

**UNIVERSITE SAAD DAHLEB DE BLIDA**

**Faculté des sciences de l'ingénieur**

Département d'électronique

# **MEMOIRE DE MAGISTER**

Spécialité : Signaux et systèmes

## **APPLICATION DES RESEAUX NEUROGLIALS AUX SYSTEMES SINGULIEREMENTS PERTURBES**

Par

**SALAH Samia**

Devant le jury composé de :

Mr. M. BOUNEKHLA	Maitre de conférences, U. de Blida	Président
Mr. M. HADJ SADOK	Maitre de conférences, U. de Blida	Examineur
Mr. M. OULD ZMIRLI	Maitre de conférences, U. de Médéa	Examineur
Mr. A. GUESSOUM	Professeur, U. de Blida	Rapporteur
Mr. H. SALHI	Maitre de conférences, U. de Blida	Co-rapporteur

Blida, Juin 2009



## RESUME

Ce mémoire, a pour objet l'application des réseaux Neuroglials aux systèmes singulièrement perturbés. Nous proposons, une nouvelle architecture qui émule une réalité biologique découverte récemment.

La validation est effectuée sur le modèle de la machine asynchrone mise sous la forme standard singulièrement perturbée. Nous avons appliqué l'approche géométrique basée sur les cercles de Gerschgorin pour le découplage des variables lentes et rapides, ainsi que la méthode des perturbations singulières pour le développement des modèles réduits.

Enfin, une étude comparative des performances, entre le nouveau modèle Neuroglial\_MAS avec deux algorithmes différents, celle du modèle à réseau de neurones unique et du modèle à deux réseaux de neurones, est présentée.

Mots clés : Réseau de Neurone, Réseau Neuroglial, Machine Asynchrone, Méthode des perturbations singulières, Cercles de Gerschgorin, Neuroglial\_MAS.

## **ABSTRACT**

This thesis has for subject the application of the neuroglial networks on singular perturbed systems. We propose thus, a new architecture which emulates a biological existence discovered recently.

The validation is made on the models of the asynchronous machine putting in standard singularly perturbed form. We applied the geometrical approach based on the Gerschgorin's circles for decoupling of the slow and fast variables, as well as the singular perturbations method for the development of the reduction models.

Finally, a comparative study of the performances between the new Neuroglial\_MAS model with two different algorithms and that of the model with single network and of the model with two neural networks is presented.

**Key words:** Neural Network, Neuroglial Network, Asynchronous Machine, Singular Perturbations method, Gerschgorin Circles, Neuroglial\_MAS Network.



## REMERCIEMENT

Je tiens, en premier lieu à remercier le Seigneur, Dieu Tout Puissant pour sa bonté, son obligeance car ces travaux ne peuvent pas être réalisés sans son aide.

Je voudrais exprimer toute ma reconnaissance à Monsieur Abderrezak GUESSOUM, professeur à l'université Saad Dahlab de BLIDA, qui a encadré mes recherches et qui s'est toujours soucié de m'offrir, de tout point de vue et qui en dirigeant ce travail m'a fait profiter de ses connaissances.

J'adresse mes vivement remerciements à Monsieur Hassen SALHI, maître de conférence à l'université Saad Dahlab de BLIDA, pour son encadrement, ses remarques pertinentes et constructives et pour ses documentations intéressantes.

J'exprime ma profonde reconnaissance à Monsieur Omar TOUHAMI professeur à l'ENP pour son aide déterminante, pour ses documentations intéressantes, sa disponibilité et sa générosité.

Je remercie très vivement Monsieur M'hamed BOUNEKHLA, Maître de conférences à l'université Saad Dahleb de BLIDA, Monsieur M'hamed HADJ SADOK Maitre de conférences à l'université Saad Dahlab de BLIDA et Monsieur Mohamed OULD ZMIRLI Maitre de conférences à l'université de Médéa, pour avoir examiné ce travail et pour l'intérêt qu'ils ont porté à ce mémoire en me faisant l'honneur de participer à ce jury.

Merci à tous mes amis (es) qui m'ont accompagné pendant ces années. Merci à tous pour votre aide, votre soutien, votre sollicitude et surtout pour votre amitié.

Je remercie également très fortement pour leur soutien, leur patience et leur amour tous mes proches : ma très chère mère, mes frères, mes sœurs et toute ma famille.

**A la mémoire de Mon très cher père**



## LISTE DES ILLUSTRATIONS GRAPHIQUES ET TABLEAUX

Figure 1.1	Cellules gliales	16
Figure 1.2	Astrocyte	18
Figure 1.3	La communication entre les neurones et les astrocytes	20
Figure 2.1	Neurone biologique	24
Figure 2.2	Modèle mathématique	26
Figure 2.3	Fonction d'activation binaire à seuil	27
Figure 2.4	Fonction d'activation à rampe avec saturation	28
Figure 2.5	Fonction d'activation sigmoïde	28
Figure 2.6	Réseau dynamique et statique	30
Figure 2.7	Réseau de neurone multicouches	31
Figure 2.8	Réseau de neurone monocouche	32
Figure 3.1	Forme globale issue de l'équation (3.12)	50
Figure 3.2	Schéma découplé issue des équations (3.14) (3.15)	50
Figure 3.3	Cercle de Gerschgorin du modèle (3.19)	53
Figure 3.4	Cercles de Gerschgorin du modèle (3.29)	57
Figure 3.5	Cercles de Gerschgorin du modèle (3.30)	58
Figure 3.6	Cercles de Gerschgorin du modèle (3.36)	61
Figure 4.1	Exemple de décomposition horizontale	67
Figure 4.2	Exemple de décomposition verticale	68
Figure 4.3	Architecture séquentielle	69
Figure 4.4	Architecture parallèle	70
Figure 4.5	Architecture hiérarchique	71
Figure 4.6	Architecture Neurogliale	74
Figure 4.7	Un réseau expert	79
Figure 4.8	Réseau glial	80
Figure 5.1	Tracé des cercles de Gerschgorin pour la machine asynchrone pour différentes vitesses	92
Figure 5.2	Evolution des couples électromagnétiques des modèles réduits de la machine asynchrone pour $\Phi_{10} = [0 \ 0 \ 0 \ 0]$	98

