliseur « extrême ». En effet, il affiche les meilleures performances, et la plus grande complexité en termes d'implantation et charge calculatoire : il n'est guère utilisable que pour des séquences binaires ( $M = 2$ ) et des canaux courts ( $L < 10$ ). En outre, on ne peut utiliser cet algorithme que si on connaît la réponse impulsionnelle du canal, ce qui nécessite donc d'identifier le canal préalablement à la transmission.

### 3.7 Résultats de simulation

Afin de valider les modèles théoriques décrits dans ce chapitre, nous allons examiner leurs performances pour une transmission numérique en modulation de phase à deux états (BPSK) sur le canal à phase non minimale décrit ultérieurement. Les simulations sont réalisées au moyen du logiciel MATLAB.

#### 3.7.1 Canal à phase non minimale utilisé dans la simulation

Le modèle du canal discret équivalent utilisé pour tester les performances des égaliseurs est un canal à phase non minimale ; il est représenté par la fonction de transfert en «  $z$  » définie par :

$$H(z) = 0.3482 + 0.8704 z^{-1} + 0.3482 z^{-2} \quad (3.21)$$

C'est un canal représentatif de la majorité des canaux rencontrés en pratique, il est très utilisé pour la simulation des égaliseurs [13][14][15][16].

Les caractéristiques de ce canal sont illustrées par la **figure 3.7**. La **figure 3.7.a** représente Les coefficients de la réponse impulsionnelle du canal qui sont au nombre

de trois. La **figure 3.7.b** représente la réponse en amplitude du canal qui comporte des zones où le signal sera amplifié (amplitude  $> 0$ ) et des zones où le signal sera très affaibli (amplitude  $< 0$ ). La **figure 3.7.c** représente la réponse en phase qui est linéaire. La **figure 3.7.d** donne la représentation dans le plan Z et montre que le canal possède deux zéros : l'un est à l'intérieur du cercle unité et l'autre est à l'extérieur de ce cercle.

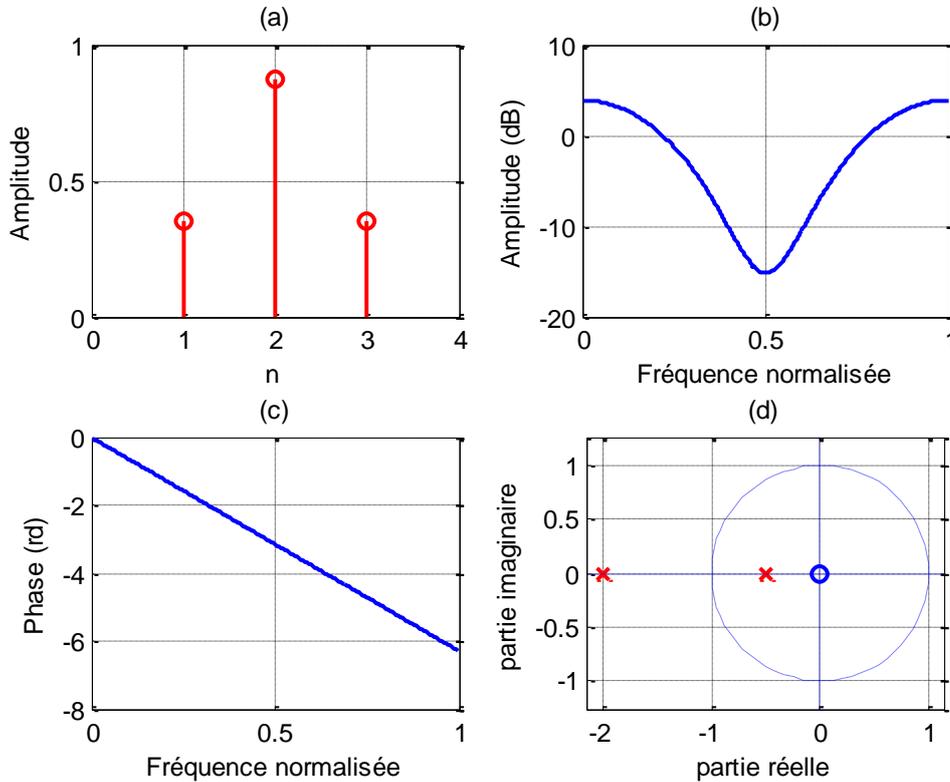


Figure 3.7 Caractéristiques du canal de simulation

### 3.7.2 Simulation des égaliseurs

La **figure 3.8** présente une comparaison des courbes de convergence de l'égaliseur linéaire transverse (*LTE*) pour l'algorithme LMS (*Least Mean Square*) avec un pas d'adaptation  $\mu = 0.035$  [9], et l'algorithme RLS (*Recursive Least Square*) avec un facteur d'oubli  $\lambda = 1$ . L'égaliseur LTE est de l'ordre de sept coefficients ( $N=7$ ). Ces courbes décrivent l'évolution de l'erreur quadratique moyenne (*Mean Square Error*) en fonction du nombre de durée symbole pour un rapport signal sur bruit (*SNR*) de 20dB.

Comme le montre la **figure 3.8**, l'EQM atteint sa valeur optimale après environ 500 échantillons pour le LTE-LMS, et 100 échantillons pour le LTE-RLS, cette valeur optimale est autour de -10 dB pour les deux égaliseurs. L'égaliseur LTE-RLS converge plus rapidement que le LTE-LMS.

La **figure 3.9** présente une comparaison des courbes de convergence de l'erreur quadratique moyenne (MSE) des égaliseurs DFE et LTE dont l'adaptation est faite à l'aide de l'algorithme du gradient stochastique (LMS) pour un rapport signal sur bruit de 20 dB. L'égaliseur DFE contient 7 coefficients dans sa section directe et 4 coefficients dans sa section récurrente (7,4). L'égaliseur LTE comporte 11 coefficients. Le pas d'adaptation  $\mu = 0.035$ .

L'égaliseur LTE converge vers un état stable de -10 dB après 500 échantillons, alors que l'égaliseur DFE atteint ce même niveau après environ 100 échantillons seulement. L'état stable du DFE est de -14 dB obtenu après 280 échantillons.

Les **figures 3.10** et **3.11** représentent le diagramme de l'œil du signal BPSK à l'entrée et à la sortie du canal respectivement, la **figure 3.11** montre que l'œil est fermé et les erreurs de décisions sont fortement probables.

Les **figure 3.12** et **3.13** montrent le diagramme de l'œil du signal à la sortie des égaliseurs LMS et RLS, l'œil est devenu légèrement ouvert, ce diagramme indique que des erreurs de décision peuvent survenir, et que ces deux égaliseurs sont sous optimales et donnent des résultats qui ne sont satisfaisants.

La **figure 3.14** représente le diagramme de l'œil du signal à la sortie de l'égaliseur DFE, cette figure montre une amélioration du diagramme de l'œil par rapport à celui de l'égaliseur LTE (LMS ou bien RLS), et l'œil devient plus ouvert.

La **figure 3.15** représente le diagramme de l'œil du signal égalisé par l'algorithme de Viterbi, elle montre que celui-ci représente le cas extrême, le signal est parfaitement égalisé, le diagramme de l'œil paraît semblable à celui du signal BPSK avant le canal.

La **figure 3.16** illustre les courbes du BER (Bit Error Rate) des différents types d'égaliseurs, ces courbes montrent que le MLSE est la meilleure solution d'égalisation, le DFE vient en deuxième rang, et enfin l'égaliseur LTE avec les deux algorithmes d'adaptation LMS et RLS qui sont proches en performances tel qu'il montre le BER. Les courbes du BER sont obtenues au moyen d'une simulation Monte-Carlo.

A l'exception du détecteur MLSE, qui est souvent inutilisé à cause de sa complexité qui augmente avec la longueur du canal, les égaliseurs LTE et DFE adaptatifs restent sous optimales, et présentent des résultats relativement médiocres, leurs courbes de convergences présentent un niveau MSE supérieur à celui du bruit.

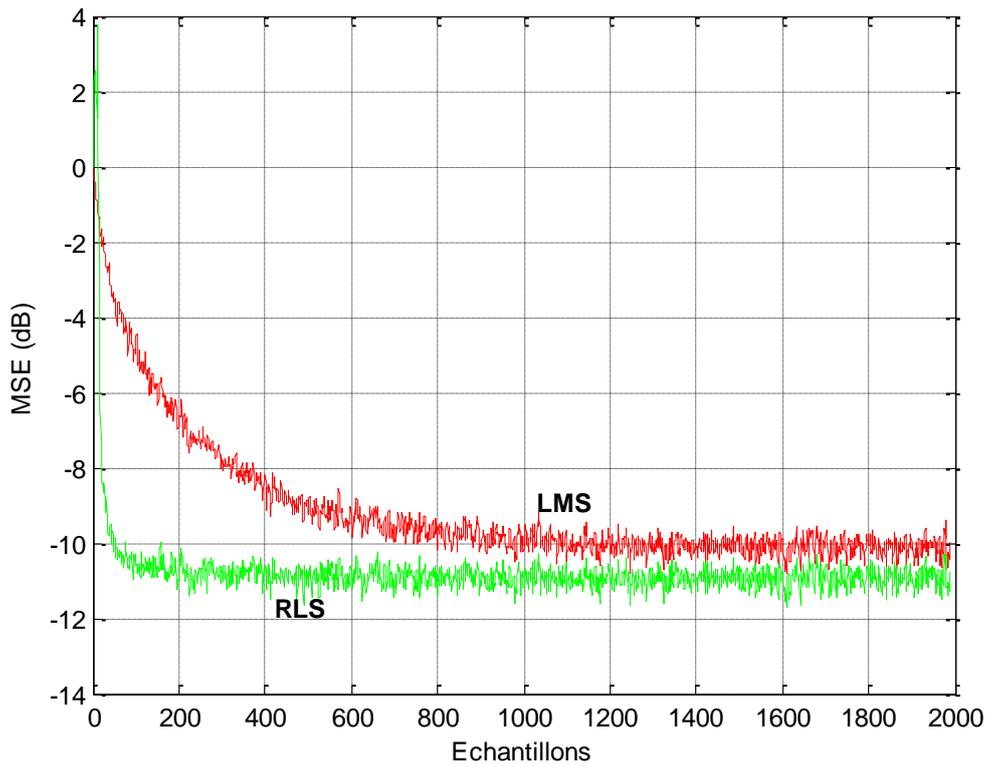


Figure 3.8 Courbes de convergence de l'EQM de l'égaliseur LTE

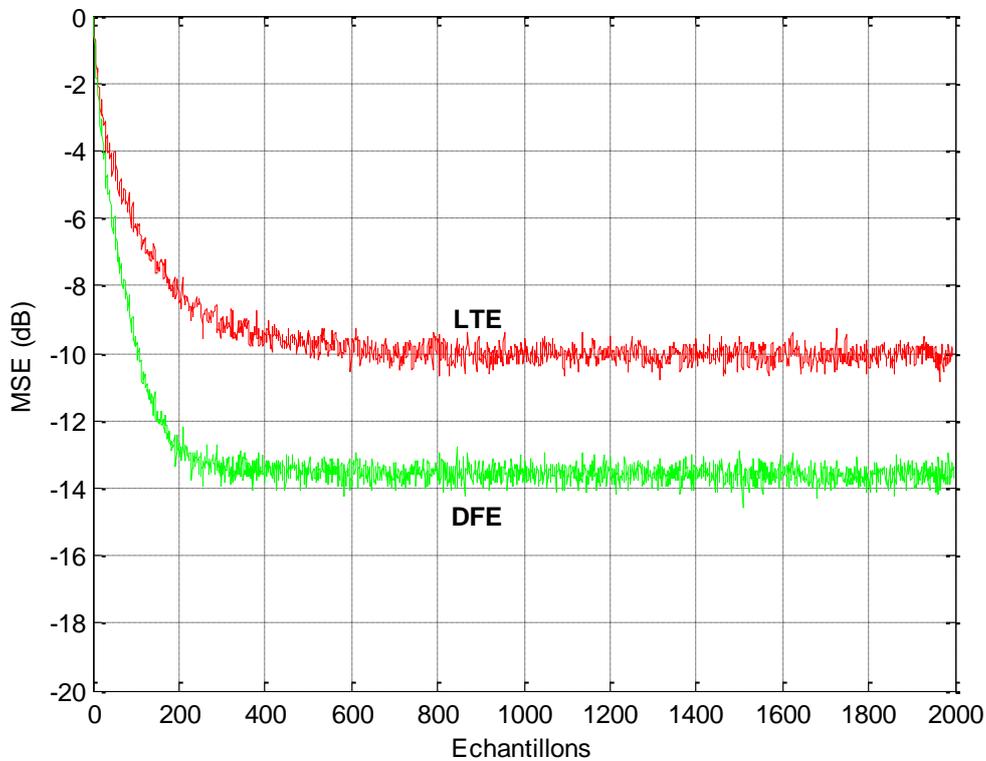
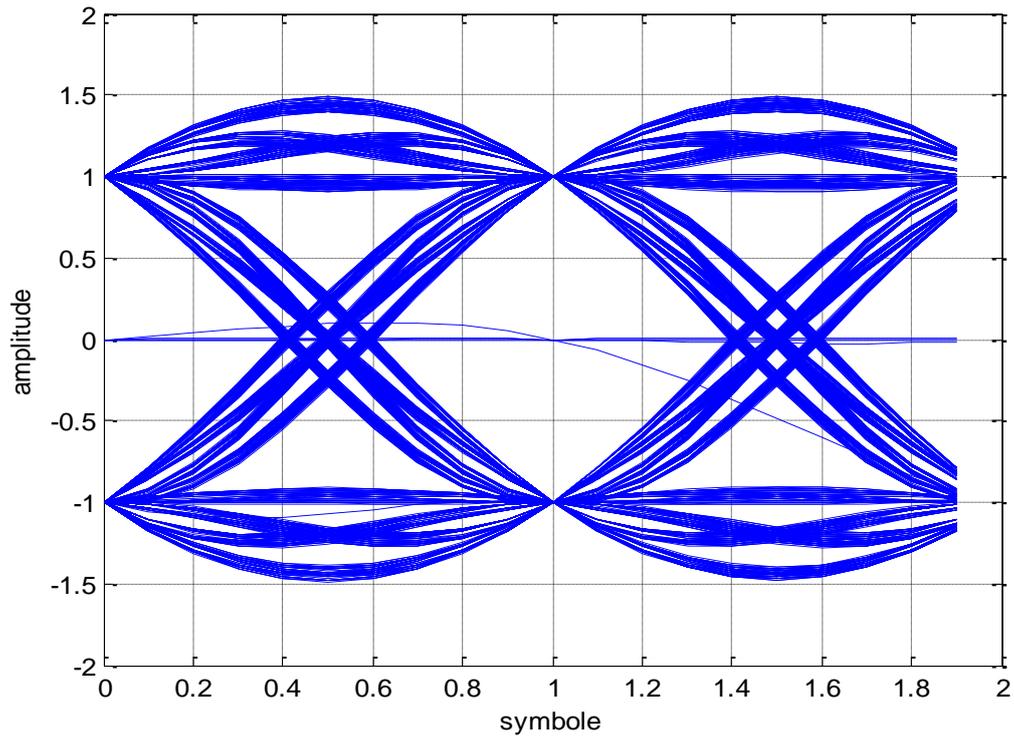
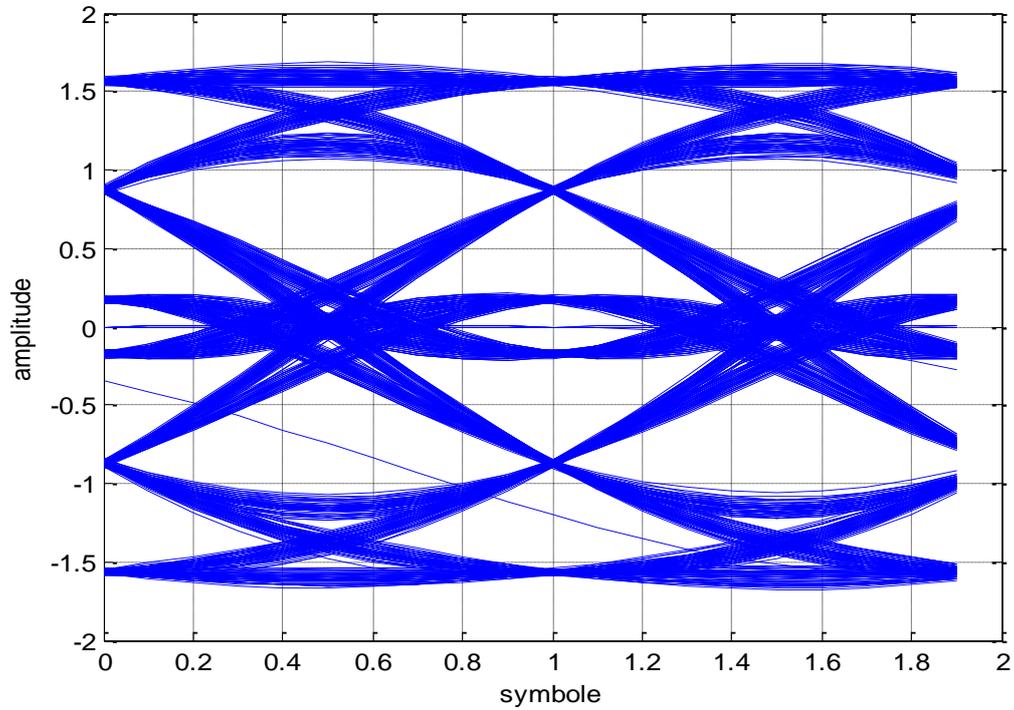


Figure 3.9 Courbes de convergence de l'EQM des égaliseurs LTE et DFE (LMS)