Figure 5.3	Evolution des couples électromagnétiques des modèles réduits de la machine asynchrone pour $\Phi_{10} = [0 \ 0.9 \ 0 \ 0]$	100
Figure 5.4	Erreur d'apprentissage après 5000 itérations et 400 exemples	103
Figure 5.5	Evolution d'EQM lors de l'entraînement du réseau MLP	103
Figure 5.6	Réponse estimée et désirée du modèle à réseau de neurones unique MLP	104
Figure 5.7	Architecture modulaire à deux réseaux de neurones	105
Figure 5.8	Erreurs d'apprentissage après 100 itérations et 400 exemples	107
Figure 5.9	Evolution d'EQM du réseau lent	107
Figure 5.10	La réponse estimée et désirée lente après 100 itérations et de 400 exemples	108
Figure 5.11	Erreur d'apprentissage après 100 itérations et de 400 exemples	109
Figure 5.12	Evolution d'EQM du réseau rapide	110
Figure 5.13	La réponse estimée et désirée rapide après 100 itérations et de 400 exemples	110
Figure 5.14	La réponse globale	114
Figure 5.15	Erreur d'apprentissage du réseau expert1 après 100 itérations et de 100 exemples	115
Figure 5.16	Evolution d'EQM du réseau expert1 après 100 itérations	115
Figure 5.17	La réponse estimée et désirée rapide du réseau expert1 après 100 itérations et avec 100 exemples	116
Figure 5.18	Erreur d'apprentissage du réseau expert2 après 100 itérations et avec 100 exemples	116
Figure 5.19	Evolution d'EQM du réseau expert2 après 100 itérations	117
Figure 5.20	La réponse estimée et désirée rapide du réseau expert2 après 100 itérations et avec 100 exemples	117
Figure 5.21	Erreur d'apprentissage du réseau expert3 après 100 itérations et avec 100 exemples	118
Figure 5.22	Evolution d'EQM du réseau expert3 après 100 itérations	118
Figure 5.23	La réponse estimée et désirée rapide du réseau expert3 après 100 itérations et avec 100 exemples	119
Figure 5.24	Erreur d'apprentissage du réseau expert4 après 100 itérations et avec 100 exemples	119
Figure 5.25	Evolution d'EQM du réseau expert4 après 100 itérations	120

Figure 5.26	La réponse estimée et désirée rapide du réseau expert4 après 100 itérations et avec 100 exemples	120
Figure 5.27	Erreur d'apprentissage des quatre réseaux experts de l'estimation de la 1 <sup>ère</sup> partie de la réponse rapide après 100 itérations et de 100 exemples	122
Figure 5.28	Evolution des EQM des quatre réseaux experts de l'estimation de la 1 <sup>ère</sup> partie de la réponse rapide après 100 itérations.	123
Figure 5.29	Erreur d'apprentissage des quatre réseaux experts de l'estimation de la 2 <sup>ème</sup> partie de la réponse rapide après 100 itérations et de 100 exemples	123
Figure 5.30	Evolution des EQM des quatre réseaux experts du 2 <sup>ème</sup> partie de la réponse rapide après 100 itérations et 100 exemples.	124
Figure 5.31	Erreur d'apprentissage des quatre réseaux experts de l'estimation de la 3 <sup>ème</sup> partie de la réponse rapide après 100 itérations et de 100 exemples	125
Figure 5.32	Evolution des EQM des quatre réseaux experts de l'estimation de la 3 <sup>ème</sup> partie de la réponse rapide après 100 itérations de avec 100 exemples	125
Figure 5.33	Erreur d'apprentissage des quatre réseaux experts de l'estimation de la de 4 <sup>ème</sup> partie de la réponse rapide après 100 itérations et de 100 exemples	126
Figure 5.34	Evolution des EQM des quatre réseaux experts de la 4 <sup>ème</sup> partie de la réponse rapide après 100 itérations et de 100 exemples	126
Figure 5.35	La réponse rapide globale estimé par 2 <sup>ème</sup> algorithme	127
Figure 5.36	La réponse lente du réseau glial estimée par 2 <sup>ème</sup> algorithme	127
Figure 5.37	La réponse globale du réseau Neuroglial estimée par 2 <sup>ème</sup> algorithme	127
Tableau 5.1	Les performances du réseau unique MLP	104
Tableau 5.2	Performances des quatre réseaux experts, réseau rapide global, réseau global avec 100 itérations	121
Tableau 5.3	Pondérations et performances des réseaux experts pour chaque domaine de validité	129



## TABLE DES MATIERES

RESUME.....	
REMERCIEMENT.....	
TABLE DES MATIERES.....	
LISTE DES ILLUSTRATIONS GRAPHIQUES ET TABLEAUX.....	
INTRODUCTION .....	11
1. CELLULES GLIALES	
1.1 Introduction.....	15
1.2 Cellules gliales biologiques.....	16
1.2.1 Classification.....	16
1.2.1.1 Les cellules microgliales.....	16
1.2.1.2 Les cellules macrogliales.....	17
1.2.1.3 Les oligodendrocytes.....	17
1.2.1.4 Les astrocytes .....	17
1.2.2 Le processus de fonctionnement des astrocytes.....	18
1.2.3 La communication entre les neurones et les astrocytes.....	20
1.3 Conclusion.....	22
2. RESEAUX DE NEURONES ARTIFICIELS.....	23
2.1 Introduction.....	23
2.2 Le neurone biologique.....	24
2.2.1 Fonctionnement du neurone biologique.....	25
2.3 Modèle mathématique générale du neurone.....	26
2.3.1 Fonctions d'activations.....	27
2.4 Réseau de neurone.....	28
2.4.1 Architectures des réseaux de neurones.....	29
2.4.2 Apprentissages des réseaux de neurones.....	32
2.4.2.1 Les différents types d'apprentissages .....	33
2.4.2.2 La procédure d'apprentissage.....	34
2.4.2.3 Les règles d'apprentissage.....	34

2.4.2.4 Propriétés des réseaux de neurones .....	38
2.4.2.5 Les avantages des réseaux de neurones.....	39
2.4.2.6 Les inconvénients des réseaux de neurones.....	39
2.4.2.7 Les applications des réseaux de neurones....	39
2.5 Conclusion.....	40
3 METHODES DES PERTURBATIONS SINGULIERES ET LES CERCLES DE GERSCHGORIN.....	41
3.1 Introduction .....	41
3.2 Méthode des perturbations singulières.....	42
3.3 Méthode des perturbations singulières appliquée aux systèmes non-linéaires .....	43
3.4 Méthode des perturbations singulières appliquée aux modèles continus linéaires.	43
3.4.1 Calcul des modèles réduits lent et rapide.....	47
3.4.2 Application .....	49
3.4.3 Interprétation.....	49
3.5 Identification géométrique des dynamiques.....	51
3.5.1 Cercles de Gerschgorin.....	51
3.5.2 Application.....	53
3.5.3 Transformations modales.....	54
3.5.3.1 Changement de taille des rayons du cercles.....	55
3.5.3.2 Déplacement des centres des cercles.....	55
3.5.3.3 Application.....	57
3.5.4 Regroupement et séparation des systèmes des dynamiques linéaires.....	59
3.5.5 Application.....	60
3.6 Conclusion.....	63
4 ARCHITECTURE NEUROGLIALE.....	64
4.1 Introduction.....	64
4.2 Le réseau artificiel Neuroglial.....	65
4.2.1 La décomposition de la tâche en sous- tâche .....	66
4.2.2 L'organisation de l'architecture modulaire .....	69
4.2.3 La Nature de l'apprentissage.....	71
4.2.4 La communication entre les modules.....	72
4.3 L'architecture Neurogliale.....	73

4.4	Apprentissage du réseau Neuroglial.....	76
4.4.1	Apprentissage des experts .....	79
4.4.2	Apprentissage du réseau glial.....	80
4.5	Conclusion.....	85
5	Application du réseau Neuroglial au modèle de la machine asynchrone et simulation	86
5.1	Introduction.....	86
5.2	Mise sous forme standard de la machine asynchrone .....	87
5.2.1	Les modes électroniques .....	87
5.2.2	Modèle d'état des flux de la machine en $\alpha, \beta$ .....	89
5.2.3	Séparation des modes électromagnétiques.....	89
5.2.4	Mise sous la forme standard singulièrement perturbée.....	93
5.2.4.1	Modèles réduits de la machine asynchrone en repère $(\alpha, \beta)$ .....	93
5.2.5	Modèles réduits en flux et en courants .....	94
5.2.6	Simulations et interprétations.....	96
5.3	Simulation.....	101
5.3.1	Réseau unique.....	101
5.3.2	Apprentissage modulaire.....	105
5.3.2.1	Apprentissage du réseau lent.....	106
5.3.2.2	L'apprentissage du réseau rapide.....	108
5.3.2.3	Combinaison et comparaison.. ..	111
5.3.3	Réseau Neuroglial .....	112
5.3.3.1	Architecture.....	113
5.3.3.2	1 <sup>er</sup> algorithme d'apprentissage.....	114
5.3.3.3	2 <sup>ème</sup> algorithme d'apprentissage.....	121
5.3.4	Comparaison entre les différentes architectures.....	130
5.4	Conclusion.....	131
	CONCLUSION .....	132
	RECOMMANDATIONS.....	133
	APPENDICE.....	134
	REFERENCES	



## INTRODUCTION

Actuellement, les réseaux de neurones artificiels (RNA) sont d'intérêt de recherche majeur associant des chercheurs de différentes disciplines, telles que: la biologie, l'informatique, l'électronique, les mathématiques, la médecine, la physique et la psychologie.

La plupart des recherches sur les réseaux de neurones sont basées sur les questions du "comment peut-on faire la différence entre deux choses" ? Et " par quel prodige arrivons-nous tout à la fois à conduire, lire les panneaux et parler " ? Durant un siècle, la réponse à ces questions était associée à un seul mot: neurone ! Mais depuis quelques années, des découvertes récentes sur le fonctionnement du cerveau humain ont mis en évidence le rôle d'autres cellules cérébrales, les cellules gliales et parmi elles, les astrocytes. Ces derniers témoigneraient d'inédites dispositions pour la communication. A tel point que, grâce à eux, notre cerveau n'aurait pas une, mais deux façons à faire circuler l'information. De ce fait, sur la voie nerveuse, les informations circulent 100 000 fois plus vite que sur la voie astrocytaire [5]. Les travaux récents montrent aussi que les astrocytes communiquent également formant un réseau séparé parallèle au réseau neuronal. Les astrocytes ont les moyens de faciliter, de ralentir ou même de bloquer le flux d'informations [5].

Le présent travail développe une nouvelle architecture qui émule cette réalité biologique, appelée Neurogliale. Sa conception est basée sur le puissant concept de modularité. La mise en œuvre de l'architecture Neurogliale est basée sur les systèmes complexes à multi-échelles de temps, ce sont des systèmes singulièrement perturbés. Ces systèmes sont des modèles mathématiques non-linéaires, fortement couplés et présentent des dynamiques différentes ou d'ordre élevé. La modélisation, l'identification, l'analyse et le contrôle de tels systèmes par les réseaux de neurones uniques à une seule échelle de temps peuvent être difficiles car présentent les problèmes suivants :

- Temps d'apprentissage des réseaux de neurones très grand.

- Configuration optimale du réseau pour un problème donné vue le nombre de neurones et de couches augmentent selon la complexité du système.
- Convergence vers des minimas locaux.
- Convergence lente.

Ces limitations sont dues au fait que les modèles existants des réseaux de neurones ne reflètent pas réellement certains comportements des neurones dans le cerveau humain. Ils ne considèrent pas la participation d'autres éléments au traitement de l'information. Aussi, ils ne tiennent pas en compte du fait que l'information est traitée en deux échelles de temps différentes lente et rapide.

En essayant d'améliorer les modèles neuronaux existants, l'approche Neurogliale est celle qui émule le mieux le traitement de l'information dans le cerveau humain. Le réseau Neuroglial est capable d'apprendre assez rapidement avec un minimum d'exemples et dispose de meilleures performances pour résoudre des problèmes les plus complexes du monde réel, avec une complexité réduite et avec une quantité limitée du calcul.

En effet, pour l'application de cette approche, on a procédé à une phase de décomposition du système en sous-systèmes lent et rapide. La méthode de décomposition choisie est la méthode des perturbations singulières [11]. Cette méthode a pour principe de base de maintenir le type de modèle utilisé pour représenter le système global et de réduire l'ordre du modèle ou de découpler les variables pour aboutir à des sous modèles d'ordre réduit. La première difficulté qui apparaît lorsqu'on veut découpler les variables réside dans l'identification des variables lentes et rapides. Ceci est effectué par la méthode géométrique, les cercles de Gerschgorin [14][17], qui permet de localiser les valeurs propres sur le plan complexe dans des faisceaux de cercles: le regroupement des modes est immédiat dès que des cercles sont disjoints, et à posteriori, le nombre de modes lents et rapides est déterminé.

Notre contribution consiste à la validation de l'approche Neurogliale sur les modèles réduits lent et rapide développés par application de la méthode des perturbations singulières sur le modèle global de la machine asynchrone.

Le premier chapitre est consacré à expliciter le fonctionnement biologique des cellules gliales ainsi le rôle important de ces cellules pour le contrôle et la modulation des activités neuronales.

Nous allons ensuite aborder, dans le deuxième chapitre, la notion des réseaux neuronaux : notions générales, l'apprentissage supervisé des réseaux non bouclés.

Le troisième chapitre est dédié à la présentation de la méthode des perturbations singulières. Nous avons détaillé cette méthode pour les systèmes linéaires et non-linéaires. L'utilisation correcte de la méthode des perturbations singulières, pour la réduction des modèles singulièrement perturbés, requière l'identification et la séparation des différentes dynamiques du système. Pour ce faire nous avons présenté une approche géométrique basée sur les cercles de Gerschgorin qui répond à ce besoin. Cette méthode donne une bonne sélection des variables lentes et rapides.

Le quatrième chapitre traite de la question générale de l'introduction des dernières connaissances acquises sur les cellules gliales dans les réseaux de neurones, qui aboutit à des éléments de réponse dans les architectures neuronales modulaires. Différentes variantes de modularité seront présentées, ainsi que les notions d'apprentissage liées à celle-ci. En se basant sur les connaissances modulaires détaillées dans ce chapitre, une nouvelle approche Neurogliale est conçue et deux algorithmes d'apprentissage seront développés.

Les travaux présentés dans le cinquième chapitre font l'objet de deux grandes parties. La première partie concerne la mise sous forme standard singulièrement perturbée des modèles des machines asynchrones où est appliquée l'approche géométrique par les cercles de Gerschgorin pour le découplage des variables lentes et rapides ainsi que la méthode des perturbations singulières pour développer ces modèles. La deuxième partie inclut la validation et l'évaluation de l'approche Neurogliale sur les modèles réduits.