**Figure 3.10** Diagramme de l'œil du signal BPSK transmis



**Figure 3.11** Diagramme de l'œil du signal BPSK en sortie du canal AWGN

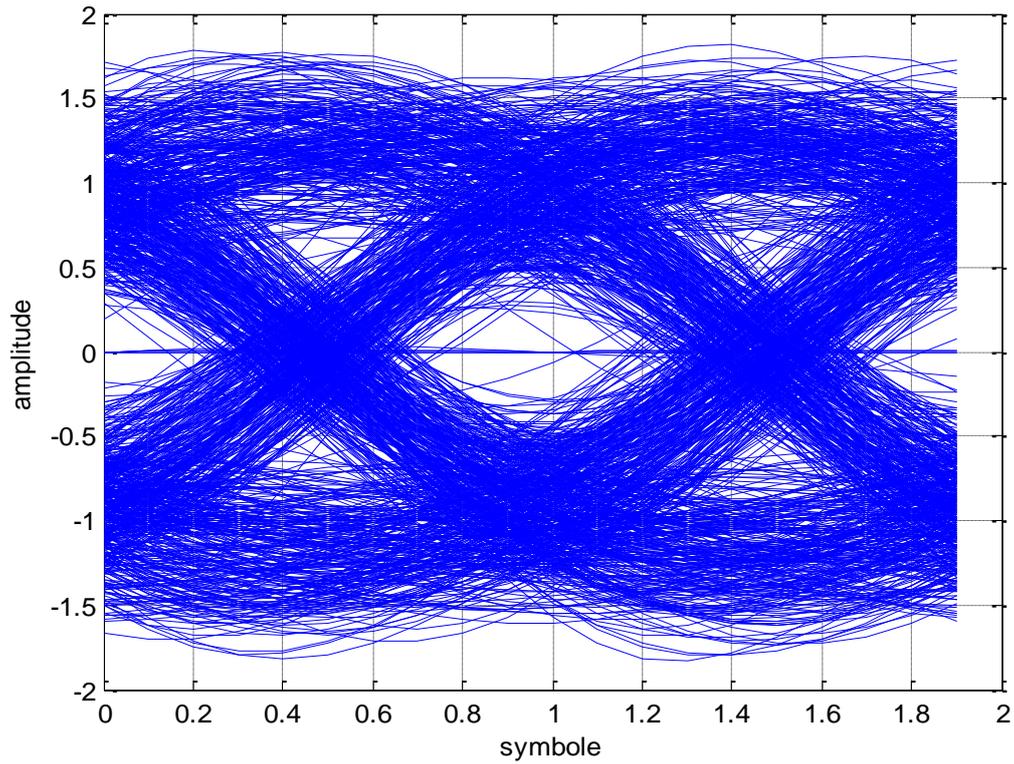


Figure 3.12 Diagramme de l'œil du signal en sortie de l'égaliseur LTE-LMS

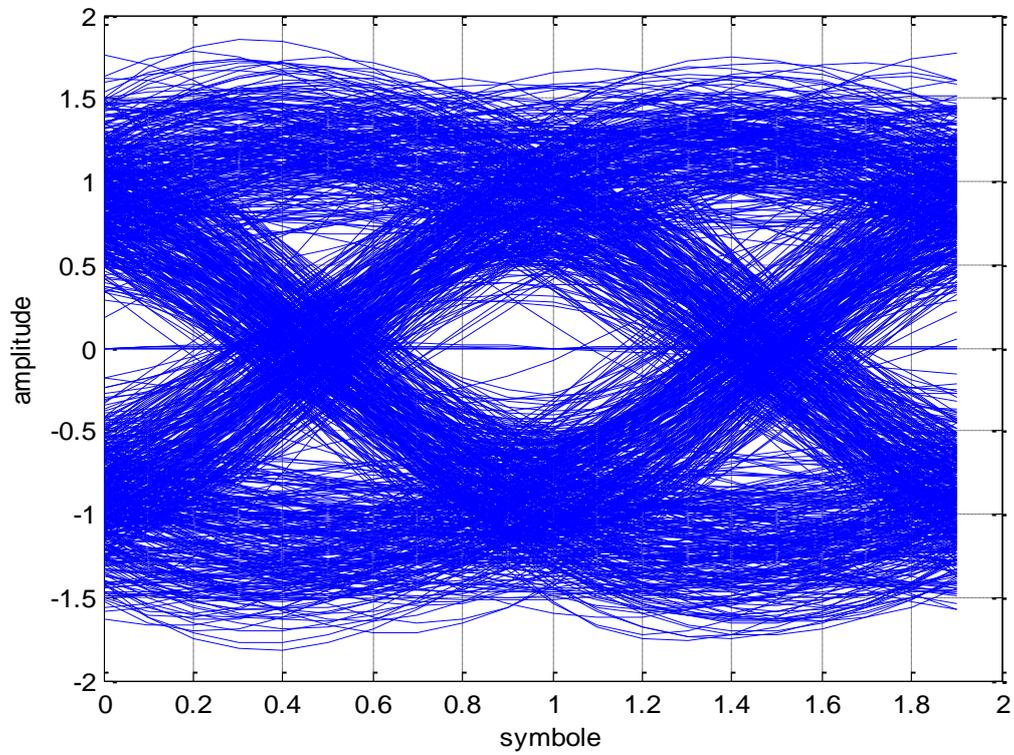


Figure 3.13 Diagramme de l'œil du signal en sortie de l'égaliseur LTE-RLS

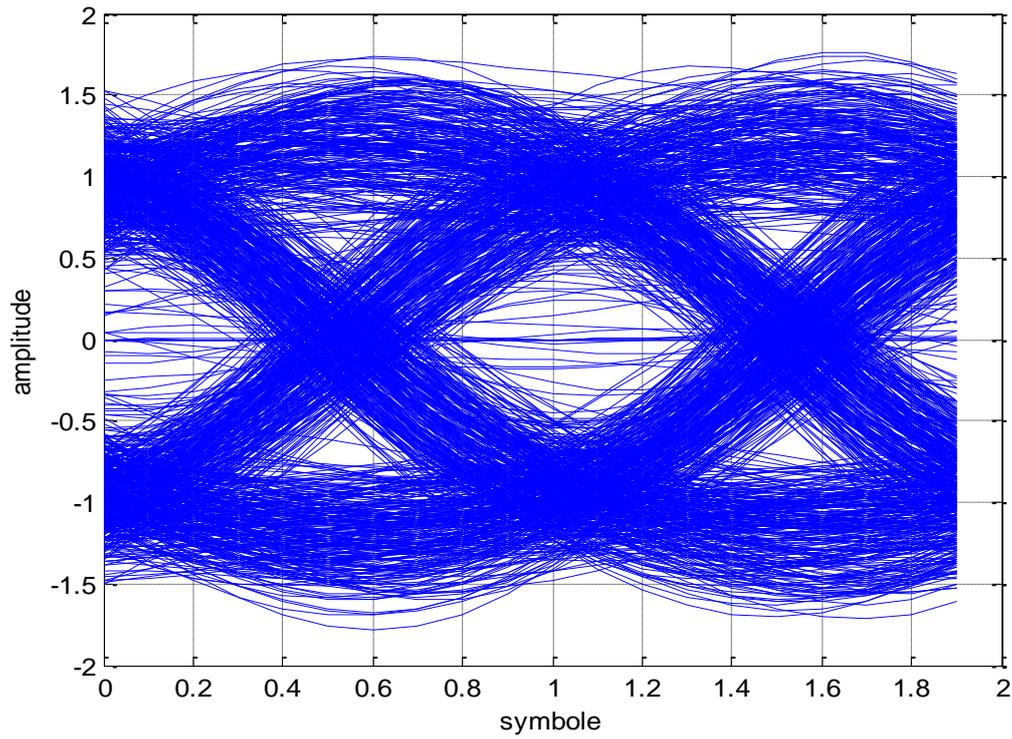


Figure 3.14 Diagramme de l'œil du signal en sortie de l'égaliseur DFE-LMS

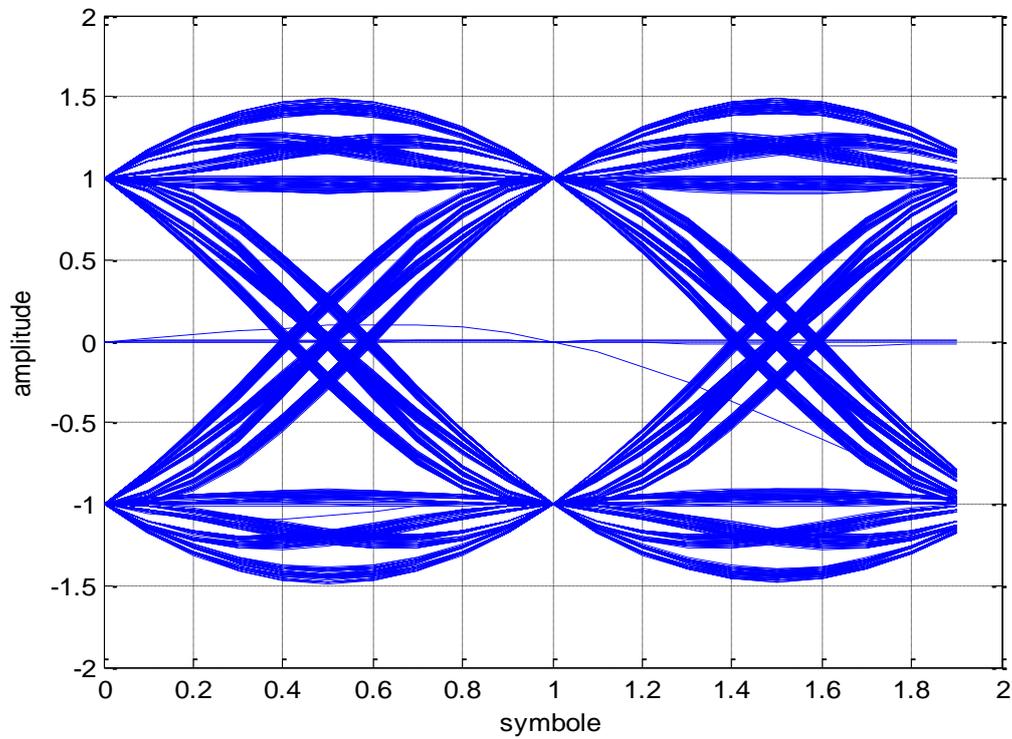


Figure 3.15 Diagramme de l'œil du signal MLSE

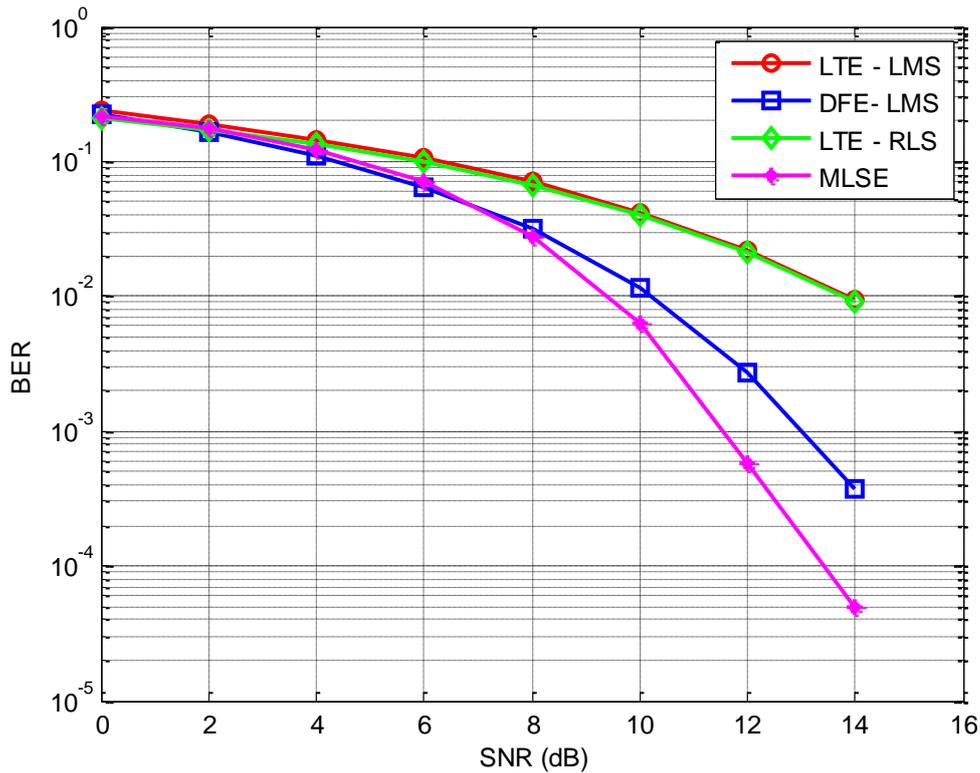


Figure 3.16 Courbes BER des différents égaliseurs

### 3.8 Conclusion

Dans ce chapitre nous avons discuté le contexte d'égalisation des canaux de communication numériques et nous avons également mis en évidence certaines structures d'égaliseurs les plus communes ; l'égaliseur LTE et l'égaliseur DFE adaptés à l'aide des algorithmes LMS et RLS, ainsi que l'égalisation au sens du maximum de vraisemblance MLSE, accomplie par l'algorithme de **Viterbi**. Les méthodes classiques ont été présentées dans le but de mettre en évidence leurs performances médiocres pour un canal à phase non minimale ; les niveaux MSE atteints sont sous optimales. Les égaliseurs conventionnels présentent une incapacité à égaliser les canaux linéaires à phase non minimale ; d'où la nécessité des architectures présentant un traitement non linéaire pour améliorer les performances. Ces architectures feront l'objet du chapitre suivant.

# Chapitre 4

## Egalisation à base des réseaux de neurones

---

L'égalisation non linéaire à base des réseaux de neurones offre une solution plus avancée au problème d'égalisation avec des améliorations dans les performances et dans les types des canaux qui peuvent être égalisés par rapport aux techniques conventionnelles qui aboutissent à des performances sous optimales. Les architectures des égaliseurs LTE et DFE peuvent profiter de la mise en œuvre de ces structures en montrant une amélioration des performances. Trois types particuliers des réseaux de neurones sont investis dans l'égalisation des canaux de communication : le perceptron multicouches MLP (*Multi Layer Perceptron*), la fonction de base radiale RBF (*Radial Basis Function*) et le réseau récurrent RNN (*Recurrent Neural Network*). Nous focaliserons notre étude principalement sur le perceptron multi couches MLP. Nous commençons d'abord par donner un aperçu général sur les réseaux de neurones.

## 4.1 Historique

L'idée d'élaborer un modèle mathématique du cerveau humain, en quête de reproduire ses aptitudes intellectuelles, est à l'origine de la création des réseaux de neurones artificiels. Les premiers travaux sur ces systèmes ont été menés par les neurologues **Warren McCulloch** et **Walter Pitts** en **1943**. Ces travaux ont été suivis en **1949** par le travail de **Donald Hebb** qui a proposé une simple règle d'apprentissage dans son ouvrage « The organization of behaviour ». Le premier processus artificiel capable d'apprendre par expérience a été par **Franck Rosenblatt** en **1957**, c'était le perceptron. Mais, **Marvin Lee Minsky** et **Seymour Papert** ont publié un travail énonçant les limitations de ce dernier, et principalement l'impossibilité de traiter des problèmes non linéaires. Cette conclusion pessimiste a étendu tous les modèles des réseaux de neurones et a abaissé leurs intérêts pendant deux décennies. Il a fallu attendre l'apparition du perceptron multicouche, introduit par **Rumelhart** en **1986**, pour donner renaissance aux réseaux de neurones. Cette découverte a été une vraie révolution qui a permis aux réseaux de neurones de connaître un essor considérable. Actuellement, la littérature sur les réseaux de neurones est devenue énorme et ne cesse de croître [18].

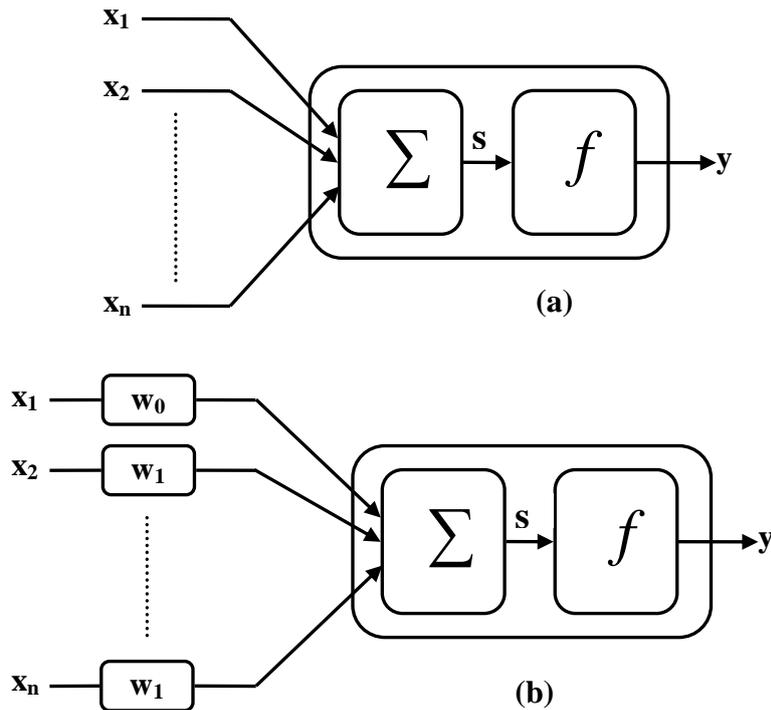
Malgré la réussite récente des réseaux de neurones artificiels, la communauté scientifique est toujours loin de la mise en œuvre de machine capable de reproduire les capacités de calcul des systèmes nerveux, même les plus simples. Il s'agit en fait d'une simple modélisation du comportement des neurones biologiques qui ne pourra, en aucune manière, de représenter leurs complexités de fonctionnement. En revanche, cette modélisation permet de mettre en œuvre des processus ayant de grandes capacités de calcul en se basant sur des simples cellules de base : les neurones formels.

## 4.2 Le neurone formel

La première modélisation du neurone remonte aux années **40** où **McCulloch** et **Pitts** ont proposé le premier modèle de neurones formels. C'est un modèle mathématique qui réalise une somme pondérée des signaux qui lui parviennent et déclenche une réponse si cette somme dépasse un certain seuil. La **figure 4.1.a** résume la chaîne de traitement développée par ce neurone. Ce modèle n'a pas possédé une règle d'apprentissage jusqu'à **1949** où **Hebb** a proposé un principe d'apprentissage sans avoir donné d'équations, **figure 4.1.b**.

Où les  $x_i$  sont les variables (ou entrées) du neurone, les  $w_i$  sont les paramètres ajustables (ou les poids),  $y$  est la sortie du neurone, et  $f$ , la fonction d'activation.

Un neurone donc est, avant tout, un opérateur mathématique qui reste loin de fournir une explication exacte sur la complexité de fonctionnement des neurones biologique, malgré cela, cette formalisation permet d'étudier les connexions entre ces neurones dans d'autres processus plus complexes comportant plusieurs neurones interconnectés.



**Figure 4.1.** Neurone formel : (a) Modèle de McCulloch et Pitts, (b) Modèle avec capacité d'apprentissage.

#### 4.2.1 Différents types de neurones

La *fonction d'activation* (de transfert) utilisée dans le modèle de **McCulloch & Pitts** est la fonction *échelon* (**figure.4.2.a**). Elle fait passer l'activation du neurone d'une valeur à une autre dès que l'entrée résultante dépasse un certain seuil. L'inconvénient de cette fonction est qu'elle n'est pas différentiable, ce qui pose un problème pour les algorithmes d'apprentissage basés sur le gradient.