Afin de mettre en avant les avantages à utiliser la structure Neurogliale, nous avons réalisé en parallèle le même modèle avec un réseau de neurone unique MLP et avec une architecture modulaire à deux réseaux. Une étude comparative entre les

performances des deux algorithmes du réseau glial sera présentée. Une autre étude comparative est effectuée entre les performances du modèle à réseau de neurone unique et celle du modèle à deux réseaux de neurones. Ces performances sont exprimées en termes de vitesse de convergence, d'erreur minimale quadratique et en termes de complexité réduite de calcul.

Enfin, nous terminerons par une conclusion générale, situant ce présent travail, les résultats obtenus et les orientations futures possibles.

# CHAPITRE 1

## LES CELLULES GLIALES

### 1.1 Introduction

Pour pouvoir créer un RNA qui émule le cerveau nous devons comprendre le fonctionnement et la structure biologique de ce dernier.

Le cerveau, quartier général du système nerveux, contrôle les actions et les sensations de toutes les parties du corps, ainsi que la pensée, la logique et la mémoire. Il est constitué de matière grise, laquelle regroupe 10 milliards de cellules composant un immense réseau où se produisent des milliers de milliards d'interactions [1]. Ce réseau est constitué de cellules nerveuses : les neurones; Ceux-ci sont les cellules qui permettent au cerveau d'exercer ses fonctions. Elles sont très importantes car elles sont responsables du traitement et de la transmission des informations issues de notre environnement [1].

Outre les neurones, le système nerveux contient des cellules moins connues : Les cellules gliales (glie, d'origine grecque, signifie « colle »). Elles constituent le type cellulaire le plus abondant du cerveau : elles sont 10 fois plus nombreuses que les neurones. Longtemps considérées comme de simples « cellules de soutien », un support sur lequel évoluent les neurones, elles exercent un rôle beaucoup plus étendu [1][2][3]. Elles communiquent avec les neurones, elles les maternent et elles participent au contrôle de l'activité neuronale. On décrypte petit à petit le langage de ce type de communication intercellulaire complexe [1][2][3][4][5].

Les cellules gliales furent découvertes en 1904 par Santiago Ramón y Cajal. Le fait fut découvert au début du XX<sup>e</sup> siècle [2]. Ils ont d'abord été décrits comme de simples éléments de cohésion du tissu nerveux. Mais les recherches menées au cours des 20 dernières années (Haydon en 2001, Newman en 2003, Volterra et Meldolesi en 2005, Takano en 2006 et autres) montrent leur rôle primordial dans l'activité cérébrale en tant que partenaires essentiels des neurones [8].

Le but de ce chapitre est de présenter des notions générales sur les cellules gliales afin d'en préciser l'importance de leur rôle pour le contrôle et la modulation des activités neuronales.

## 1.2 Cellules gliales biologiques

Les cellules gliales assurent l'isolement des tissus nerveux, les fonctions métaboliques, le soutien squelettique et la protection vis à vis des corps étrangers en cas de lésions. De récents travaux ont montrés que certaines de ces cellules jouent un rôle actif dans la transmission de l'influx nerveux.

La glie détermine la formation des synapses et participe à la sélection des connexions neuronales (mémoire à long terme, plasticité du cerveau) [3].

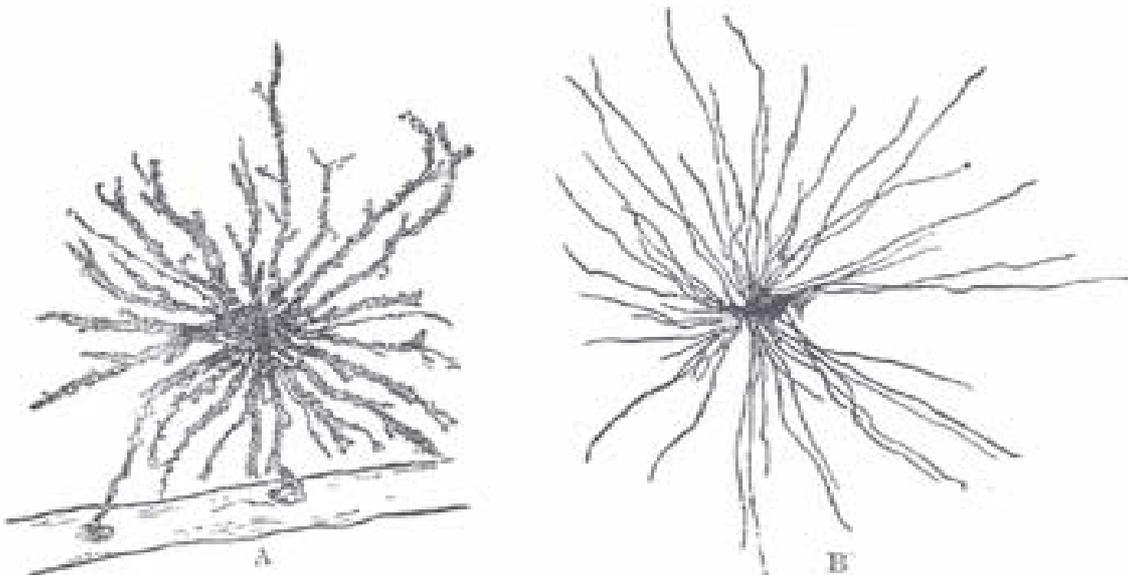


Figure 1.1: Cellules gliales [2]

### 1.2.1 Classification

Différents types de cellules gliales assurent, chacune à leur façon, le bon fonctionnement des neurones du système nerveux central.

#### 1.2.1.1 Les cellules microgliales

Ces cellules représentent 5% à 25% de toutes les cellules gliales du SNC. Elles sont différentes des cellules macrogliales, que ce soit au niveau de leur morphologie, de leur fonction ou de leur origine. Ces petites cellules, de forme variable et au noyau allongé, sont mobiles. Elles appartiennent au groupe des

macrophages [2][6], Les cellules microgliales constituent la première ligne de la défense contre les envahisseurs étrangers.

#### 1.2.1.2 Les cellules macrogliales

Ces cellules sont très nombreuses dans le système nerveux central, mais elle existe aussi dans le système nerveux périphérique, où ces cellules ont des fonctions analogues à certaines cellules gliales du système nerveux central [2][6].

#### 1.2.1.3 Les oligodendrocytes

Ces cellules constituent la gaine de myéline qui entoure les axones de nombreux neurones. Le motif très particulier de cet enroulement accélère la conduction nerveuse [2][6].

Des noms différents sont donnés aux cellules gliales qui assument ces fonctions dans le système nerveux périphérique. On parle alors de cellules satellites pour le support mécanique des neurones et de cellules de Schwann pour la fabrication de la myéline [2][6].

Mais de toutes les cellules gliales, l'astrocyte est sans doute celui qui a les fonctions les plus complexes [2][4].

#### 1.2.1.4 Les astrocytes [2][4][6]

Les astrocytes sont des cellules gliales de forme étoilée que l'on trouve généralement dans le cerveau et plus généralement dans le système nerveux central. On distingue deux grands types d'astrocytes:

- Astrocyte de type I : Les astrocytes de type I sont en contact avec les capillaires sanguins grâce à leurs pieds astrocytaires. Ils jouent un rôle actif dans le métabolisme et l'alimentation en glucose des neurones. La sécrétion de glutamate par le neurone "active" l'action astrocytaire de pompage du glucose.
- Astrocyte de type II : Les astrocytes de type II entourent le neurone et la fente synaptique. Le taux de couverture peut varier de 1 à 100% suivant le type de

neurone. Ils jouent un rôle dans la propagation du signal nerveux en agissant sur la dispersion des neurotransmetteurs. De plus les astrocytes agissent sur la capture des différents neurotransmetteurs, influant ainsi sur l'intensité d'un signal et sa durée.

Le corps cellulaire contient un noyau et un cytoplasme peu abondant. Les prolongements cytoplasmiques sont quant à eux très importants, parcourant tout le neuropile.

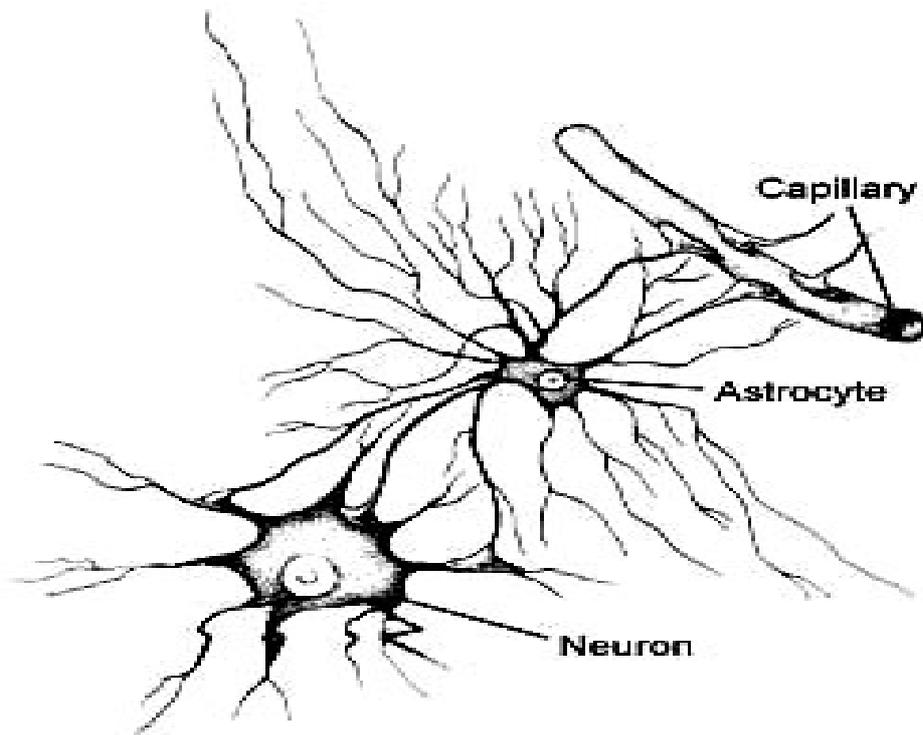


Figure 1.2 : Astrocyte [3]

### 1.2.2 Le processus de fonctionnement des astrocytes

Les neurologues s'interrogent depuis longtemps sur le rôle des cellules gliales qui remplissent l'espace interneuronal et constituent une part importante de la masse du cerveau. On a cru jusqu'à ces dernières années qu'elles avaient une utilité, d'ailleurs mal perçue, dans le soutien et l'alimentation des neurones. Mais aujourd'hui, diverses équipes s'intéressent directement à leur rôle dans la neurotransmission des informations. Ce sont en particulier certaines de ces cellules gliales, les astrocytes, qui retiennent l'attention [5] [4].

Ces derniers sont considérés comme les plus intéressants. Ils jouent un rôle dans la propagation du signal nerveux en agissant comme suit :

### 1-Les astrocytes interceptent les influx nerveux :

Dans le cerveau, des groupes d'astrocytes forment des ensembles de communication, appelés domaines astrocytaires, qui semblent travailler spécifiquement avec certains neurones, pour capter l'influx qui circule dans les axones. Les astrocytes se branchent sur les interconnexions neuronales (synaptiques). Lorsqu'ils détectent certains neuromédiateurs, ils s'activent et envoient un signal vers les autres astrocytes, qui s'activent à leur tour, propageant le signal [5].

### 2-puis ils envoient un message par voie chimique :

Le signal véhiculé par les astrocytes n'est pas électrique comme dans les neurones, mais chimique (augmentation du taux de calcium). Il se propage lentement d'astrocyte en astrocyte, jusqu'à l'extrémité d'un autre bras astrocytaire, qui va influencer sur la communication des neurones qu'il enserre [5].

### 3-Ils vont modifier l'influx nerveux :

Les astrocytes agiraient, soit après activation par un neurone, soit de leur propre « chef » comment ? En modifiant les messages entre neurones de trois manières [5] :

- Par « nettoyage » de l'espace synaptique en recyclant les neuromédiateurs pour fluidifier la communication.
- Par diffusion de pièges aux neuromédiateurs neutralisant la communication.
- Par multiplication des récepteurs des neuromédiateurs, amplifiant l'influx nerveux.

Donc les astrocytes ont des moyens de faciliter, de ralentir ou même de bloquer ce flux d'informations [5].

### 4- Ils contrôlent aussi l'alimentation des neurones :

Des bras astrocytaires pompent du glucose dans les vaisseaux sanguins. Ce sucre, qui sert de carburant biologique, est ensuite distribué aux neurones par les astrocytes qui semblent en maîtriser le flux [5].

### 1.2.3 La communication entre les neurones et les astrocytes

Les astrocytes libèrent des transmetteurs chimiques, en particulier, ils augmentent le calcium en eux-mêmes amenant la libération de glutamate, celui-ci étant le neurotransmetteur le plus abondant dans le système nerveux [4]. Les différentes expériences ont mis en évidence que le fonctionnement physiologique de la synapse n'est pas simplement lié au présynaptique, postsynaptique mais aussi aux astrocytes autour [4].

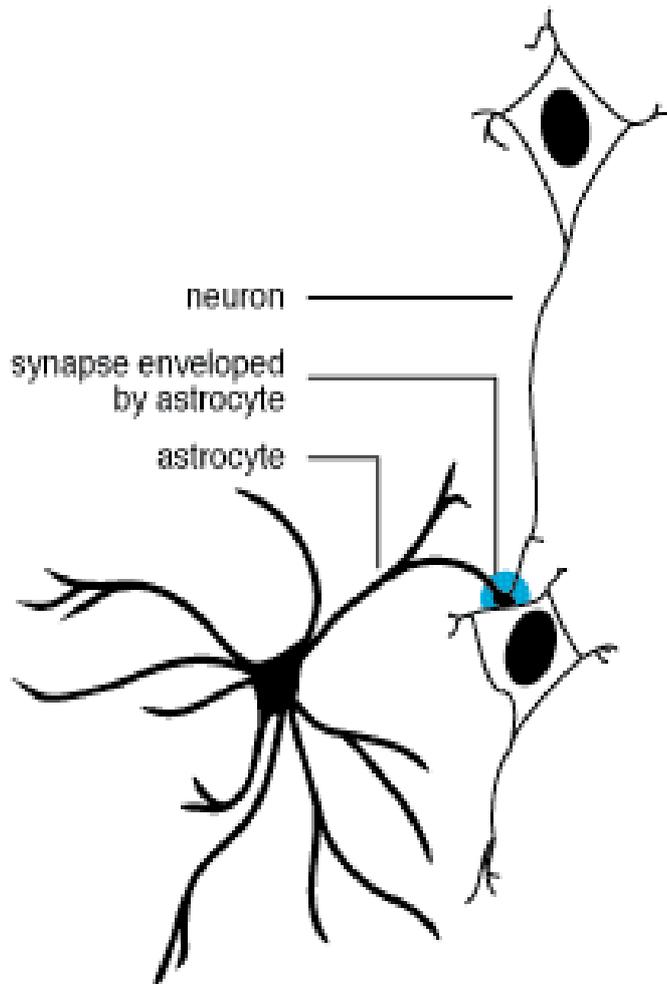


Figure 1.3 : La communication entre les neurones et les astrocytes [4]