Pour remédier à cet inconvénient, on cherche à approximer cette fonction d'activation par une fonction « non linéaire » différentiable. Deux fonctions de ce type sont particulièrement intéressantes et sont souvent utilisées : la fonction tangente hyperbolique (**figure.4.2.c**) définie par :

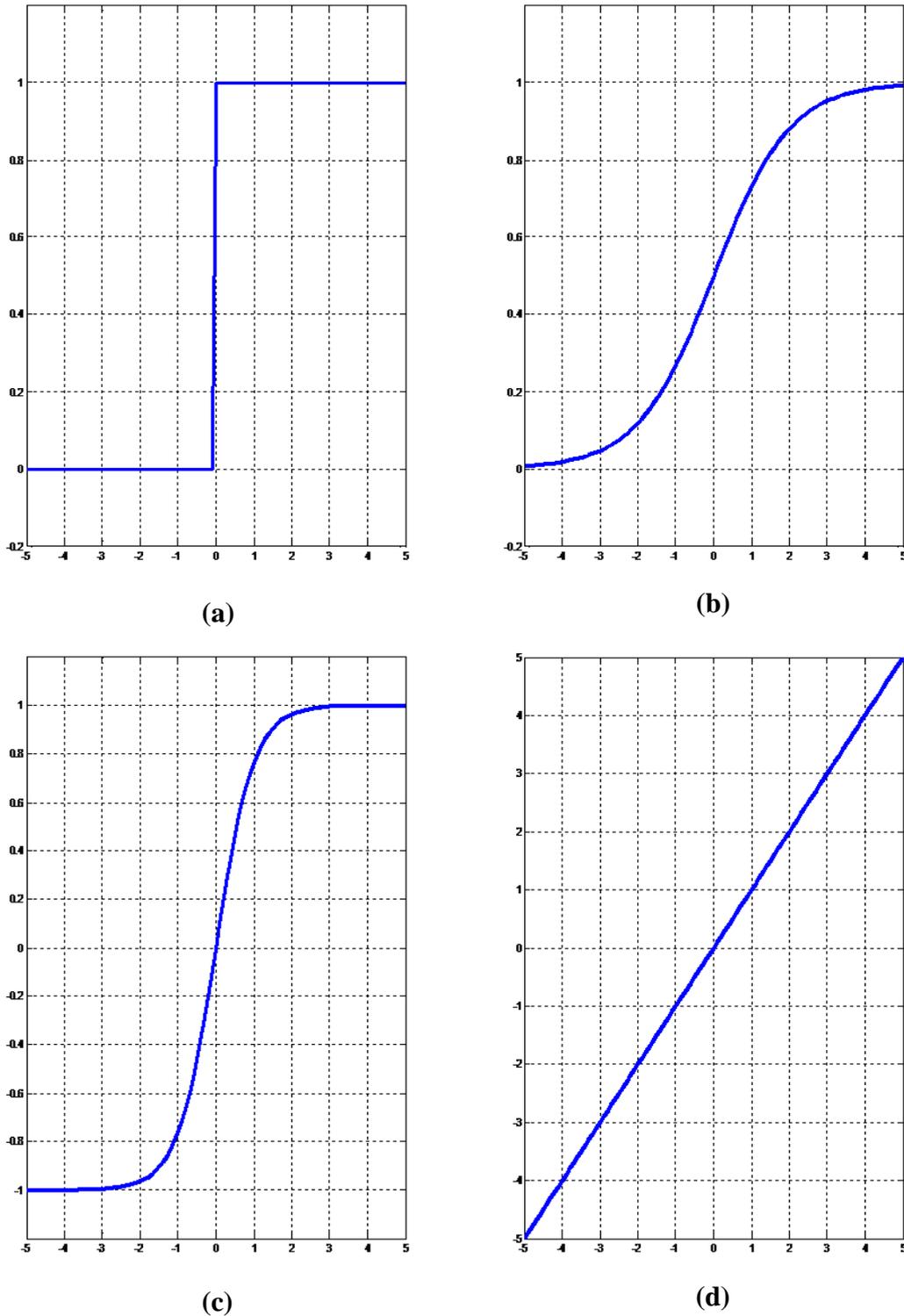
$$f_{\tanh}(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}} \quad (4.1)$$

Et la fonction logistique figure (4.2.b) dont l'expression est la suivante :

$$f_{\text{sig}}(s) = \frac{1}{1 + e^{-s}} \quad (4.2)$$

Les fonctions sigmoïdales ont la propriété d'être différentiables, ce qui est nécessaire pour les algorithmes basés sur le gradient. Une autre propriété intéressante

est le fait que les fonctions dérivées peuvent s'exprimer facilement à l'aide des fonctions elles mêmes, ce qui permet un gain significatif de temps de calcul [17].



**Figure 4.2.** Différentes fonctions d'activation

(a) Fonction échelon

(b) Fonction sigmoïde unipolaire

(c) Fonction tangente hyperbolique

(d) Fonction rampe

## 4.3 Les réseaux de neurones formels

### 4.3.1 Définition

Il n'y a pas de définition universellement tenue pour les réseaux de neurones. Cependant, nous considérons quelques définitions qui sont suffisantes et complémentaires. Selon **Haykin**, un réseau de neurones est un processeur massivement distribué en parallèle qui a une disposition naturelle pour stocker de la connaissance empirique et la rendre disponible à l'usage. Il ressemble au cerveau sur deux aspects. D'abord, la connaissance est acquise par le réseau au travers d'un processus d'apprentissage, et deuxièmement; les connexions entre les neurones, connues sous le nom de poids synaptique, servent à stocker la connaissance. Selon **Zurada**, les systèmes de neurones artificiels, ou réseaux de neurones, sont des systèmes physiques cellulaires qui peuvent acquérir, stocker, et utiliser de la connaissance empirique. Selon **Nigrin**, un réseau de neurones est un circuit composé d'un nombre très important d'unités de calcul simples basées sur des neurones. Chaque élément opère seulement sur l'information locale. Chaque élément opère de façon asynchrone; il n'y a donc pas d'horloge générale pour le système [18].

### 4.3.2 Domaines d'application des réseaux de neurones

L'essor des réseaux de neurones dans divers domaines actuels est certainement dû à leurs grandes capacités de calcul et à leurs hautes habilités d'apprentissage. De plus, l'estimation de leurs paramètres est indépendante de la complexité du problème traité ce qui leur permet d'être bien adaptés aux problèmes actuels qui ne cessent d'être de plus en plus complexes. Les applications des réseaux de neurones peuvent être récapitulées en trois grands domaines [18]:

- la modélisation : la plus part des problèmes industriels ou de recherche, que ce soit mécanique, physique, chimique ou même économique, nécessitent une représentation à l'aide d'un modèle mathématique permettant de reproduire le comportement du processus mis en œuvre. De telles tâches nécessitent des outils de calcul ayant de grandes capacités de calcul, d'apprentissage et surtout des outils dont leur conception est peu dépendante de la complexité et de la taille du problème traité. Les réseaux de neurones semblent être l'une des solutions les plus adéquates à ce type de problèmes.
- la commande : commander un processus industriel consiste à concevoir un système permettant le calcul de la commande à appliquer à ce processus de manière à lui assurer un comportement dynamique désiré. Les réseaux de neurones permettent de bonnes performances en tant que partie de commande à cause de leur souplesse d'auto adaptation.
- La classification : une autre grande catégorie de problèmes industriels consiste à attribuer, de façon automatique, un objet à une classe parmi d'autres classes possibles. La résolution de ce type de problèmes demande de représenter les exemples à classifier à l'aide d'un ensemble de caractéristiques. Il s'agit ensuite de concevoir un système capable de classifier

ces exemples en se basant sur leur représentation et les réseaux de neurones sont particulièrement bien adaptés à ce type de problème. La classification est d'ailleurs le domaine privilégié des réseaux de neurones.

### 4.3.3 Différents types des réseaux de neurones formels

On distingue deux grands types d'architectures de réseaux de neurones : les réseaux de neurones **non bouclés** (*statiques*), représentés graphiquement par un ensemble de neurones connectés entre eux, l'information circulant des entrées vers les sorties sans retour en arrière, et les réseaux de neurones **bouclés** (*dynamiques* ou *récurrents*).

#### 4.3.3.1 Le perceptron multicouche (Multi Layer Perceptron)

Le perceptron multicouche (**figure 4.3**) est un réseau de neurones non bouclé, très fréquemment utilisé, qui a une structure particulière. Ses neurones sont organisés en couches successives. Chaque neurone d'une couche reçoit des signaux de la couche précédente et transmet le résultat à la suivante, si elle existe. Les neurones d'une même couche ne sont pas interconnectés. L'orientation du réseau est fixée par le sens unique de la propagation de l'information, de la couche d'entrée vers la couche de sortie.

Par convention, les neurones d'entrée ont toujours une fonction d'activation « identité », laissant passer l'information sans la modifier, alors que les neurones de sortie, on peut leur associer une fonction d'activation linéaire ou non, dérivable ou non, suivant la nature du problème à résoudre.

En ce qui concerne la fonction d'activation associée aux neurones cachés, on utilise dans ce mémoire une fonction d'activation sigmoïde.

Les réseaux de neurones non bouclés sont des objets statiques, c.-à-d. si les entrées sont indépendants du temps, les sorties le sont également.

La sortie du  $J^{ème}$  neurone de la  $i^{ème}$  couche ( $S_{i,j}$ ) est donnée par :

$$S_{i,j} = f_{i,j} \left( \sum_{n=1}^{N_{i-1}} w_{n,j} S_{i-1,n} \right) \quad (4.3)$$

Où  $f_{i,j}$  est la fonction d'activation de ce neurone ;  $N_{i-1}$  est le nombre de neurone de la  $(i-1)^{ème}$  couche et  $w_{n,j}$  est le poids reliant le  $n^{ème}$  neurone de la couche précédente à ce neurone.

L'importance du MLP réside essentiellement dans ces capacités de former des frontières de décision complexe. Ces derniers sont déterminés par le nombre de couches cachées et le nombre de neurones de chaque couche.

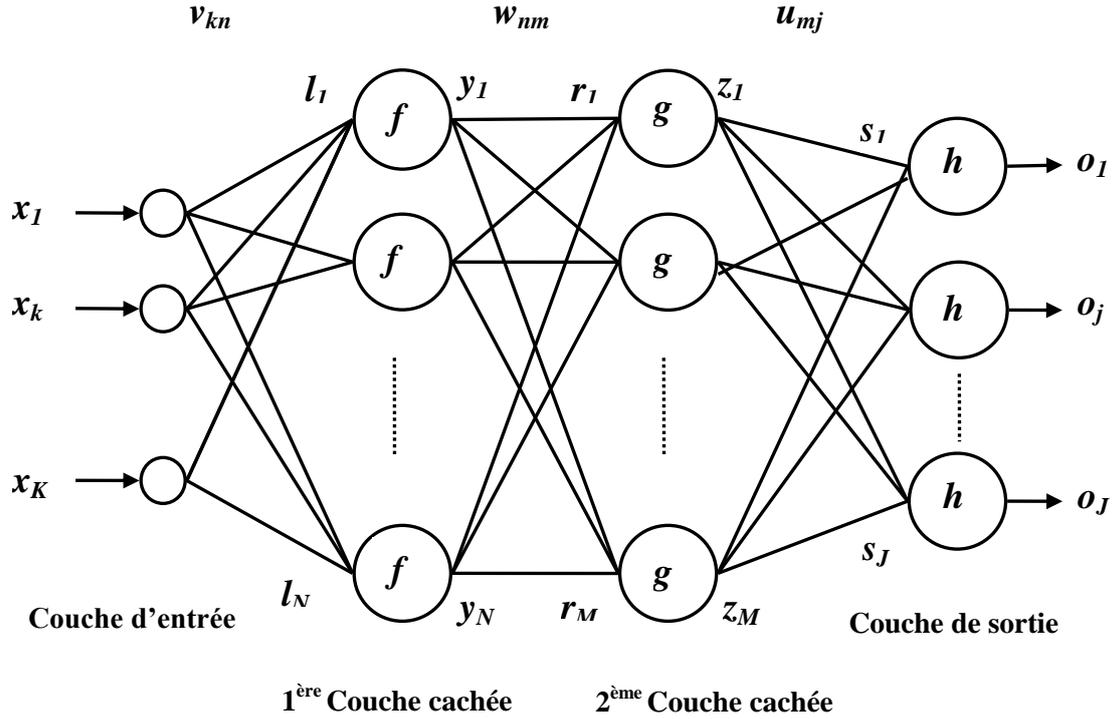


Figure 4.3. Perceptron multicouche

#### 4.3.3.2 Les réseaux de fonctions à base radiale (RBF)

Les réseaux de fonctions à base radiale (*Radial Basis Function*) sont usuellement des réseaux à trois couches (**figure 4.4**) :

- Une couche d'entrée, dont les neurones sont à fonctions d'activation linéaires ;
- Une couche cachée dont les neurones sont à fonctions d'activation gaussiennes ;
- Une couche de sortie dont les neurones sont à fonctions d'activation linéaires.

La sortie du réseau est donnée par :

$$y_k = \sum_{m=1}^M \omega_{mk} \varphi_m, \quad k = 1, \dots, K \quad (4.4)$$

Où  $\varphi_m$  représente la fonction gaussienne suivante (figure 4.5) :

$$\varphi_m = e^{-\frac{\|X - C_m\|^2}{2\sigma_m}} \quad (4.5)$$

Avec :  $X = x_1, x_2, \dots, x_N$ ,  $C_m = c_1^m, c_2^m, \dots, c_N^m$ , et  $m = 1 \dots M$

Et  $\omega_{mk}$  sont les poids entre la couche de sortie et la couche cachée.

Avec  $\|\cdot\|$  est la distance euclidienne : 
$$\|X - C\| = \sqrt{\sum_{n=1}^N (x_n - c_n)^2} \quad (4.6)$$

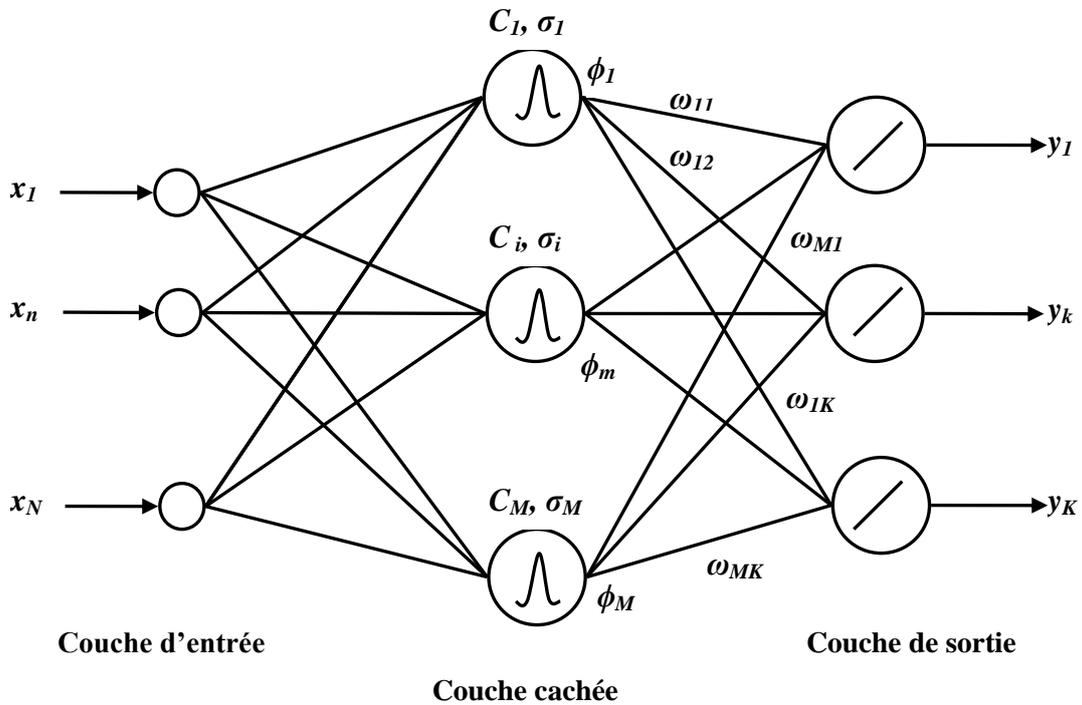


Figure 4.4 Réseau RBF

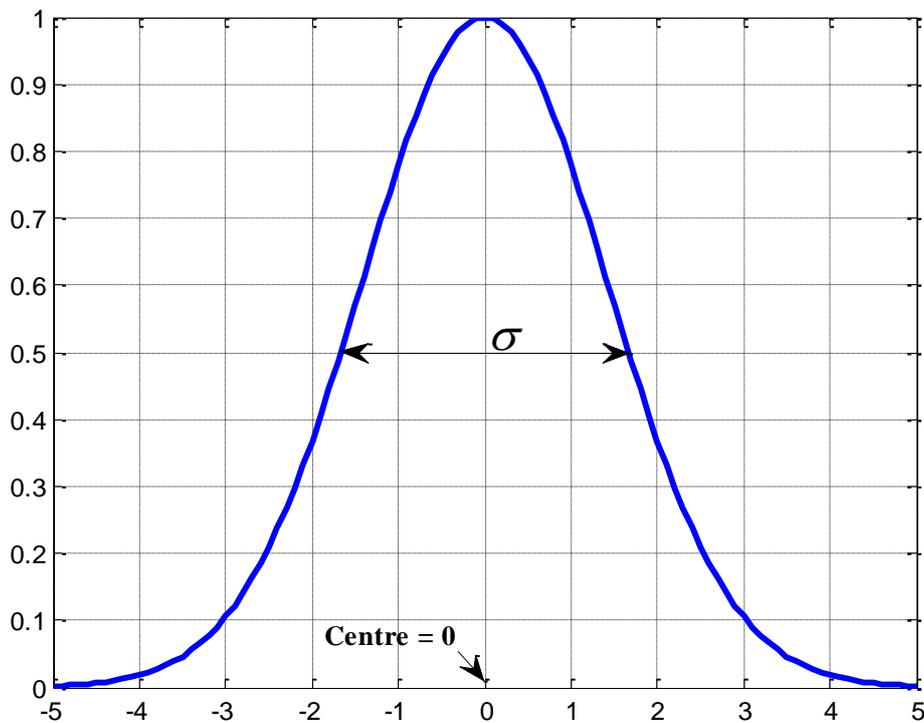


Figure 4.5 la fonction à base radiale (RBF)

Dans cette écriture,  $C_m$  sont les centres des fonctions gaussiennes et  $\sigma_m$  caractérise la dispersion. Ce type de réseau est principalement utilisé dans les problèmes de classifications et de représentation de fonctions.

Les réseaux de fonction à base radiale sont utilisés pour la première fois par **Broomhead** et **Lowe** [18]. Ils ne répondent qu'à une partie de l'espace du signal d'entrée. Une fonction à base radiale est une fonction qui fournit une sortie différente de zéro seulement pour des entrées qui se situent dans une région locale de l'espace des entrées (champ récepteur).

L'apprentissage des réseaux RBF consiste à chercher des valeurs optimales pour les vecteurs,  $c$  et  $\omega$ , en utilisant différentes techniques (**Annexe 4**).

### 4.3.3.3 Les réseaux de neurones bouclés (récurrents)

Contrairement aux réseaux de neurones non bouclés dont le graphe de connexions est acyclique, les réseaux de neurones bouclés peuvent avoir une topologie de connexions quelconque, comprenant notamment des boucles qui ramènent aux entrées la valeur d'une ou plusieurs sorties. Pour qu'un tel système soit causal, il faut évidemment qu'à toute boucle soit associé un *retard* : un réseau de neurones bouclé est donc un système *dynamique*, régi par des équations différentielles ; comme l'immense majorité des applications sont réalisées par des programmes d'ordinateurs, on se place dans le cadre des systèmes à temps discret, où les équations différentielles sont remplacées par des équations aux différences.

Un réseau de neurones bouclé à temps discret est donc régi par une (ou plusieurs) équations aux différences non linéaires, résultant de la composition des fonctions réalisées par chacun des neurones et des retards associés à chacune des connexions.

La forme la plus générale des équations régissant un réseau de neurones bouclé est appelée *forme canonique*, qui est la représentation d'état minimale de la fonction réalisée par ce réseau. Cette forme canonique est constituée d'un graphe acyclique, et de connexions à retard unité reliant certaines sorties de ce graphe à ses entrées. La fonction réalisée par un réseau de neurones ayant cette structure particulière est décrite par les équations aux différences suivantes :

$$x(k+1) = \phi [x(k), u(k)] \quad (4.7)$$

$$y(k) = \psi [x(k), u(k)] \quad (4.8)$$

Où  $x(k)$  est le vecteur d'état à l'instant  $k$ ,  $u(k)$  est le vecteur des variables de commande exogènes,  $y(k)$  le vecteur des sorties,  $\psi$  et  $\phi$  sont deux fonctions qui dépendent de la structure de la partie acyclique du réseau.