Il est à noter que la principale différence entre les astrocytes et les neurones étant que de nombreux neurones étendent leurs axones sur de large distance et qu'ils propagent leur potentiel d'action en une très courte période à très haute vitesse contrairement aux astrocytes qui n'utilisent pas d'excitation électrique mais propagent de fortes concentration de calcium sur une longue durée (10 seconde, environ 6 fois que plus long) [7].

Proportion cellules gliales : neurone [7]

- Nematodes un micro organisme <1
- Rodents rongeur 1:1
- Cerveau humain environ 50:1

Les astrocytes peuvent moduler le seuil pour la plasticité synaptique et l'excitation neuronale par la libération de glutamate. Combiné avec le potentiel d'agir sur 140000 synapses chacun, les astrocytes ajouteraient une nouvelle couche nous permettant de nous aider à établir les différences de capacités du système nerveux entre les êtres humains [7].

Suite aux dernières avancées de la science, il est à noter une implication beaucoup plus importante des astrocytes dans la communication nerveuse notamment que les astrocytes se comportent comme un syncytium : un seul et même élément. Ce réseau permettrait la propagation d'ions par exemple le calcium dans un grand nombre de synapses en même temps. Ce réseau se superposerait au réseau neuronal jouant un rôle majeur de modulation dans les variations d'activités neuronales [7] [5].

Les astrocytes témoigneraient d'inédites dispositions pour la communication. A tel point que grâce à eux, notre cerveau n'aurait pas une mais deux façons de faire circuler l'information. De fait sur la voie nerveuse, les informations circulent 100.000 fois plus vite que sur la voie astrocytaire. Donc il y a deux réseaux de vitesse [5].

### 1.3 Conclusion

Les neurones ne seraient donc pas les seuls à échanger des informations: des messages seraient transmis d'un point à un autre du cerveau par une autre voie de communication. Les informations propagées par cette vague circulent très lentement. Elles atteignent au mieux des vitesses de l'ordre de 15 à 30 micromètres par seconde contre 1 mètre par seconde au minimum pour l'influx nerveux [5]. L'utilisation de ces deux réseaux, l'un rapide, l'autre lent, permettrait à notre cerveau de mieux coordonner son activité.

Ce qu'il faut retenir c'est que les réseaux neuronaux ne peuvent se constituer par simple auto-organisation locale comme on le croyait mais que le réseau de connexions doit être contrôlé et alimenté par un deuxième réseau agissant plus globalement et de façon coordonnée à distance. Il ne suffit pas qu'une connexion se fasse, il faut qu'on lui fournisse les ressources pour durer.

La prochaine partie montrera l'intégration dans les systèmes interconnectés de nouvelles connaissances qui émulent réellement le traitement de l'information dans le cerveau humain. Les modèles informatiques vont nous aider à mieux comprendre les causes et les facteurs qui sont impliqués dans le fonctionnement particulier des connexions du cerveau. Les systèmes interconnectés proposés sont appelés réseaux artificiels Neuroglials.

## **CHAPITRE 2 RESEAUX DE NEURONES ARTIFICIELS**

### 2.1 Introduction

Conçus à l'origine par des biologistes pour étudier le cerveau humain, les réseaux de neurones artificiels sont, de nos jours, utilisés dans plusieurs domaines. Derrière le vocable de réseaux de neurones, aussi appelés réseaux connexionnistes ou réseaux neuromimétiques, se cache une grande diversité d'architectures de réseaux et d'algorithmes d'apprentissage.

L'idée est de modéliser l'entité de base du cerveau humain, le neurone, puis d'en assembler plusieurs, formant ainsi un réseau de neurones. Celui-ci doit être capable de réaliser certaines des fonctions du cerveau humain comme la mémorisation, la généralisation et le travail en parallèle. Le réseau de neurones doit être aussi s'adapter à son environnement. Ceci est réalisé par sa capacité d'apprentissage à partir d'exemples. L'écart en performance entre les réseaux de neurones et le cerveau humain demeure toujours grand, vu la complexité de ce dernier. L'absence de normalisation ajoute aussi à la difficulté de présenter clairement une théorie sur les réseaux de neurones. La complexité et une connaissance toujours améliorée du cerveau ont amené une multitude de solutions pour la conception des réseaux de neurones.

Une problématique importante se situe au niveau de la configuration des réseaux de neurones dont la théorie est inexistante jusqu'à ce jour. C'est un domaine de recherche largement ouvert mais une nouvelle technique provenant des plans d'expériences permettrait toutefois d'optimiser la configuration d'un réseau de neurones selon le contexte du problème.

## 2.2 Le neurone biologique

Le cerveau humain, le plus étonnant des systèmes de traitement de l'information, est constitué d'environ 50-100 milliards d'unités appelées neurones [10]. Les neurones sont des cellules du système nerveux spécialisées dans la transmission de l'information.

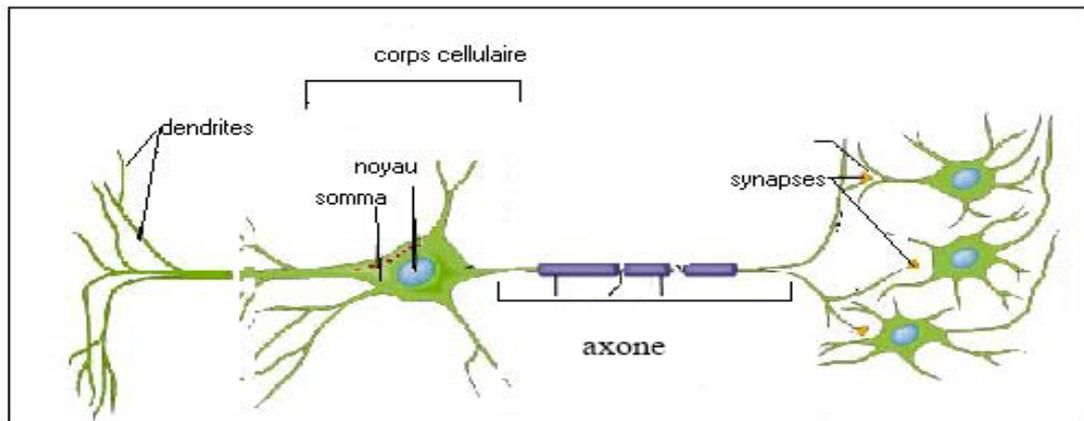


Figure 2.1 Neurone biologique [10]

Le neurone est une cellule constituée principalement de trois parties, qui ont un rôle bien défini. Ce sont : les dendrites, le soma et l'axone (figure 2.1).

### a) Dendrites :

Cette partie, qui présente une sorte de ramification appelée arbre dendritique, collecte des signaux venant d'autres cellules ou de l'extérieur. La réception des signaux en provenance des autres neurones se fait par des points de contacts appelés synapses (certaines cellules peuvent compter jusqu'à 100 000 synapses). Ces signaux, qui ne sont que des impulsions électriques, sont par la suite acheminés vers le corps cellulaire (le soma).

### b) Le Soma :

L'arbre dendritique fait converger vers le soma des influx nerveux d'une très large étendue autour du neurone. Le soma, outre son rôle concernant le métabolisme de la cellule, recueille et concentre les informations reçues et en fait une sommation dite "spatio – temporelle". Si le potentiel somatique dépasse un

certain seuil, il y a émission d'un potentiel d'action appelé "spike", qui correspond à une oscillation électrique très brève (1ms) [9].

c) L'Axone :

Le signal traité est transmis vers l'extérieur le long de l'axone et réparti sur les synapses des neurones cibles grâce à l'arborisation terminale que possède l'axone.

Un neurone possède exactement une sortie, correspondant au signal parcouru à travers l'axone; il peut recevoir plus de 10000 entrées au niveau de ses dendrites. L'essence du traitement de l'information du neurone est l'intégration de l'information reçue au niveau des dendrites et sa réduction en une seule quantité (sortie du neurone).

Le cerveau humain contient environ  $10^{11}$  neurones et chacun d'eux peut avoir  $10^4$  connexions synaptiques avec les autres neurones [9].

### 2.2.1 Fonctionnement du neurone biologique

L'information, sous forme de signaux électroniques et chimiques, parcourt le cerveau continuellement.

Dans le cerveau, le neurone émet un signal en fonction des signaux collectés en provenance d'autres neurones après leur traitement. Le traitement neuronal est une opération assez complexe, il s'agit en effet, d'une intégration spatio-temporelle [10] des signaux collectés, c'est-à-dire une sorte de sommation. Quand la somme dépasse un certain seuil, le neurone émet à son tour un signal à travers son axone auquel sont reliés d'autres neurones. La zone de contact entre l'axone et la dendrite est appelée synapse. C'est au niveau de cette synapse que s'effectue la transmission de l'information d'une cellule à une autre.

Les neurones communiquent en émettant des trains de potentiels rapides et très courts de l'ordre de quelques millisecondes. Le message est modulé en termes de fréquences qui peuvent varier de quelques hertz à quelques centaines de hertz.

Des considérations temporelles du même ordre montrent que la quantité d'information transmise d'un neurone à l'autre doit être très petite (quelques bits), et donc que l'information critique n'est pas transmise directement, mais qu'elle est contenue et répartie dans les interconnexions, d'où le nom de modèle connexionniste et mémoire associative pour décrire les RNA.

### 2.3 Modèle mathématique général du neurone

S'inspirant de leurs travaux sur le neurone biologique Mac culloch et Pitts, ont proposé le modèle représenté dans la (figure 2.2).

La plupart des algorithmes connexionnistes en ce moment s'inspirent du modèle de la (figure 2.2) dans lequel le neurone fait une somme pondérée des potentiels d'actions qui lui parviennent (Chacun de ces potentiels est une valeur numérique qui représente l'état du neurone qui l'a émis), puis s'active suivant la valeur de cette sommation pondérée. Si cette somme dépasse un certain seuil, le neurone est activé et transmet une réponse (sous forme de potentiel d'action) dont la valeur est celle de son activation. Si le neurone n'est pas activé, il ne transmet rien.

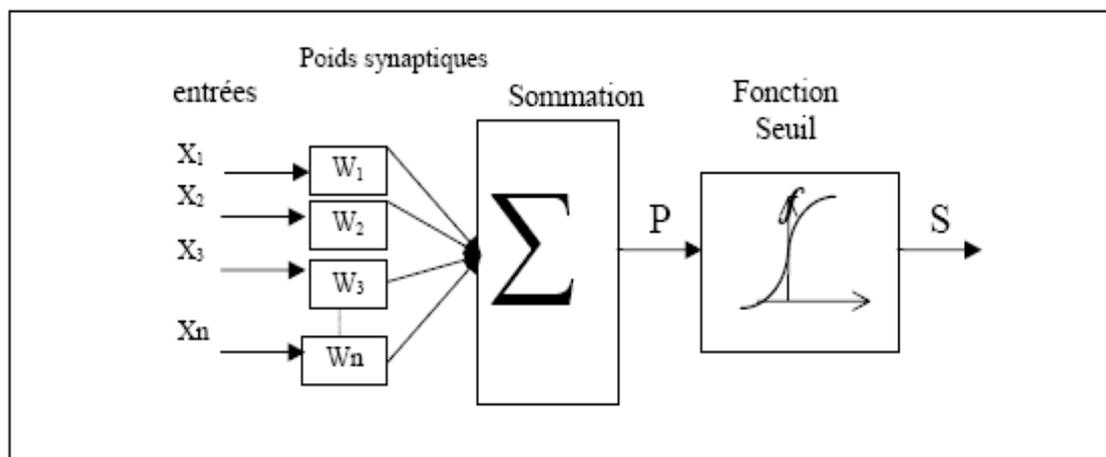


Figure 2.2 : Modèle mathématique

D'une façon plus générale, on peut définir un neurone formel par les cinq éléments suivants :

- 1-La nature de ses entrées qui peuvent être binaires ou réelles.
- 2-La fonction d'entée totale P qui définit le prétraitement effectué sur les entrées.

avec 
$$P = \sum_{i=1}^n X_i W_i \quad (2.1)$$

ou  $X_i$  : représente les entrées des neurones.

$W_i$  : représente les poids des neurones.

3- La fonction d'activation (ou d'état) du neurone qui définit son état interne en fonction de son entrée totale, elle peut être : une fonction binaire à seuil, une fonction linéaire à seuil, une fonction gaussienne, la fonction de Heaviside, la fonction sigmoïde la fonction signe.

4- La fonction de sortie qui calcule la sortie du neurone en fonction de son état d'activation.

$$S = f(p) \quad (2.2)$$

5- La nature de la sortie du neurone qui peut être binaire ou réelle.

### 2.3.1 Fonctions d'activations

a)- Fonction binaire à seuil

Le seuil introduit une non-linéarité dans le comportement du neurone, c'est le modèle tout ou rien.