Les réseaux de neurones bouclés sont utilisés pour effectuer des tâches de *modélisation de systèmes dynamiques, de commande de processus, ou de filtrage.*

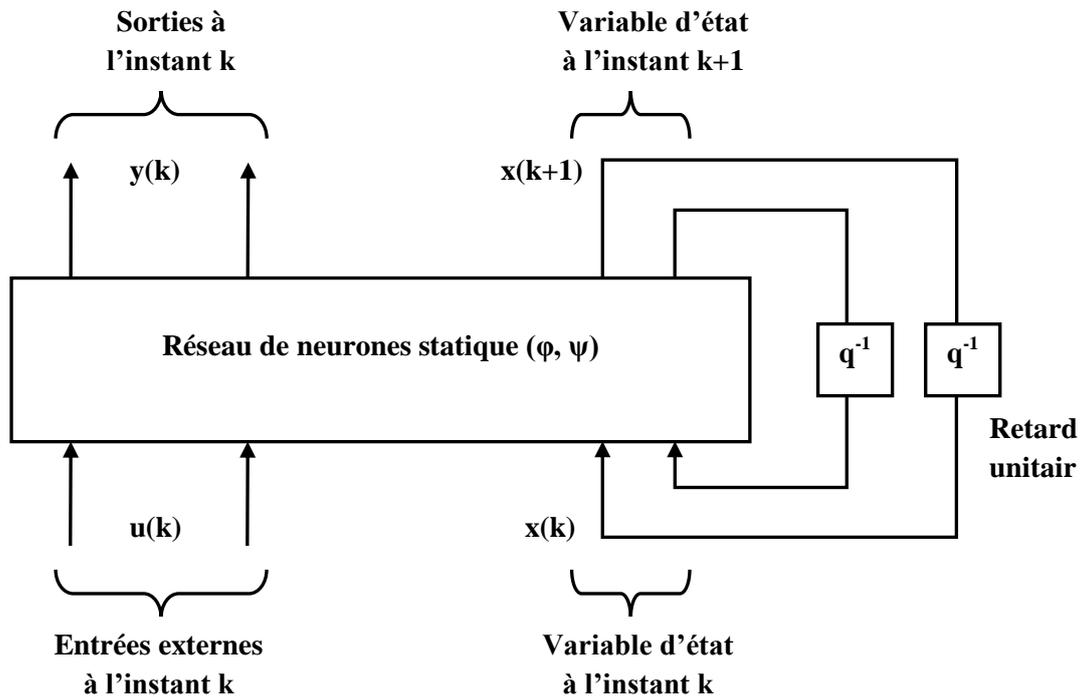


Figure 4.6 Forme canonique d'un réseau de neurone bouclé

#### 4.3.4 Apprentissage des réseaux de neurones

L'apprentissage est une propriété fondamentale pour les réseaux de neurones, c'est le processus permettant au réseau de se spécialiser sur un problème spécifique à partir de son expérience. Il consiste généralement à modifier les poids synaptiques, qui stockent de l'information, jusqu'à ce que le réseau puisse effectuer la tâche désirée. Donc l'apprentissage se traduit par une modification dans la valeur des poids reliant les neurones du réseau. Chaque poids  $w_{i,j}$  reliant un neurone  $i$  au un neurone  $j$  à l'itération  $r$  est modifié selon l'équation suivante :

$$w_{i,j}(r) = w_{i,j}(r-1) + \Delta w_{i,j}(r-1) \quad (4.9)$$

Où  $w_{i,j}(r)$  et  $w_{i,j}(r-1)$  sont respectivement les valeurs de ce poids à la  $r^{\text{ème}}$  et la  $(r-1)^{\text{ème}}$  itération et  $\Delta w_{i,j}(r-1)$  est le changement correspondant.

L'apprentissage des réseaux de neurones peut être catégorisé en deux grandes familles, apprentissage supervisé et non supervisé. Dans la première catégorie, chaque exemple de la base d'apprentissage est associé à une solution désirée. Cette dernière permet au réseau de connaître ces erreurs et de s'adapter à la présentation de chaque exemple d'apprentissage afin de se rapprocher du résultat souhaité. En revanche, dans la deuxième catégorie le réseau ne dispose pas d'une solution désirée sur les exemples d'apprentissage pour l'aider à ajuster ses paramètres, il doit chercher à représenter au

mieux l'espace des exemples qui lui sont présentés. La rétro-propagation du gradient est l'une des méthodes les plus utilisées pour l'apprentissage supervisé des réseaux de neurones.

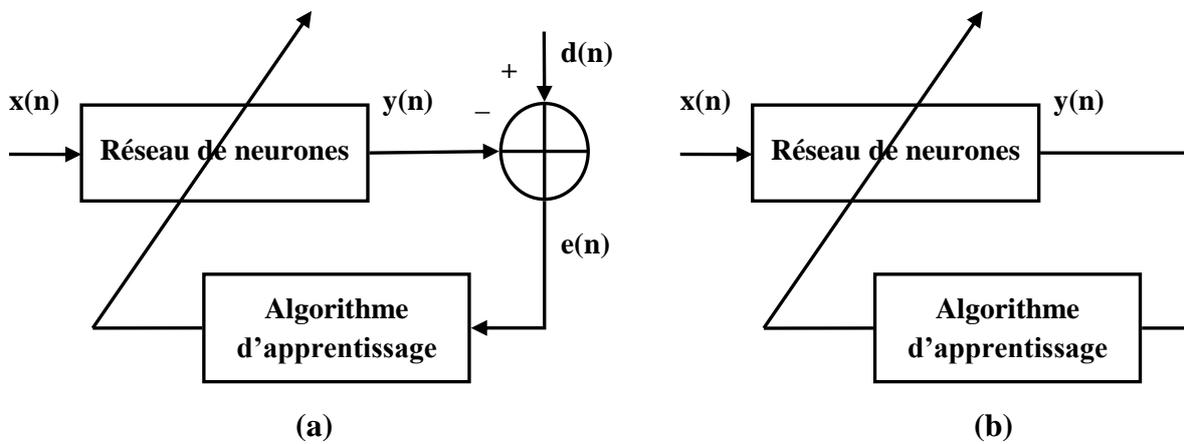


Figure 4.7 (a) Apprentissage supervisé, (b) Apprentissage non supervisé

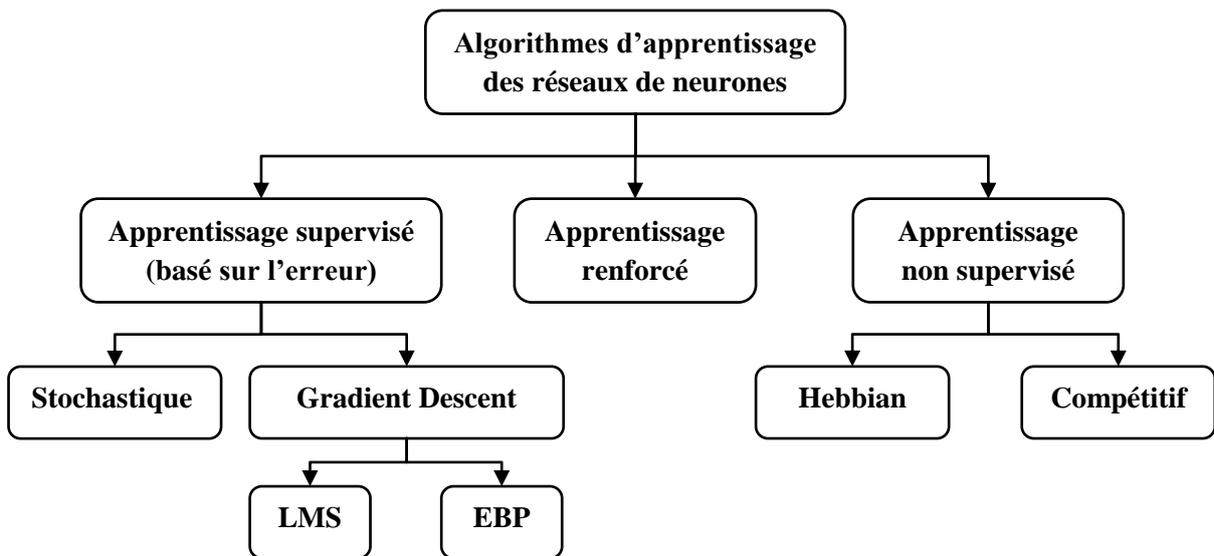


Figure 4.8 Classification des algorithmes d'apprentissage

#### 4.3.4.1 L'algorithme de rétro-propagation (Back Propagation)

La rétro-propagation (RP) du gradient est une extension de la règle de **Widrow** appliquée aux réseaux monocouches. Son développement est lié aux travaux de **Rumelhart** et al [17]. La rétro-propagation consiste à minimiser la distance entre la sortie calculée  $Z^{(q)}$  et la sortie désirée  $T^{(q)}$  correspondante à chaque exemple d'apprentissage  $X^{(q)}$ . L'erreur quadratique est souvent employée comme étant la fonction coût de la RP. La fonction coût usuelle est l'erreur quadratique moyenne à l'exemple  $q$  définie par :

$$E = \frac{1}{2} \sum_{q=1}^Q \sum_{j=1}^J \left( t_j^{(q)} - z_j^{(q)} \right)^2 \quad (4.10)$$

On considère la notation du réseau de neurones à deux couches cachées de la **figure 4.3**.

L'algorithme de la RP modifie les poids du réseau de façon à minimiser l'erreur. Au début de l'apprentissage, les poids sont initialisés aléatoirement et modifiés ensuite dans une direction qui réduira l'erreur. La modification d'un poids  $w$  est donnée par :

$$\Delta w = -\eta \frac{\partial E}{\partial w} \quad (4.11)$$

Où :  $\eta$  est une constante positive (entre 0 et 1) appelée pas d'apprentissage.

Chaque poids est modifié à l'itération  $(r + 1)$  en fonction de sa valeur à l'itération  $(r)$  par :

$$w^{(r+1)} = w^{(r)} + \Delta w^{(r)} \quad (4.12)$$

Pour un MLP avec une couche cachée, la mise à jour des poids de la couche cachée et ceux de la couche de sortie est donnée par :

$$u_{mj}^{(r+1)} = u_{mj}^{(r)} - \eta \frac{\partial E^{(r)}}{\partial u_{mj}} \quad (4.13)$$

$$w_{nm}^{(r+1)} = w_{nm}^{(r)} - \eta \frac{\partial E^{(r)}}{\partial w_{nm}} \quad (4.14)$$

$$v_{kn}^{(r+1)} = v_{kn}^{(r)} - \eta \frac{\partial E^{(r)}}{\partial v_{kn}} \quad (4.15)$$

Le développement des équations (4.14), (4.15), et (4.16) donne les équations suivantes :

$$u_{mj}^{(r+1)} = u_{mj}^{(r)} + \eta_1 (t_j - o_j) h'(s_j) z_m \quad (4.16)$$

$$w_{nm}^{(r+1)} = w_{nm}^{(r)} + \eta_2 \left( \sum_{j=1}^J (t_j - o_j) h'(s_j) u_{mj} \right) g'(r_m) y_n \quad (4.17)$$

$$v_{kn}^{(r+1)} = v_{kn}^{(r)} + \eta_3 \left( \sum_{j=1}^J \sum_{m=1}^M (t_j - o_j) h'(s_j) u_{mj} g'(r_m) w_{nm} \right) f'(l_n) x_k \quad (4.18)$$

Où :  $\eta_1$ ,  $\eta_2$ , et  $\eta_3$ , sont les pas d'apprentissage, et  $f$ ,  $g$ , et  $h$ , sont respectivement les fonctions d'activation des neurones de la première et la deuxième couches cachées, et de la couche de sortie. Dans le cas des fonctions sigmoïdes, les dérivées sont données par :

$$g'(s) = o(1-o) \quad (4.19)$$

Dans le cas d'une fonction d'activation tangente hyperbolique :

$$g'(s) = (1+o)(1-o) \quad (4.20)$$

#### 4.3.5 Généralisation et contrôle de complexité des réseaux de neurones

La généralisation est la tâche accomplie par le réseau lorsque son apprentissage est achevé. Elle peut être évaluée en testant le réseau sur des données qui n'ont pas servi à l'apprentissage. Elle est influencée essentiellement par quatre facteurs : la complexité du problème, l'algorithme d'apprentissage, la complexité de l'échantillon (le nombre d'exemples et la manière dont ils représentent le problème) et enfin la complexité du réseau (nombre de poids).

Parmi ces quatre facteurs, la complexité du réseau constitue un facteur important. En effet, lorsque l'on traite des données bruitées, l'objectif est de trouver le modèle optimal, présentant le meilleur compromis possible entre performance d'apprentissage et capacité de généralisation.

Le problème de la généralisation est souvent vu sous deux perspectives différentes. Dans la première, la taille du réseau est fixée (en accord avec la complexité du problème) et la question est : combien d'exemples d'apprentissage sont nécessaires pour atteindre une bonne généralisation ? Cette perspective est intéressante dans les applications où l'on a la possibilité d'acquérir autant d'exemples que l'on veut. Dans le second cas, le nombre d'exemples d'apprentissage est fixé et la question est : quelle taille du réseau donne la meilleure généralisation pour ces données ? On adopte cette perspective lorsqu'on est limité en termes de données d'apprentissage : il importe alors de déterminer quelle est la taille permettant au réseau de décrire au mieux les données en notre possession.

En effet, l'une des caractéristiques essentielles des réseaux de neurones artificiels est leur flexibilité et leur capacité d'adaptation à des problèmes nouveaux grâce à la modification des poids des connexions selon une règle d'apprentissage.

Comment déterminer le nombre de couches cachées et le nombre d'unités par couche cachée nécessaires pour réaliser une bonne approximation? La difficulté de répondre à cette question a quelquefois été considérée comme un inconvénient majeur des modèles connexionnistes puisqu'un mauvais choix peut conduire à de médiocres performances du réseau correspondant.

Les premières tentatives de résolution du problème de détermination de l'architecture ont consisté à tester plusieurs réseaux ayant des architectures différentes jusqu'à atteindre la performance désirée.

Ces dernières années, de nombreux travaux ont été consacrés au développement de méthodes d'optimisation de l'architecture des réseaux de neurones.

Les principaux algorithmes qui ont été proposés peuvent être classés en cinq familles : Les méthodes de pénalisation (*weight decay*), Les méthodes d'injection de bruit, Les algorithmes d'élagage (*pruning*), Les algorithmes constructifs ou ascendants, et enfin, les algorithmes directs [17].

#### 4.3.6 Propriétés des réseaux de neurones

Les réseaux de neurones à couches ont la propriété générale d'être des approximateurs universels parcimonieux. Il s'agit en fait de deux propriétés distinctes détaillées ci-dessous [25].

##### 4.3.6.1 La propriété d'approximation universelle

La propriété d'approximation universelle a été démontrée et s'énonce de la façon suivante :

*Toute fonction bornée suffisamment régulière peut être approchée uniformément, avec une précision arbitraire, dans un domaine fini de l'espace de ses variables, par un réseau de neurones comportant une couche de neurones cachés en nombre fini, possédant tous la même fonction d'activation, et un neurone de sortie linéaire.*

Le nombre de neurones cachés doit être choisi convenablement pour obtenir la précision voulue.

##### 4.3.6.2 La propriété de parcimonie

Lorsque l'on cherche à modéliser un processus à partir des données, on s'efforce toujours d'obtenir les résultats les plus satisfaisants possibles avec un nombre minimum de paramètres ajustables. Dans cette optique, il est démontré que :

*Si le résultat de l'approximation (c'est-à-dire la sortie du réseau de neurones) est une fonction non linéaire des paramètres ajustables, elle est plus parcimonieuse que si elle est une fonction linéaire de ces paramètres. De plus, pour des réseaux de neurones à fonction d'activation sigmoïdale, l'erreur commise dans l'approximation varie comme*

*l'inverse du nombre de neurones cachés, et elle est indépendante du nombre de variables de la fonction à approcher.*

*Par conséquent, pour une précision donnée, donc pour un nombre de neurones cachés donné, le nombre de paramètres du réseau est proportionnel au nombre de variables de la fonction à approcher.*

Ce résultat s'applique aux réseaux de neurones à fonction d'activation sigmoïdale puisque la sortie de ces neurones n'est pas linéaire par rapports aux poids synaptiques. Cette propriété montre l'intérêt des réseaux de neurones par rapport à d'autres approximateurs comme les polynômes dont la sortie est une fonction linéaire des paramètres ajustables : pour un même nombre d'entrées, le nombre de paramètres ajustables à déterminer est plus faible pour un réseau de neurones que pour un polynôme. Cette propriété devient d'autant plus intéressante dans le cas du filtrage de textes car le nombre d'entrées est typiquement de l'ordre de plusieurs dizaines.

#### **4.3.7 Conclusion sur les réseaux de neurones**

Après cette présentation générale on a pu montrer que les réseaux de neurones reposent sur des bases mathématiques solides qui permettent d'envisager des applications dans presque tous les domaines y compris industriel et à grande échelle, notamment dans le domaine de la classification. Il n'y a aucun doute que les réseaux de neurones présentent maintenant un outil qui peut alléger considérablement la tâche de toute personne travaillant dans différents domaines (statistiques, traitement du signal, automatique...,etc.) en permettant une approche efficace et générique des problèmes non linéaires.

#### **4.4 Réseaux de neurones et égalisation**

Les réseaux de neurones sont capables d'apporter des solutions à des problèmes complexes en communications numériques grâce à leur traitement non linéaire, leur architecture parallèlement distribuée, leur auto-organisation, leur capacité d'apprentissage et de généralisation et leur implantation efficace.

Parmi les applications des réseaux de neurones aux communications numériques on trouve, l'identification, l'égalisation, le codage et le décodage, la quantification vectorielle, le traitement d'images, le filtrage non linéaire, les techniques d'étalement de spectre, etc. Le point clé pour une utilisation efficace des réseaux de neurones est de trouver une architecture adaptée au problème et qui donne les meilleurs résultats [19].

Les égaliseurs à base des réseaux de neurones ont été proposés comme une alternative des égaliseurs classiques, et ont fourni des améliorations significatives en performance dans une variété des canaux de transmissions.

Plusieurs architectures des réseaux de neurones sont mises en œuvre pour l'égalisation des canaux de communications, telles que, le perceptron multicouche (MLP), les réseaux de fonctions à base radiale (RBF), les réseaux récurrents (RNN), les cartes auto-organisatrices, etc. Les égaliseurs MLP, en particulier, sont capables de générer des courbes de séparation non-linéaires complexes et donc capables d'égaliser fortement les canaux non linéaires [20].

On distingue deux types des égaliseurs à base du réseau MLP, l'égaliseur MLP direct et l'égaliseur DFE-MLP, qui est la version neuronale de l'égaliseur DFE. La **figure 4.9** montre les schémas illustratifs des deux égaliseurs neuronaux, où (a) représente le MLP direct et (b) le DFE-MLP.

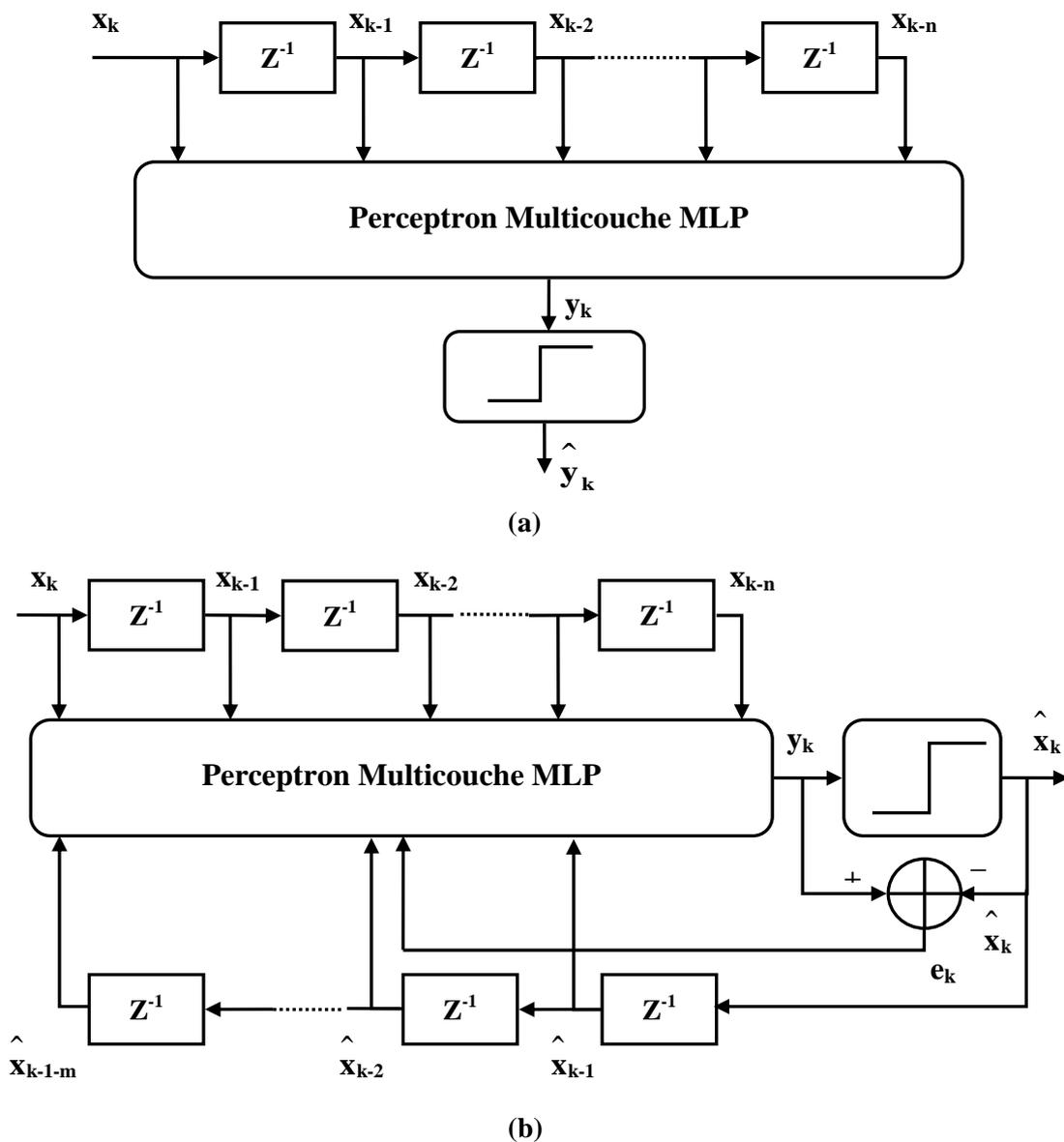


Figure 4.9 Egaliseur à base du perceptron multicouche, (a) MLP, (b) DFE-MLP

## 4.5 Performances des égaliseurs et résultats de simulation

Nous présentons ici les résultats de simulation obtenus en utilisant le logiciel MTLAB pour les égaliseurs à base des réseaux de neurones en faisant une comparaison de leurs performances. Cette comparaison se fait sur les signaux transmis/égalisés, les courbes de convergence de l'EQM, ainsi que les diagrammes de l'œil et les courbes BER.

Le canal de transmission utilisé est celui du chapitre précédent, il définit par :

$$H(z) = 0.3482 + 0.8704 z^{-1} + 0.3482 z^{-2} \quad (4.21)$$

La séquence de données est constituée des symboles (-1,1) uniformément distribués, le rapport signal sur bruit SNR est de 20 dB.

### 4.5.1 Structure de l'égaliseur MLP

Pour l'égalisation du canal, la couche de sortie du réseau contient uniquement un seul neurone, un tel réseau (MLP) produit une application  $f: R^m \rightarrow R$ .

Où  $R^m$  est l'ensemble d'entrée,  $R$  est l'ensemble de sortie,  $f$  est la fonction réalisée par le réseau.

Bien que les performances désirées puissent être réalisées avec toutes précisions spécifiées par un perceptron à deux couches (couche cachée, et couche de sortie), un nombre très élevé de neurones dans la couche cachée peut être demandé. Or, un perceptron à trois couches (deux couches cachées et une couche de sortie) peut atteindre la même précision de classification avec un nombre moins, de neurones dans les couches cachées en comparaison avec un réseau à deux couches [20], ce qui nous amène à employer dans notre simulation un réseau à trois couches.

Notons que, si un grand nombre de neurones cachés est employé, la convergence du processus d'apprentissage peut être très lente. En effet, il est très difficile, voire impossible, de trouver avec précision le nombre optimal de neurones nécessaires à chaque couche cachée [20].

On utilise la notation  $(m, n1-n2-o)$  pour désigne un égaliseur à  $m$  entrées,  $n1$  neurones dans la première couche cachée,  $n2$  neurones dans la deuxième couche cachée, et  $o$  neurones dans la couche de sortie. Dans cette simulation, les structures utilisées reposent sur le principe des expériences (Essai/Erreur).

En ce qui concerne l'égaliseur DFE-MLP, il se distingue du premier par son entrée. En fait, il garde ces  $m$  entrées, dont  $m-1$  représentant les versions retardées du signal en sortie du canal, comme indiqué sur la **figure 4.9.b**, et une entrée qui n'est que la sortie retardée de l'égaliseur.



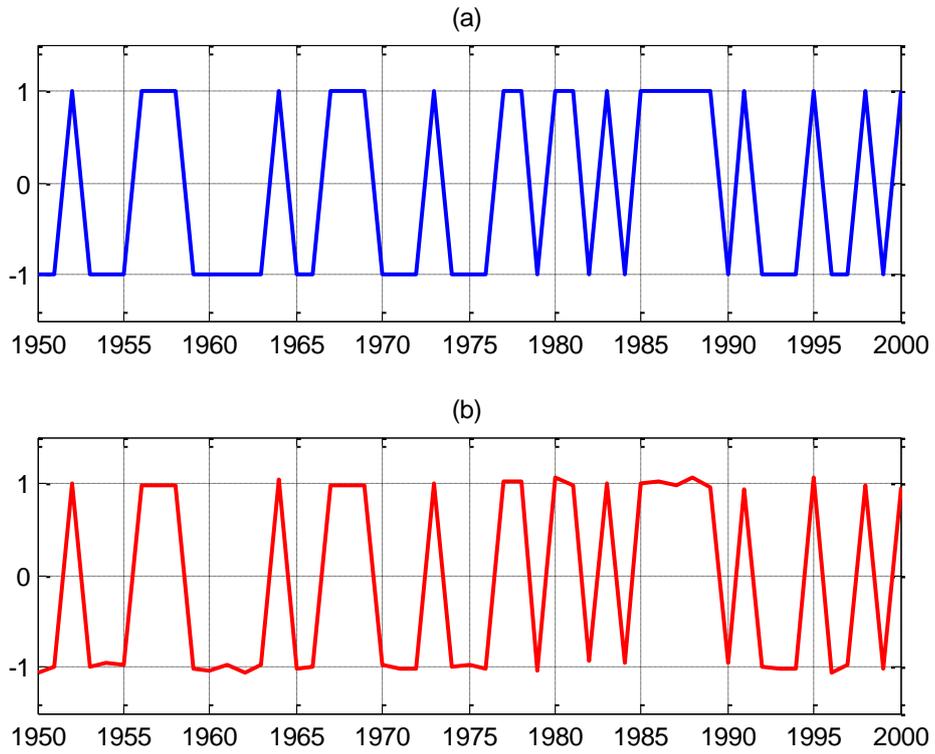


Figure 4.11 signal transmis et signal égalisé DFE-MLP (BP)

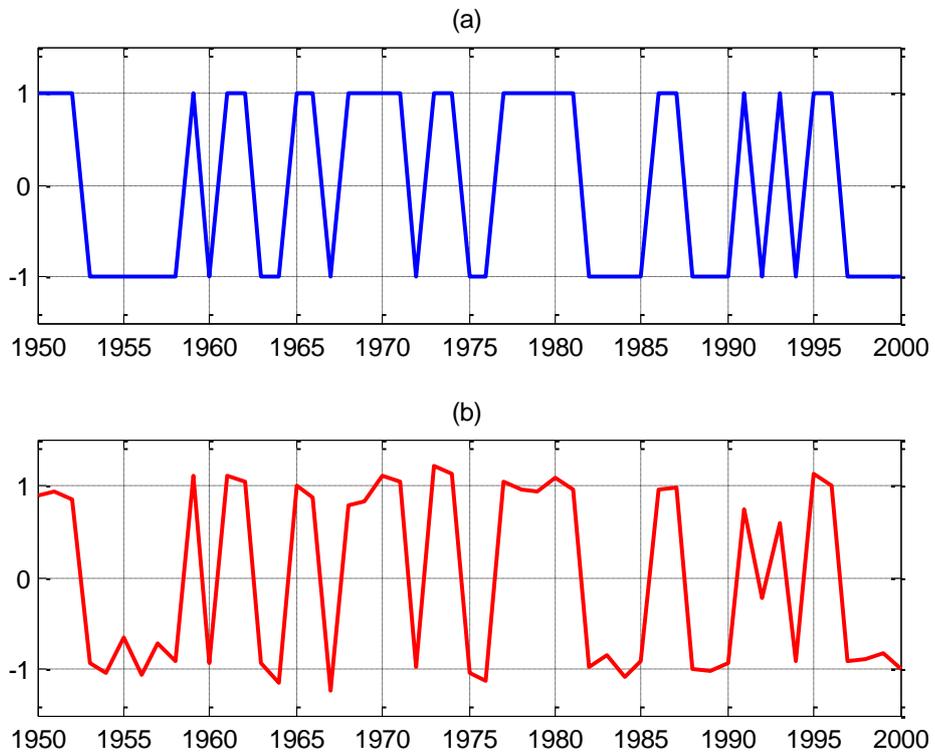


Figure 4.12 signal transmis et signal égalisé DFE-RBF (K-Means/BP)

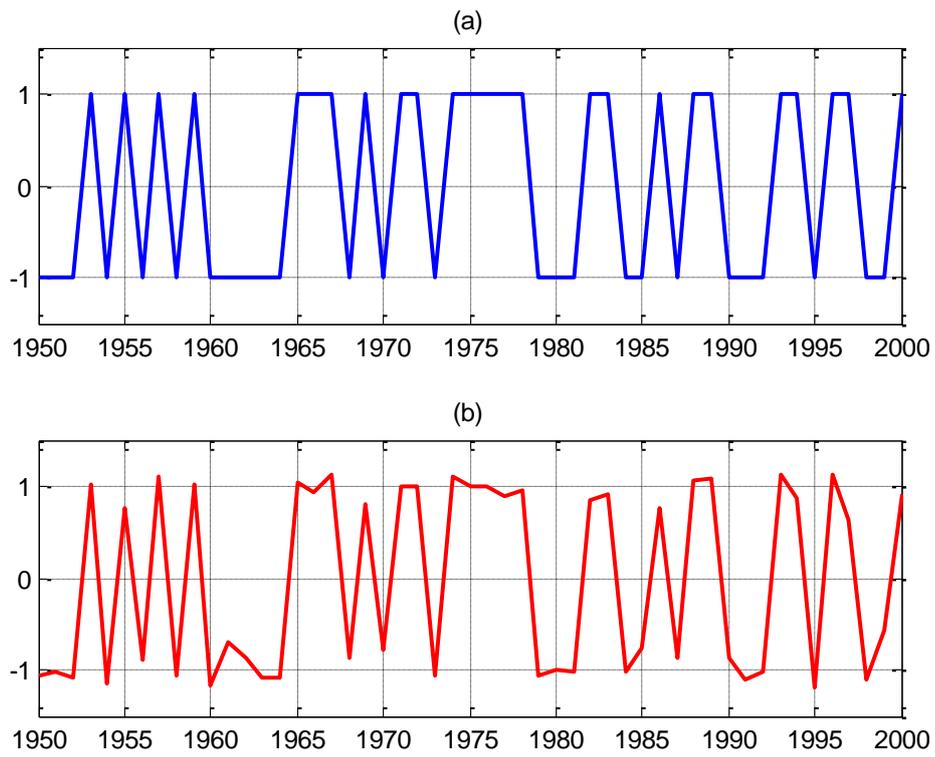


Figure 4.13 signal transmis et signal égalisé DFE-RBF (BP)

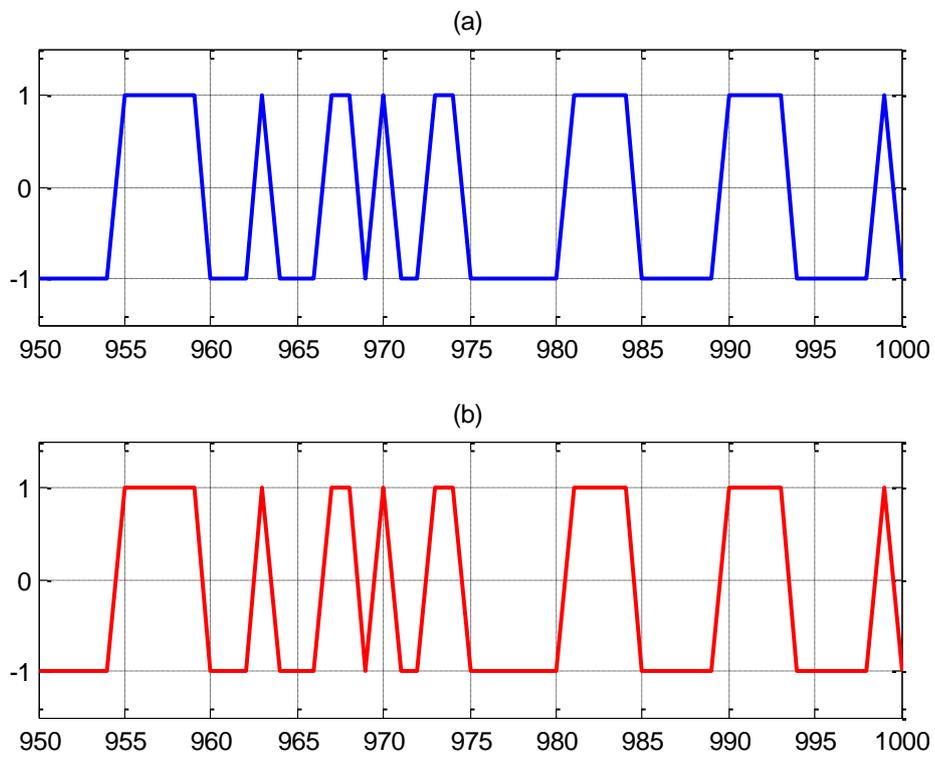


Figure 4.14 signal transmis et signal égalisé DFE-MLP (LM)

### 4.5.3 Diagramme de l'œil

Il est bien connu que le diagramme de l'œil est un autre indicateur de l'efficacité du processus d'égalisation. Les figures 4.15 à 4.19 représentent les diagrammes de l'œil des signaux aux sorties des égaliseurs, MLP-BP, DFE-MLP-BP, DFE-RBF-K means, DFE-RBF-BP et respectivement. Ces figures montrent clairement que les signaux en sorties des égaliseurs proposés sont bien concentrés sur les valeurs souhaitées  $\pm 1$ , avec une ouverture de l'œil qui se diffère d'un égaliseur à l'autre. Ce qui confirme les performances meilleurs de l'égaliseur DFE-MLP-LM, relativement aux autres égaliseurs.

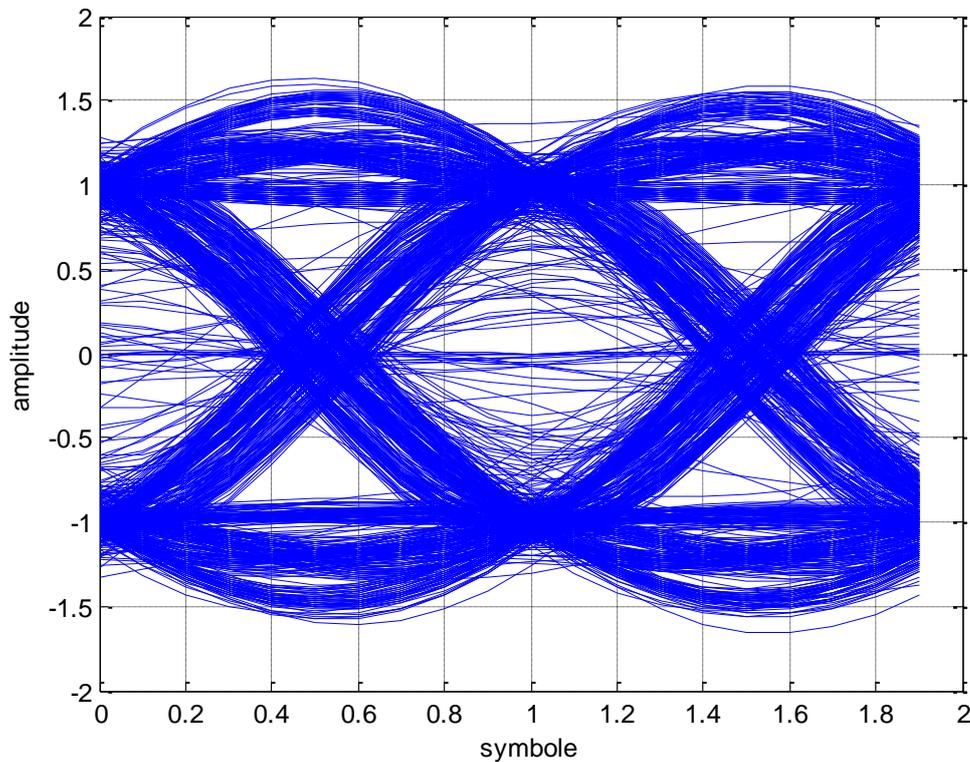
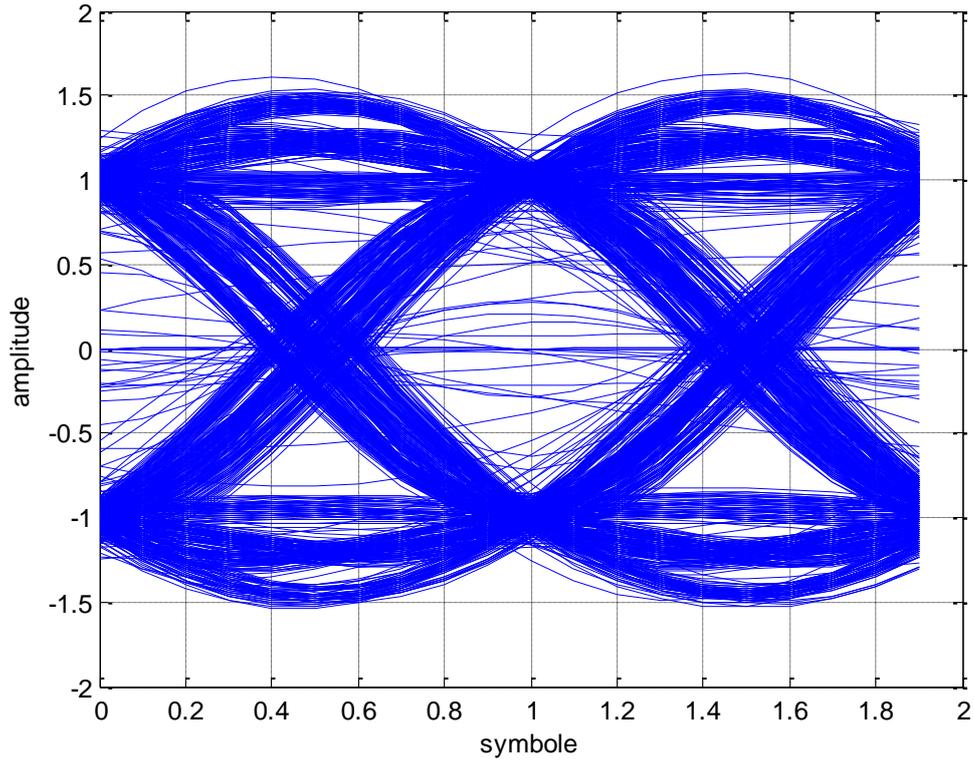
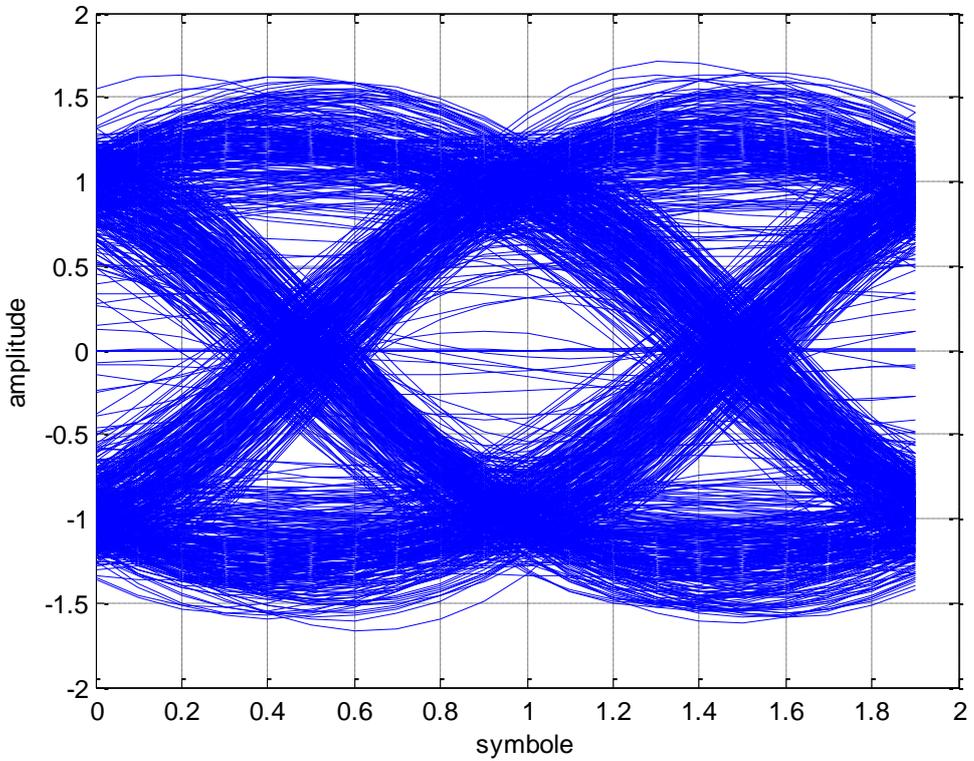


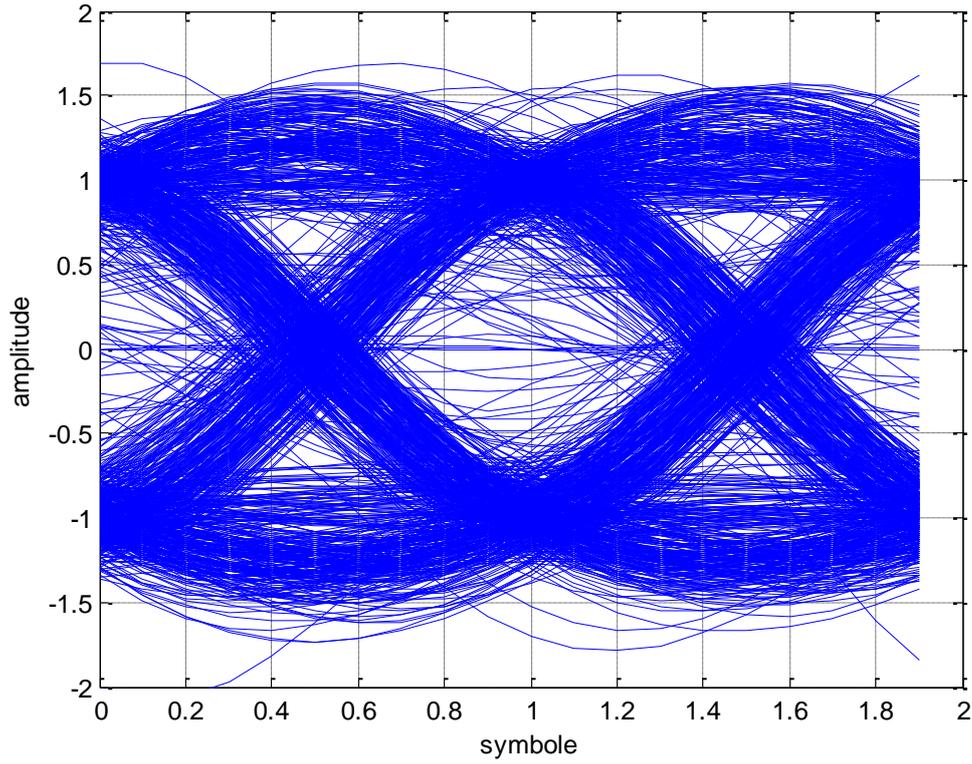
Figure 4.15 Diagramme de l'œil de l'égaliseur MLP (BP)



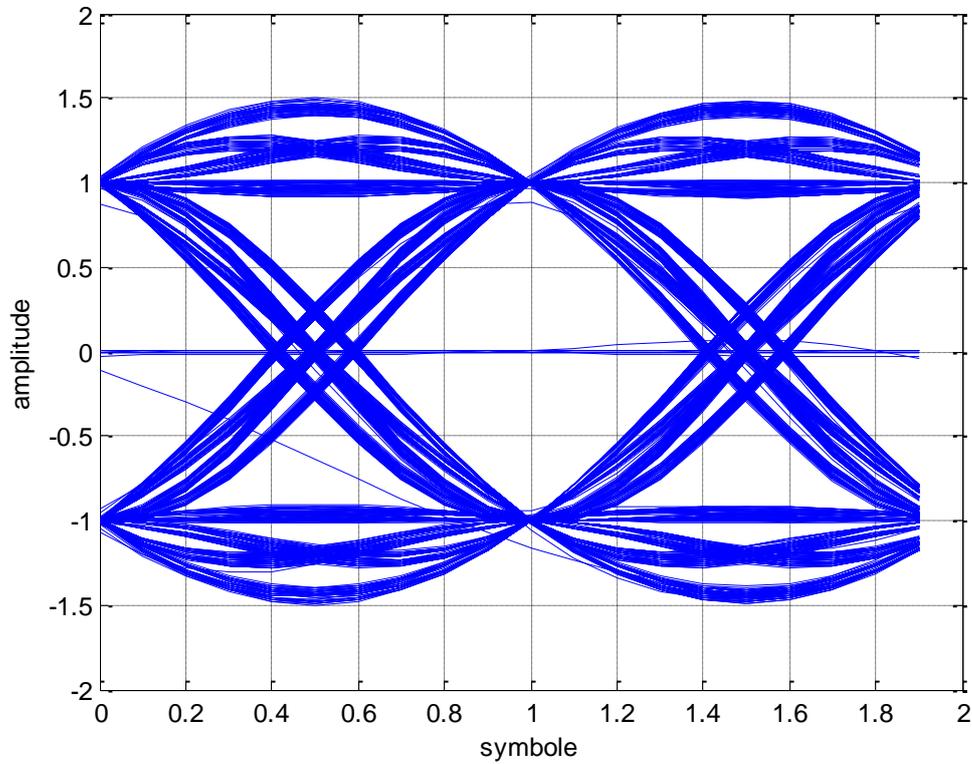
**Figure 4.16** Diagramme de l'œil de l'égaliseur DFE-MLP (BP)



**Figure 4.17** Diagramme de l'œil de l'égaliseur DFE-RBF (K-means/BP)



**Figure 4.18** Diagramme de l'œil de l'égaliseur DFE-RBF (BP)



**Figure 4.19** Diagramme de l'œil de l'égaliseur DFE-MLP (LM)

#### 4.5.4 Comparaison des courbes MSE

Les figures 4.20, 4.21 et 4.22 présentent les courbes de convergence de l'erreur quadratique moyenne (MSE) des égaliseurs considérés. Ces figures montrent une amélioration claire de performance par rapport aux résultats obtenus à l'aide des égaliseurs linéaires (LTE, et DFE). Les égaliseurs MLP-LM (considérés comme des égaliseurs non linéaires) offrent les meilleurs résultats et atteignent le niveau -20 dB de l'erreur quadratique moyenne (écart entre le signal égalisé et le signal désiré au sens des moindres carrés) après uniquement 40 échantillons (symboles). Le même niveau est atteint après 600 échantillons pour le MLP-BP et 200 échantillons pour le DFE-MLP-BP. Cependant il est constaté que ce niveau n'est pas atteint dans le cas des égaliseurs RBF qui se stabilisent autour de -17 dB.

Pour l'égaliseur DFE-MLP-LM, il est constaté que l'erreur quadratique moyenne dépasse le niveau de -50dB ce qui prouve la puissance de cet égaliseur par rapport aux autres.

Il est à signaler au passage que les courbes MSE des égaliseurs MLP et DFE-MLP préservent leurs allures décroissantes même après  $5 \cdot 10^6$  échantillons (symboles) [22].

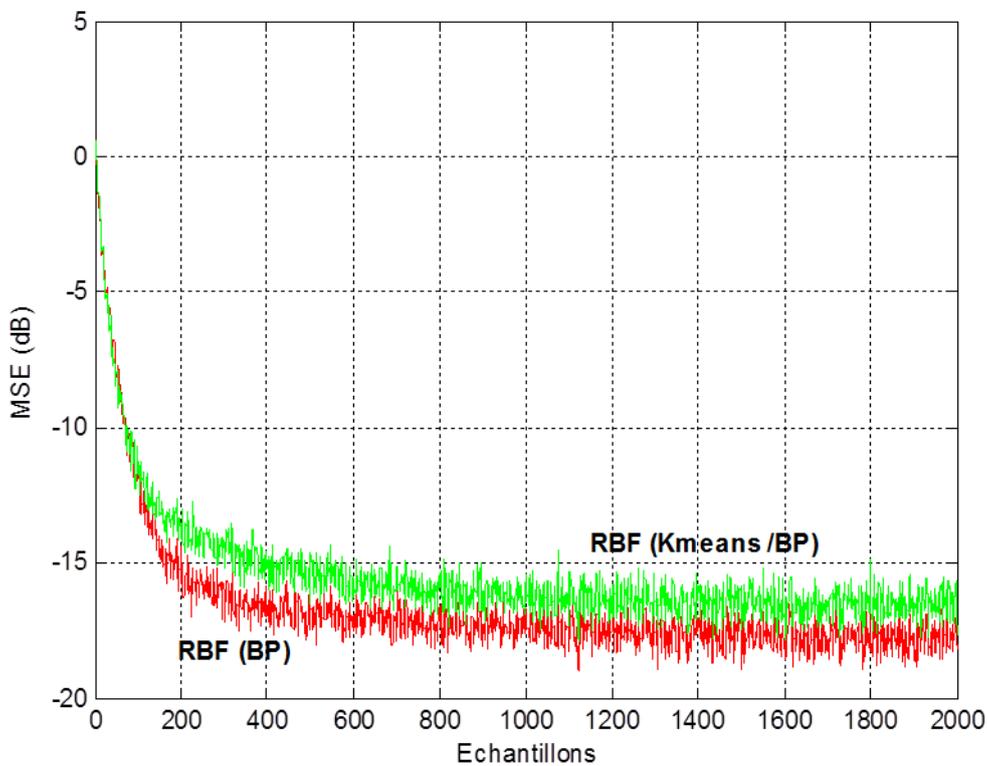


Figure 4.20 Courbes de convergence de l'EQM des égaliseurs DFE-RBF

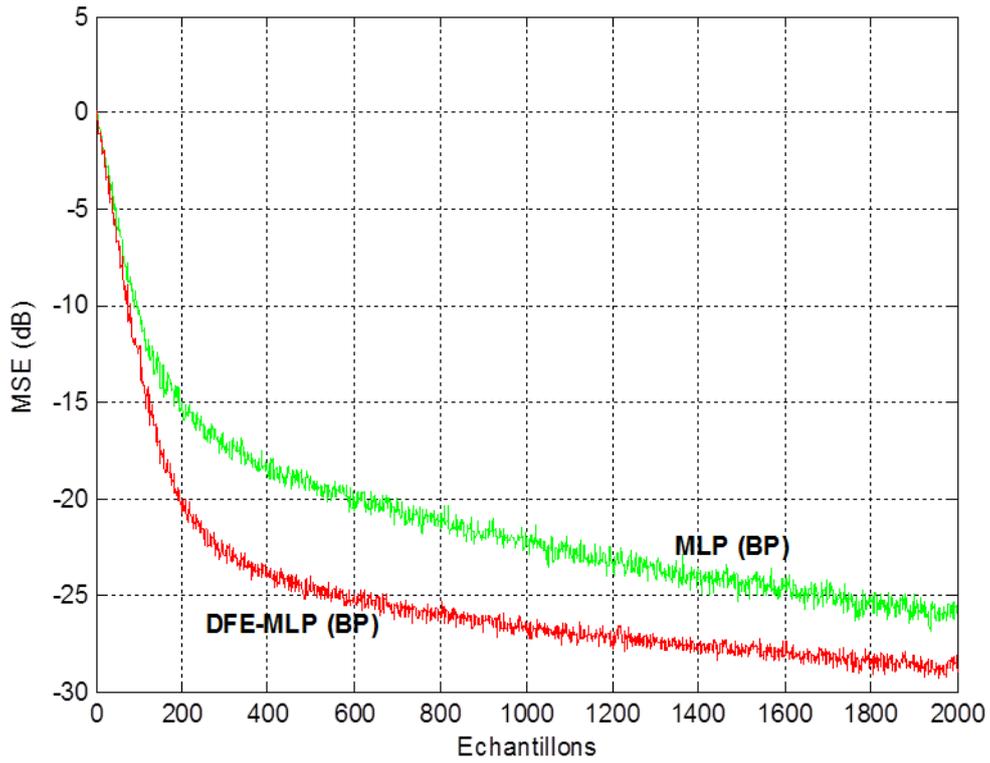


Figure 4.21 courbes de convergence de l'EQM des égaliseurs MLP (BP)

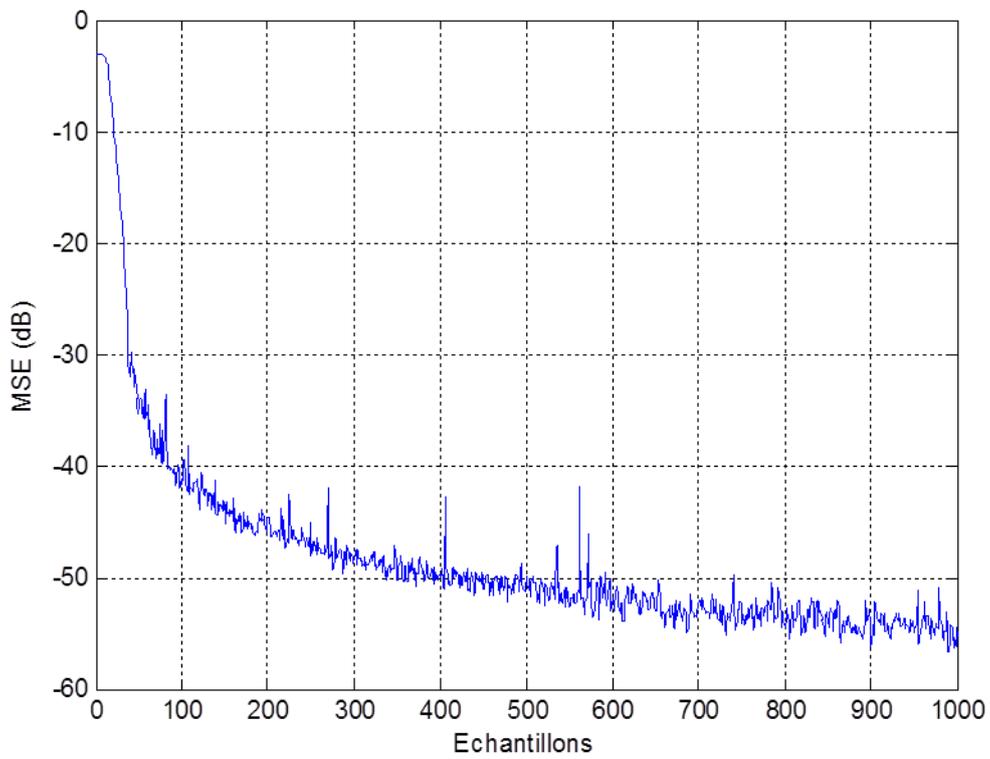
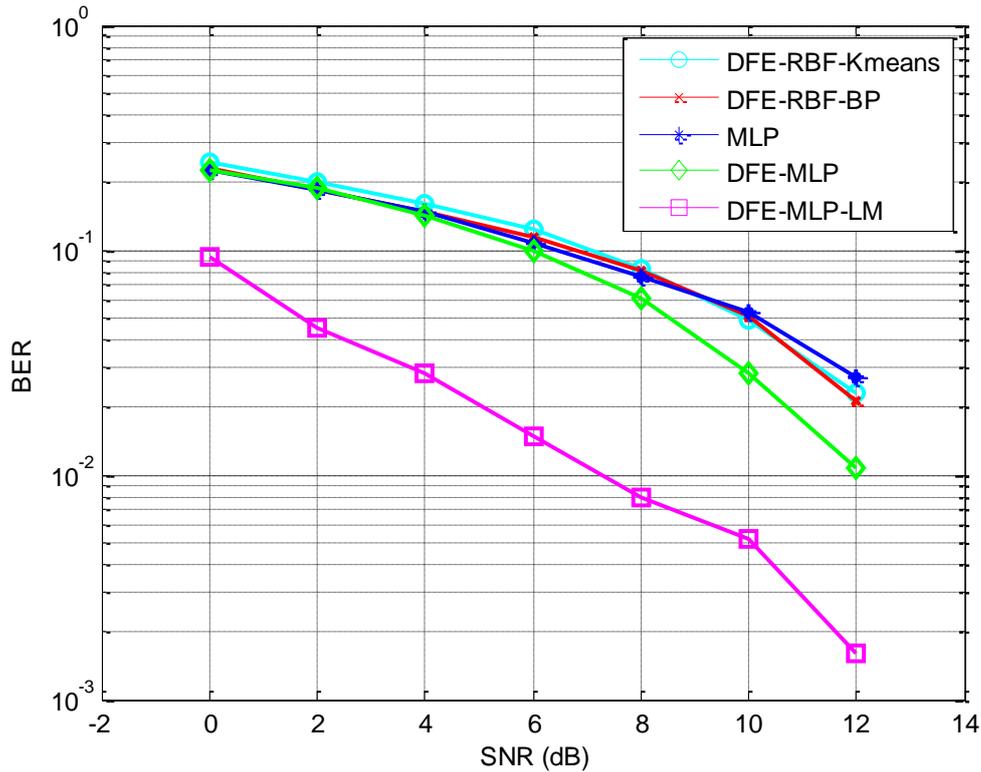


Figure 4.22 courbes de convergence de l'EQM de l'égaliseur DFE-MLP (LM)

### 4.5.5 Comparaison de performance BER

La **figure 4.23** montre Les courbes du taux d'erreur binaire (BER) des égaliseurs considérés. Ces courbes sont obtenues avec une séquence de  $10^4$  symboles et une variation du rapport signal sur bruit SNR entre 2dB et 12dB.

Les courbes BER agrément les résultats atteints à l'aide des autres critères de performances et démontrent avec clarté la même classification des égaliseurs en termes de performances, en prouvant la supériorité de l'égaliseur DFE-MLP-LM.



**Figure 4.23** Courbes BER des différents égaliseurs.

## 4.6 Conclusion

Nous avons utilisé, dans ce chapitre, les réseaux de neurones dans la fonction d'égalisation. Nous avons examiné la fonctionnalité et les performances de ces égaliseurs en tenant compte du type du réseau de neurones et des algorithmes d'apprentissage utilisés.

La comparaison des différents égaliseurs considérés dans notre travail a montré la supériorité flagrante de l'égaliseur DFE-MLP (LM), de part son niveau d'EQM (le plus faible), son diagramme de l'œil (très proche à celui du signal original) et son BER (le plus réduit).

# Conclusion Générale

---

Ce mémoire est consacré à l'égalisation adaptative des canaux de transmission, en utilisant des structures à base des réseaux de neurones artificielles et ce pour palier les inconvénients des égaliseurs conventionnels (l'égaliseur linéaire transversal LTE et l'égaliseur à retour de décision DFE basés sur les filtres numériques).

Notre choix des réseaux de neurones comme outils pour l'égalisation est motivé par les travaux de **M. Ibnkahla** [19] concernant les applications des réseaux de neurones à la fonction l'égalisation, et aux communications numériques en général.

Et en vue d'améliorer davantage la fonction d'égalisation, nous avons considéré la combinaison des structures conventionnelles (LTE et DFE) avec les structures neuronales proposées.

Le travail nous a permis d'appréhender d'une manière méthodologique l'opération d'égalisation et les problèmes ayant trait.

D'une manière générale, nous avons eu l'occasion d'introduire des notions générales sur les systèmes de communications numériques, telles que les modèles des canaux de transmissions, les modulations numériques et notamment le bruit et l'interférence entre symboles qui représentent des vrais problèmes justifiant le recours à l'égalisation.

De même, Nous avons donné un aperçu sur les égaliseurs conventionnels (LTE et DFE) et nous avons testé leurs performances par les courbes de convergence des erreurs quadratiques moyennes (MSE) et le taux d'erreur binaire (BER).

La mise en œuvre des réseaux de neurones (MLP et RBF) dans le contexte d'égalisation est réalisée après avoir donné un aperçu sur leurs structures et leurs différents algorithmes d'apprentissage.

En tenant compte du type du réseau de neurones et des algorithmes d'apprentissage utilisés, la comparaison des différents égaliseurs considérés dans notre travail a montré une nette supériorité en termes de performances de l'égaliseur DFE-MLP (LM), de part son niveau d'erreur quadratique moyenne MSE (le plus faible), son diagramme de l'œil (très proche à celui du signal original) et son BER (le plus réduit).

Cependant, l'éventuelle implémentation en temps réel de ces techniques d'égalisation nécessite une étude plus approfondie en tenant compte de toutes les contraintes de l'environnement de calcul considéré et des contraintes du temps.

## Bibliographie

- [1] **John G. Proakis.** « Digital Communications », McGraw-Hill 4th Edition 2000
- [2] **Michel Joindot, Alain Glavieux.** « Introduction aux communications numériques », DUNOD 2007.
- [3] **Steven W. Smith** « Digital Signal Processing, A Practical Guide for Engineers and Scientists », Newnes, Elsevier Science, 2003.
- [4] **Geneviève Baudoin et coll.** « Radiocommunications Numériques/1, Principes, modélisation, et simulation » 2<sup>e</sup> édition, DUNOD 2007.
- [5] **Jean-Marc Brossier.** « Signal et communication numérique : égalisation et synchronisation », HERMES 1997.
- [6] **Christophe LAOT.** « Egalisation autodidacte et turbo-égalisation. Applications aux canaux sélectifs en fréquences », Thèse de Doctorat, Université Rennes 1, 1997.
- [7] **ZERDOUMI Zahra.** « Applications de Réseaux de Neurones Artificiels à la Poursuite des non Linéarités fluctuantes des systèmes satellitaire », Mémoire de Magister, Université Mohamed Boudiaf, M'sila, 2006.
- [8] **Edmund Lai.** « Practical Digital Signal Processing for Engineers and Technicians », Newnes, Elsevier, 2003.
- [9] **B. Farhang-Boroujeny.** « Adaptive Filters, Theory and Applications », John Wiley and Sons Ltd, 1998.
- [10] **Andrea Goldsmith.** « Wireless Communications », Cambridge University Press 2005.
- [11] **S. Qureshi.** « Adaptive equalization », Proceeding of IEEE, vol. 73, no.9, pp.1349–1387, Sep. 1985.
- [12] **Simon Haykin.** « Communication systems », John Wiley and Sons Inc, 4th Edition 2001.

- [13] **Kashif Mahmood, Abdelmalek Zidouri, Azzedine Zerguine**, « Performance analysis of a RLS-based MLP-DFE in time-invariant and time-varying channels », *Digital Signal Processing* 18, 307–320, 2008.
- [14] **A. Zerguine, A. Shafi, and M. Bettayeb**. « Multilayer perceptron based DFE with lattice structure », *IEEE Transaction on Neural Networks*, vol.12, no.3, pp.532–545, May. 2001.
- [15] **Haiquan Zhao, Xiangping Zeng, Xiaoqiang Zhang, Jiashu Zhang, Yangguang Liu, Tiao Wei**. « An adaptive decision feedback equalizer based on the combination of the FIR and FLNN », *Digital Signal Processing*, (2011)
- [16] **S. Chen, B. Mulgrew and P. M. Grant**. « A Clustering Technique for Digital Communications Channel Equalization Using Radial Basis Function Networks » *IEEE Transactions on Neural Networks*, vol. 4, no. 4, pp.570-579, July 1993.
- [17] **A. K MOUSSAOUI**. « Contribution à la Modélisation et la Commande des Processus Industriels par Réseaux de Neurones », Thèse de Doctorat en Electronique, option Automatique, Université BADJI Mokhtar, Annaba, 2006.
- [18] **Mohamed NEMISSI**. « Classification et reconnaissances des formes par algorithmes hybrides », Thèse de Doctorat, Université 8 mai 45, Guelma, 2009.
- [19] **Mohamed IBNKAHLA**. « Applications of neural networks to digital communications », *Signal Processing* 80, pp 1185-1215, 2000.
- [20] **S. Chen, G.J. Gibson. C. Cowan, P. Grant**. « Adaptive equalization of finite non-linear channels using multilayer perceptron », *Signal Processing* 20 107-119, 1990.
- [21] **A. Shafi, M. Bettayeb, A. Zerguine**, « Neural network-based decision feedback equalizer with lattice structure », *Electronics Letters*, Vol. 35, No. 20, September 1999.
- [22] **Sapthotharan K. Nair, Jaekyun Moon**, « Data storage channel equalization using neural networks », *IEEE Transaction In Neural Networks*, Vol. 8, No. 5. pp 1037-1048, 1997.

- [23] **Ananth Ranganathan**, « The Levenberg-Marquardt Algorithm », 2004.
- [24] **Martin T. Hagan and Mohammad B. Menhaj**, « Training feedforwards Networks with the Levenberg-Marquardt Algorithm », IEEE Transactions on Neural Networks, Vol. 5, No. 6. pp 989-993, November 1994.
- [25] **Mathieu Stricker**. « Réseaux de neurones pour le traitement automatique du langage : conception et réalisation de filtres d'informations », Thèse de Doctorat, Université Paris 6, 2000.
- [26] **F. Rotella, P. Borne**, « Théorie et pratique du calcul matriciel », Editions TECHNIP, 1995.
- [27] **Gérard Dreyfus**, « Apprentissage statistiques », Editions EYROLLES, 2008.

# Annexe 1

## Algorithme des moindres carrés LMS (Lest Mean Square)

L'algorithme LMS proposé par **Widrow** et **Hoff** en 1960, est l'algorithme le plus utilisé dans le filtrage adaptatif à cause de sa simplicité et sa robustesse. On considère un filtre transversal à  $n$  coefficients, son entrée  $x(n)$ , sa sortie désirée  $d(n)$ , et sa sortie  $y(n)$  est donnée par :

$$y(n) = \sum_{i=0}^N w(i)x(n-i) \quad (\text{A1.1})$$

Cette sortie est supposée à valeurs réelles, les coefficients  $w(i), \dots, w(n)$  sont ajustés pour minimiser la différence :

$$e(n) = d(n) - y(n) \quad (\text{A1.2})$$

L'algorithme LMS change les coefficients du filtre pour que l'erreur soit minimisée au sens des moindres carrés, d'où le nom, algorithme de moindres carrés.

L'algorithme LMS est une implémentation stochastique de l'algorithme du gradient déterministe, il remplace la fonction coût  $J=E(e^2)$  par son estimé instantané  $J(n)=[e(n)]^2$ , ainsi, et en se retournant à l'algorithme du gradient déterministe, on obtient :

$$w(n+1) = w(n) - \mu \nabla e^2(n) \quad (\text{A1.3})$$

Où  $w(n)=[w_0(n), w_1(n), \dots, w_{N-1}(n)]^T$ , et  $\mu$  le pas d'apprentissage, et  $\nabla$  est le gradient :

$$\nabla = \left[ \frac{\partial}{\partial w_0} \quad \frac{\partial}{\partial w_1} \quad \dots \quad \frac{\partial}{\partial w_{N-1}} \right]^T \quad (\text{A1.4})$$

On note que le  $i^{\text{ème}}$  élément du gradient  $\nabla e^2(n)$  est :

$$\frac{\partial e^2(n)}{\partial w_i} = 2 e(n) \frac{\partial e(n)}{\partial w_i} \quad (\text{A1.5})$$

Prenant l'équation (A1.2), et notant que  $d(n)$  est indépendant de  $w_i$ , on remplace dans l'équation (A1.5) :

$$\frac{\partial e^2(n)}{\partial w_i} = -2 e(n) \frac{\partial y(n)}{\partial w_i} \quad (\text{A1.6})$$

On remplace  $y(n)$  par sa valeur dans l'équation (A1.1) :

$$\frac{\partial e^2(n)}{\partial w_i} = -2 e(n)x(n-i) \quad (\text{A1.7})$$

En utilisant les équations (A1.4) et (A1.7) on obtient :

$$\nabla e^2(n) = -2e(n)x(n) \quad (\text{A1.8})$$

Où  $x(n) = [x(n), x(n-1), \dots, x(n-N+1)]^T$ , on substituant (A1.8) dans (A1.3), on obtient :

$$w(n+1) = w(n) + 2 e(n)x(n) \quad (\text{A1.9})$$

La propriété éminente de l'algorithme LMS qui en fait le plus populaire pour le filtrage adaptatif est sa simplicité, en effet, son implémentation nécessite  $2N+1$  multiplications ( $N$  multiplications pour calculer  $y(n)$ , une multiplication pour  $(2\mu) \times e(n)$ , et  $N$  multiplications pour calculer  $2\mu e(n) \times x(n)$ ) et  $2N$  additions.

### Résumé de l'algorithme LMS :

Filtrage :

$$y(n) = \sum_{i=0}^N w(i)x(n-i)$$

Estimation d'erreur :

$$e(n) = d(n) - y(n)$$

Adaptation des coefficients :

$$w(n+1) = w(n) + 2 e(n)x(n)$$

## Annexe 2

### Algorithme des moindres carrés récurrents RLS (Recurrent Least Square)

Les algorithmes RLS sont connus pour leur rapidité de convergence. Ces algorithmes ont d'excellentes performances dans des environnements variables dans le temps, au coût d'une complexité accrue de calcul, et certains problèmes de stabilité, qui ne sont pas aussi critiques dans les algorithmes basés sur LMS.

L'objectif ici est de choisir les coefficients du filtre adaptatif tel que le signal de sortie  $y(k)$ , pendant la période d'observation, correspond au signal désiré aussi proche que possible dans le sens des moindres carrés. Le processus de minimisation nécessite le signal d'entrée à ce moment d'observation. En outre, la fonction que nous cherchons à minimiser est déterministe.

Le signal d'entrée d'un filtre RIF à l'instant  $k$  est donné par :

$$x(n) = [x(n), x(n-1), \dots, x(n-N)]^T \quad (\text{A 2.1})$$

Où  $N$  est l'ordre du filtre.  $w_j(k)$ ,  $j=0, 1, \dots, N$ , sont les coefficients ajustés pour minimiser une fonction coût. Dans le cas des algorithmes des moindres carrés la fonction coût est déterministe et donnée par :

$$J(n) = \sum_{i=0}^n \lambda^{n-i} e^2(i) \quad (\text{A 2.2})$$

$$J(n) = \sum_{i=0}^n \lambda^{n-i} [d(i) - x^T(i)w(k)]^2 \quad (\text{A 2.3})$$

$e(i)$  est l'erreur à posteriori à l'instant  $i$ , c'est-à-dire l'erreur calculée après la mise à jour des coefficients, en tenant  $x(k)$  le plus récent.  $\lambda$  est le facteur d'oubli, choisi tel que  $0 << \lambda < 1$ , ce facteur est appelé facteur d'oubli car il néglige (oublie) d'une manière croissante l'effet des données passées sur la mise à jour des coefficients [1].

On dérive la fonction  $J(n)$  par rapport à  $w(k)$  :

$$\frac{\partial J(n)}{\partial w(n)} = -2 \sum_{i=0}^n \lambda^{n-i} x(i) [d(i) - x^T(i)w(n)] \quad (\text{A 2.4})$$

Il est possible de trouver le vecteur  $w(k)$  optimal qui minimise l'erreur quadratique minimale, par la relation suivante :

$$-\sum_{i=0}^n \lambda^{n-i} x(i) x^T(i) w(n) + \sum_{i=0}^n \lambda^{n-i} x(i) d(i) = [0 \dots 0]^T \quad (A 2.5)$$

Le vecteur des coefficients optimaux est donné par :

$$\left[ \sum_{i=0}^n \lambda^{n-i} x(i) x^T(i) \right] w(n) = \sum_{i=0}^n \lambda^{n-i} x(i) d(i) \quad (A 2.6)$$

$$R_D(n) w(n) = p_D(n)$$

Où  $R$  et  $p$  sont respectivement, la matrice de corrélation du vecteur d'entrée  $x(n)$ , et le vecteur d'intercorrélation de  $x(n)$  et  $d(n)$ .

$$\left[ \sum_{i=0}^n \lambda^{n-i} x(i) x^T(i) \right] w(n) = \lambda \left[ \sum_{i=0}^{n-1} \lambda^{n-i-1} x(i) d(i) \right] + x(n) d(n) \quad (A 2.7)$$

Sachant que :

$$R_D(n-1) w(n-1) = p_D(n-1) \quad (A 2.8)$$

L'équation (A2.7) se réécrit :

$$\begin{aligned} R_D(n) w(n) &= R_D(n) w(n-1) - x(n) x^T(n) w(n-1) + x(n) d(n) \\ &= R_D(n) w(n-1) + [d(n) - x^T(n) w(n-1)] x(n) \end{aligned} \quad (A 2.9)$$

Soit :

$$e(n) = d(n) - x^T(n) w(n-1) \quad (A 2.10)$$

Alors :

$$w(n) = w(n-1) + e(n) S_D(n) x(n) \quad (A 2.11)$$

Où  $S_D(n)$  est l'inverse de la matrice  $R_D(n)$  calculée en utilisant le lemme d'inversion d'une matrice [Annexe 5], on a alors :

$$S_D(n) = R_D^{-1}(n) = \frac{1}{\lambda} \left[ S_D(n-1) - \frac{S_D(n-1) x(n) x^T(n) S_D(n-1)}{\lambda + x^T(n) S_D(n-1) x(n)} \right] \quad (A 2.12)$$

**Résumé de l'algorithme RLS :**

Initialisation :

$$S_D(-1) = \delta I$$

Où  $\delta$  peut être l'inverse du signal d'entrée.

$$x(-1) = w(-1) = [0 \dots 0]^T$$

Pour  $n \geq 0$  :

$$e(n) = d(n) - x^T(n) w(n-1)$$

$$S_D(n) = R_D^{-1}(n) = \frac{1}{\lambda} \left[ S_D(n-1) - \frac{S_D(n-1)x(n)x^T(n)S_D(n-1)}{\lambda + x^T(n)S_D(n-1)x(n)} \right]$$

$$w(n) = w(n-1) + e(n) S_D(n) x(n)$$

# Annexe 3

## Méthode de Levenberg-Marquardt

### Méthodes d'optimisation du second ordre :

Les méthodes d'optimisation du second ordre (Elles sont dites du second ordre parce qu'elles prennent en considération la dérivée seconde de la fonction de coût) sont des méthodes itératives de descente du gradient qui consistent à remplacer la fonction de coût par son approximation en série de Taylor du premier ordre au voisinage du poids courant, tel que la fonction de coût est :

$$f(\omega) = \frac{1}{2} \sum_{j=1}^M e_j^2(\omega) \quad (\text{A 3.1})$$

L'approximation en série de Taylor du premier ordre de la fonction de coût est :

$$f(\omega) = f(\omega_0) + (\omega - \omega_0) \nabla f(\omega_0) \quad (\text{A 3.2})$$

Où  $\nabla$  est le gradient, et :

$$e_j(\omega) = (t_j - y_j(\omega)) \quad (\text{A 3.3})$$

En considérant l'équation suivante :

$$e(\omega) = [e_1(\omega), \dots, e_m(\omega)] \quad (\text{A 3.4})$$

La dérivée, et la dérivée seconde de la fonction de coût deviennent :

$$\nabla f(\omega) = \sum_{j=1}^m e_j(\omega) \nabla e_j(\omega) = J^T(\omega) e(\omega) \quad (\text{A 3.5})$$

$$\nabla^2 f(\omega) = J^T(\omega) J(\omega) + \sum_{j=1}^m e_j(\omega) \nabla^2 e_j(\omega) \quad (\text{A 3.6})$$

La propriété spécifique des problèmes aux moindres carrés est que, étant donné le jacobien  $J$  (formule A3.14), on peut facilement calculer le hessien s'il est possible de faire une approximation des  $e_j$  par des fonctions linéaires ( $\nabla^2 e_j(\omega)$  est très petit), ou si les  $e_j$  eux-mêmes sont petits. Le hessien dans ce cas devient tout simplement :

$$\nabla^2 f(\omega) = J^T(\omega) J(\omega) \quad (\text{A 3.7})$$

Qui est le même pour le cas linéaire.

Il est important de noter que l'équation (A3.7) est valide uniquement si les  $e_j$  sont petits. Un problème avec des grandes valeurs de  $e_j$  peut ne pas être résolu en utilisant l'approximation quadratique, et par conséquent, la performance de l'algorithme devient faible dans de tels cas [23].

Posant maintenant :

$$\nabla f(\omega) = 0 \quad (\text{A3.8})$$

$$\nabla f(\omega_0) + \Delta\omega \nabla^2 f(\omega_0) = 0 \quad (\text{A3.9})$$

Avec : 
$$\Delta\omega = (\omega - \omega_0) \quad (\text{A3.10})$$

$$\Delta\omega = -[\nabla^2 f(\omega_0)]^{-1} \nabla f(\omega_0) \quad (\text{A3.11})$$

En tenant compte des équations (A3.5), (A3.7), et (A3.14), la méthode de **Gauss-Newton** pour la mise à jour des coefficients, est donnée par l'équation :

$$\Delta\omega = [J^T(\omega) J(\omega)]^{-1} J^T(\omega) e(\omega) \quad (\text{A3.12})$$

La modification de **Levenberg-Marquardt** à la méthode de **Gauss-Newton** est la suivante :

$$\Delta\omega = [J^T(\omega) J(\omega) - \lambda I]^{-1} J^T(\omega) e(\omega) \quad (\text{A3.13})$$

Où  $\lambda$  est le pas d'adaptation, notant que pour  $\lambda$  grande l'algorithme devient une descente de gradient (avec un pas  $1/\lambda$ ), tandis que pour un petit  $\lambda$  le Levenberg-Marquardt devient un Gauss-Newton [24].

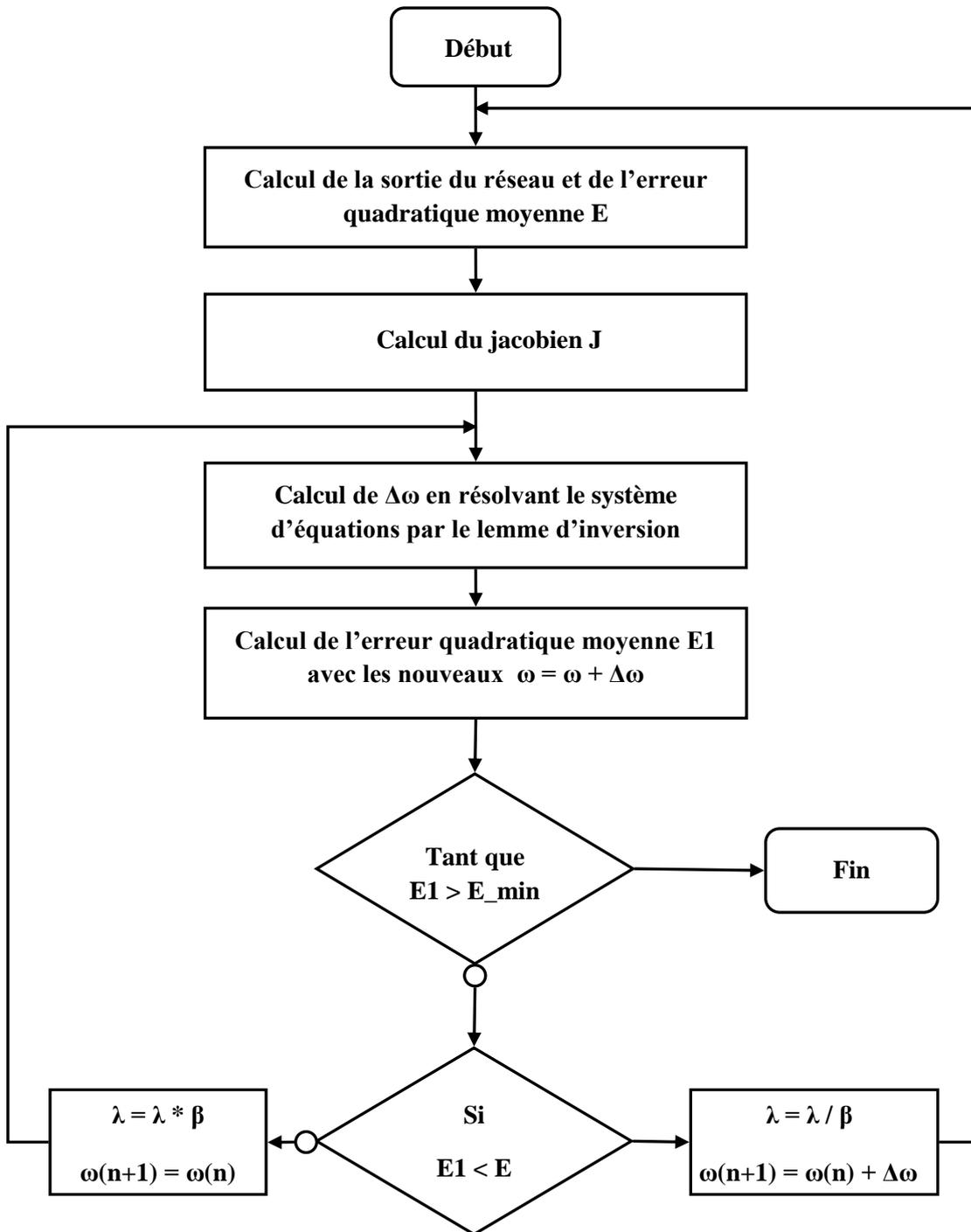
La méthode de Levenberg-Marquardt combine la rapidité de Gauss-Newton et la stabilité de l'algorithme de descente du gradient.

En tenant compte de l'équation (A3.3), le jacobien est donné par :

$$J = \begin{pmatrix} \frac{\partial e_1}{\partial \omega_1} & \dots & \frac{\partial e_1}{\partial \omega_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_m}{\partial \omega_1} & \dots & \frac{\partial e_m}{\partial \omega_n} \end{pmatrix} = - \begin{pmatrix} \frac{\partial y_1}{\partial \omega_1} & \dots & \frac{\partial y_1}{\partial \omega_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial \omega_1} & \dots & \frac{\partial y_m}{\partial \omega_n} \end{pmatrix} \quad (\text{A3.14})$$

### Un résumé de l'algorithme de Levenberg-Marquardt [24]:

1. Présenter toutes les entrées au réseau et calculer les sorties et les erreurs  $e_j(\omega) = (t_j - y_j(\omega))$  correspondantes, et calculer la somme des carrés des erreurs  $f(\omega)$  sur toutes les entrées.
2. Calculer le jacobien  $J$ .
3. Résoudre l'équation (10), (En utilisant le lemme d'inversion d'une matrice).
4. Recalculer  $f(\omega)$  avec les nouveaux coefficients  $\omega = \omega + \Delta\omega$ .
  - Si la nouvelle  $f(\omega)$  est plus petite que celle calculée dans l'étape 1, alors, on réduit  $\lambda$  par  $\beta$ , les coefficients restent  $\omega = \omega + \Delta\omega$ , et on retourne à l'étape 1.
  - Sinon, on jette les nouveaux coefficients ( $\omega = \omega + \Delta\omega$ ) et on garde ceux de l'étape 1 (avant l'actualisation), on augmente  $\lambda$  par  $\beta$ , et on retourne à l'étape 3.
5. L'algorithme est supposé être convergé lorsque la norme du gradient (A3.5) est inférieure à une valeur prédéterminée, ou lorsque la somme des carrés  $f(\omega)$  est réduite à une valeur cible ; il est également conseillé de limiter le nombre d'itérations concernant le critère d'arrêt afin d'éviter la boucle infinie.



**Figure A3.1** Organigramme de l'algorithme de Levenberg-Marquardt

# Annexe 4

## Apprentissage des réseaux RBF

L'apprentissage des réseaux RBF consiste à chercher les valeurs optimales des vecteurs,  $\mathbf{c}$  et  $\boldsymbol{\omega}$ , en utilisant les techniques suivantes :

- La technique du pseudo-inverse (technique off line)
- Apprentissage par algorithme de la descente du gradient (technique on line)
- Apprentissage hybride (technique on line)

### La technique du pseudo-inverse :

C'est une technique off line, c.-à-d. qu'on ne fait l'apprentissage que lorsqu'on termine le calcul de la sortie du réseau par rapport à tous les échantillons de l'ensemble d'entrée, supposant qu'on a un ensemble d'entrée de dimension (N x P), et on suppose une fonction radiale gaussienne, premièrement les centres sont choisis d'une manière aléatoire, et la largeur de la fonction gaussienne est déterminée par un choix ad hoc. L'apprentissage se fait comme suit :

On dispose de la matrice des sorties des neurones cachés :

$$\Phi = \begin{pmatrix} \phi_1^1 & \dots & \phi_M^1 \\ \vdots & \ddots & \vdots \\ \phi_1^P & \dots & \phi_M^P \end{pmatrix} \quad (\text{A 4.1})$$

Avec  $\phi_i^j$  : est la sortie du  $i^{\text{ème}}$  neurone caché, pour le  $j^{\text{ème}}$  échantillon d'entrée.

Et des vecteurs de sortie désirée et des poids de la couche de sortie :

$$\mathbf{Y}^d = [y_1^d \dots y_P^d] \quad (\text{A 4.2})$$

$$\boldsymbol{\Omega} = [\omega_1 \dots \omega_M]^T \quad (\text{A 4.3})$$

$\mathbf{Y}^d$  : Le vecteur de sortie désirée.

$\omega_i$  : Les poids entre la couche cachée et la sortie du réseau.

La sortie désirée est donné par :

$$\Phi \Omega = \mathbf{Y}^d \quad (\text{A 4.4})$$

$$\Phi^T \Phi \Omega = \Phi^T \mathbf{Y}^d \quad (\text{A 4.5})$$

$$\Omega = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{Y}^d \quad (\text{A 4.6})$$

$(\Phi^T \Phi)^{-1} \Phi^T$  : est le pseudo inverse de  $\Phi$ .

La solution existe uniquement si  $(\Phi^T \Phi)^{-1}$  existe.

### Apprentissage par descente du gradient :

L'un des approches les plus populaires pour l'actualisation des poids et des centres d'un réseau RBF, est la technique de descente du gradient qui est une technique supervisée. Etant donné la fonction de coût suivante :

$$E = \frac{1}{2} (y^d - y)^2 \quad (\text{A 4.7})$$

Les lois de mise à jour des centres et des poids sont données par :

$$c_{nm}(t+1) = c_{nm}(t) - \eta_1 \frac{\partial E}{\partial c_{nm}} \quad (\text{A 4.8})$$

Avec :  $n = 1 \dots N$  et  $m = 1 \dots M$

$$\omega_{mk}(t+1) = \omega_{mk}(t) - \eta_2 \frac{\partial E}{\partial \omega_{mk}} \quad (\text{A 4.9})$$

Avec :  $k = 1 \dots K$  (ici  $K=1$ )

Après simplification (voir chapitre 4 § 4.3.4.1), les lois d'apprentissage deviennent :

$$\omega_m(t+1) = \omega_m(t) + \eta_2 (y^d - y) \phi_m \quad (\text{A 4.10})$$

$$c_{nm}(t+1) = c_{nm}(t) + \eta_1 (y^d - y) \omega_m \frac{\phi_m}{\sigma^2} (x_n - c_{nm}(t)) \quad (\text{A 4.11})$$

## Apprentissage hybride :

L'apprentissage hybride est ainsi nommé parce qu'il regroupe de mode d'apprentissage, l'un non supervisé pour les centres, et l'autre supervisé pour les poids, l'algorithme *k-means* est l'un des algorithmes utilisés pour l'ajustement des centres, tandis que pour les poids, on utilise un algorithme tel que la descente du gradient.

### Algorithme k-means

L'algorithme des (*k-means*) est une méthode d'entraînement non supervisée pour le classement des données. La version la plus couramment utilisée est la classification par *k-means* adaptatif basé sur la distance euclidienne. Elle consiste à diviser l'espace d'entrée en  $k$  classes comme suit :

- 1- Le nombre  $k$  (nombre de fonction RBF, ou neurones cachés), est le nombre de classes.
- 2- Initialiser les centres des fonctions de base radiale.
- 3- Calculer la distance euclidienne entre les centres de chaque fonction RBF et le vecteur

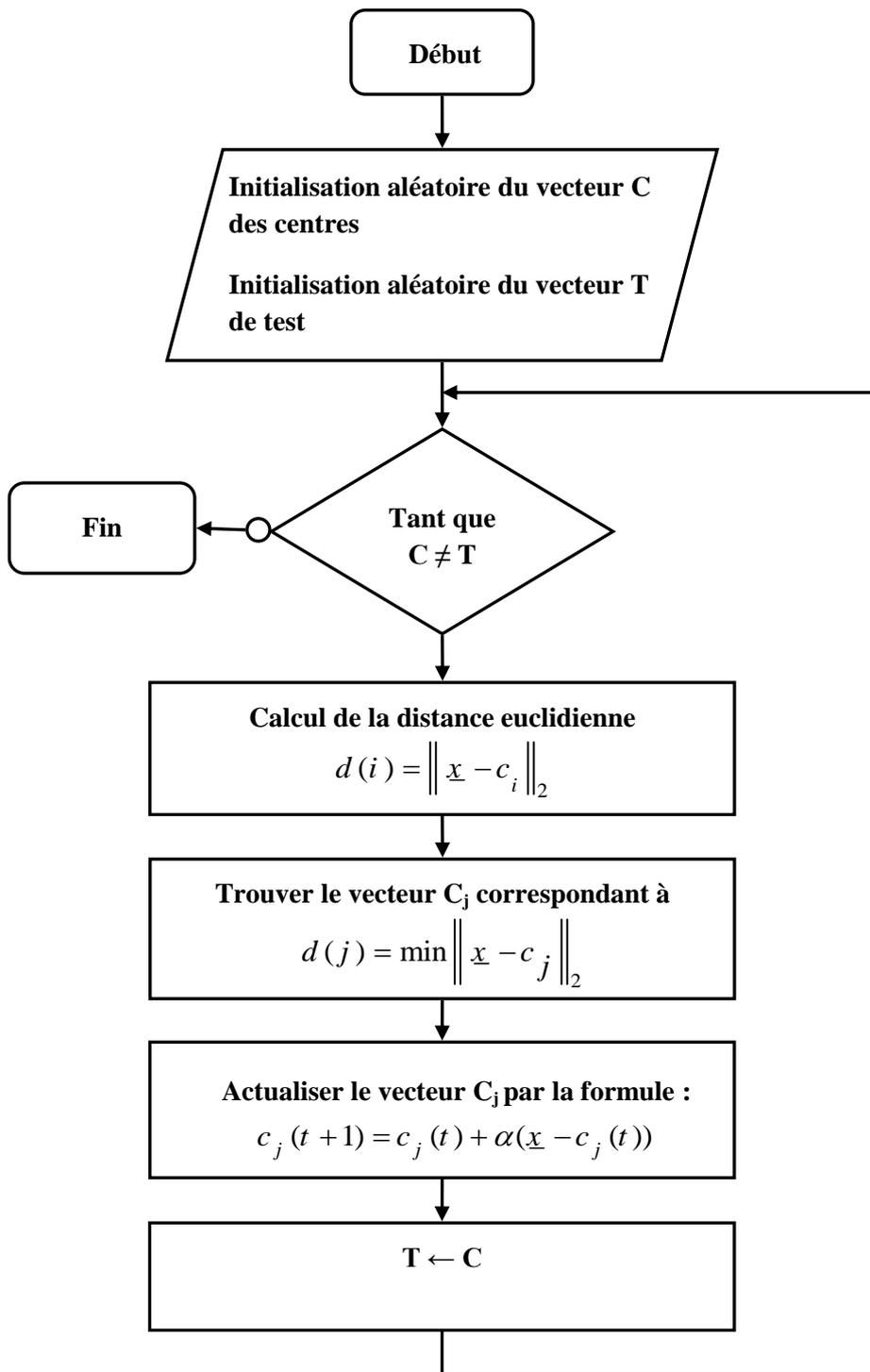
$$\text{d'entrée } \underline{x} \text{ i.e. } d(i) = \left\| \underline{x} - c_i \right\|_2, \quad i = 1 \dots N$$

- 4- Puis, ajuster le vecteur des centres  $c_i$  qui correspond à la distance minimal

$$d(j) = \min \left\| \underline{x} - c_j \right\|_2 \text{ en utilisant la loi d'adaptation suivante :}$$

$$c_j(t+1) = c_j(t) + \alpha(\underline{x} - c_j(t)) \quad (A 4.12)$$

Où  $j$  est l'indice du vecteur des centres le plus proche du vecteur d'entrées  $\underline{x}(t)$  (en d'autre terme,  $j$  est l'indice de la fonction de base radiale qui correspond à la distance euclidienne minimale). Le taux d'adaptation  $\alpha$  est un gain appartenant à l'intervalle  $[0 \ 1]$ , et peut être sélectionné par plusieurs méthodes.



**Figure A4.1** Organigramme de l’algorithme k-means

# Annexe 5

## Lemme d'inversion d'une matrice

**Théorème :** soit une matrice régulière écrite sous la forme  $A+BCD$  où  $A$  et  $C$  sont régulières, alors :

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B [C^{-1} + DA^{-1}B]^{-1} DA^{-1} \quad (A5.1)$$

Cette formule est désignée parfois sous le nom de formule de Sherman-Morrison-Woodburg ou plus communément sous le nom de lemme d'inversion matricielle.

Si la décomposition est judicieuse, elle permet de calculer l'inverse d'une matrice à l'aide d'une matrice plus facilement calculable ( $A^{-1}$ ) et de l'inverse d'une matrice éventuellement plus petite ( $C^{-1} + DA^{-1}B$ ).

**Démonstration :** en reprenant l'expression donnée dans le lemme, on a :

$$\begin{aligned} (A + BCD)(A + BCD)^{-1} &= (A + BCD) \left( A^{-1} - A^{-1}B [C^{-1} + DA^{-1}B]^{-1} DA^{-1} \right) \\ &= I - B [C^{-1} + DA^{-1}B]^{-1} DA^{-1} + BCDA^{-1} - BCDA^{-1}B [C^{-1} + DA^{-1}B] DA^{-1} \\ &= I - B \left( [C^{-1} + DA^{-1}B]^{-1} - C + CDA^{-1}B [C^{-1} + DA^{-1}B]^{-1} \right) DA^{-1} \\ &= I - B \left( [I + CDA^{-1}B] [C^{-1} + DA^{-1}B]^{-1} - C \right) DA^{-1} \\ &= I - B \left( C [C^{-1} + DA^{-1}B] [C^{-1} + DA^{-1}B]^{-1} - C \right) DA^{-1} \\ &= I \end{aligned}$$

**Remarque :** Si  $B$  et  $D$  sont respectivement des matrices colonne et ligne, et  $C$  un scalaire non nul, on obtient :

$$(A + BCD)^{-1} = A^{-1} - \frac{C}{1 + CDA^{-1}B} A^{-1}BDA^{-1} \quad (A5.2)$$