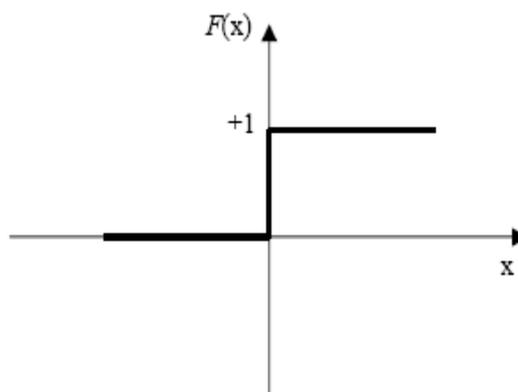


Figure 2.3 : Fonction d'activation binaire à seuil.

b)- Fonction à rampe avec saturation

Cette fonction représente un compromis entre la fonction linéaire et la fonction seuil: entre ses deux bornes, elle confère au neurone une combinaison linéaire de l'entrée. À la limite, la fonction linéaire est équivalente à la fonction seuil (figure 2.4).

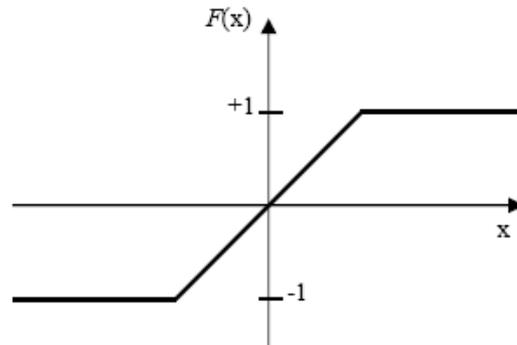


Figure 2.4 : Fonction d'activation à rampe avec saturation.

#### c)- Fonction sigmoïde

La fonction sigmoïde est une fonction continue qui maintient la sortie dans l'intervalle  $[0,1]$  (Figure 2.5). Son avantage principal est l'existence de sa dérivée en tout point. Elle est employée en général dans le perceptron multicouche [10].

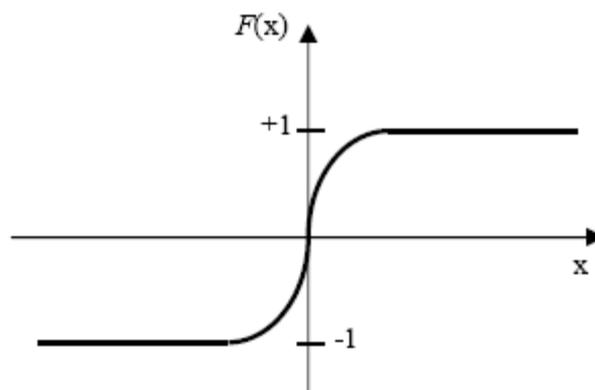


Figure 2.5 : Fonction d'activation sigmoïde.

## 2.4 Réseaux de neurones

Un réseau de neurones peut être représenté par un graphe direct composé d'un ensemble de noeuds ou éléments, processeurs, fortement interconnectés par des liens orientés ou connexion.

Les algorithmes neuronaux dépendent :

- de l'architecture, qui définit la façon dont la réponse est calculée et dont interviennent les poids.
- du paradigme d'apprentissage : apprentissage non-supervisé, supervisé ou encore par renforcement.

#### 2.4.1 Architecture des réseaux de neurones

L'architecture d'un réseau de neurones est fonction de plusieurs paramètres; le nombre de neurones, leurs fonctions d'activation et la nature de leurs connections qui peut être :

- Total : un neurone est connecté à l'ensemble des neurones du réseau y compris lui-même.
- Partielle : un neurone est connecté à un sous ensemble des neurones du réseau.

Les architectures peuvent aussi se différencier par la présence ou l'absence des boucles synaptiques (neurales) fermées. Cette caractéristique partage les réseaux de neurones en deux catégories :

- Réseaux statiques : chaque neurone d'une couche reçoit ces entrées à partir des neurones de la couche précédente. Il n'existe pas de retour d'informations (Feed-back).
- Réseaux dynamiques : Ils ne contiennent qu'une seule couche cachée. L'introduction de feed-back entre les neurones rend le réseau dynamique. L'étude de la stabilité de ce dernier devient alors nécessaire.
- Réseaux à architecture évolutive et auto-organisée : Les réseaux auto-organisés sont des réseaux qui changent leurs structures internes pendant l'utilisation. Ces réseaux sont dits évolutifs vu leurs méthodes d'apprentissage. Ainsi, la dimension du réseau change pendant l'entraînement, conséquence de la diminution ou l'augmentation du nombre de neurones.

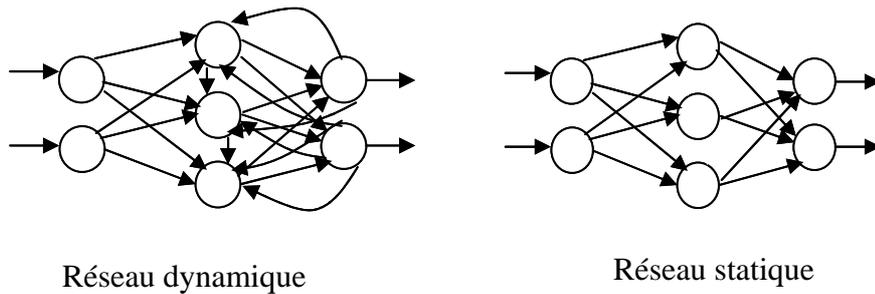


Figure 2.6 Réseau dynamique et statique

L'étude bibliographique que nous avons menée, révèle l'existence de plusieurs architectures intéressantes, ainsi que plusieurs critères de classifications des réseaux de neurones. Ceci explique par le nombre important de paramètres qui entrent en jeu ainsi que la possibilité de combinaison entre les anciennes architectures. De ce fait, nous nous sommes contentés de présenter quelques architectures qui ont un lien direct avec notre application. Ceci a fait l'objet de plusieurs travaux.

#### 2.4.1.1 Le perceptron

Le perceptron est la forme la plus simple d'un réseau de neurones. Il modélise la perception visuelle. Le perceptron comprend trois principaux éléments : la rétine, les cellules d'association et les cellules de décision. La fonction d'activation utilisée dans ce réseau est de type tout ou rien (0 ou 1). L'apprentissage du perceptron peut se faire avec plusieurs méthodes déjà utilisées, il n'y a qu'une seule couche de poids modifiables entre les cellules d'association et les cellules de décision. Le perceptron est limité dans ses applications. Premièrement, il ne peut être applicable que dans la classification dont les variables sont linéairement séparables et deuxièmement la sortie ne peut être que 0 ou 1.

#### 2.4.1.2 Perceptron multicouche

Cette classe est la plus importante des réseaux de neurones car elle représente la généralisation du perceptron monocouche avec une fonction d'activation de type sigmoïde et une ou plusieurs couches cachées. Le vecteur d'entrée se propage dans le réseau de couche en couche jusqu'à la sortie. L'entraînement de celui-ci se fait avec l'algorithme par la rétropropagation de

l'erreur[9]. Ce réseau est caractérisé par son modèle de neurones traitant les non-linéarités. Il peut comporter une ou plusieurs couches cachées et un plus grand nombre de connexions permettant de résoudre la majorité des problèmes.

Même avec les avantages des couches cachées et la performance de l'algorithme d'apprentissage, il reste plusieurs problèmes non réglés comme le choix du nombre de couches, le nombre de neurones par couche et le problème des minimums locaux où le réseau peut converger.

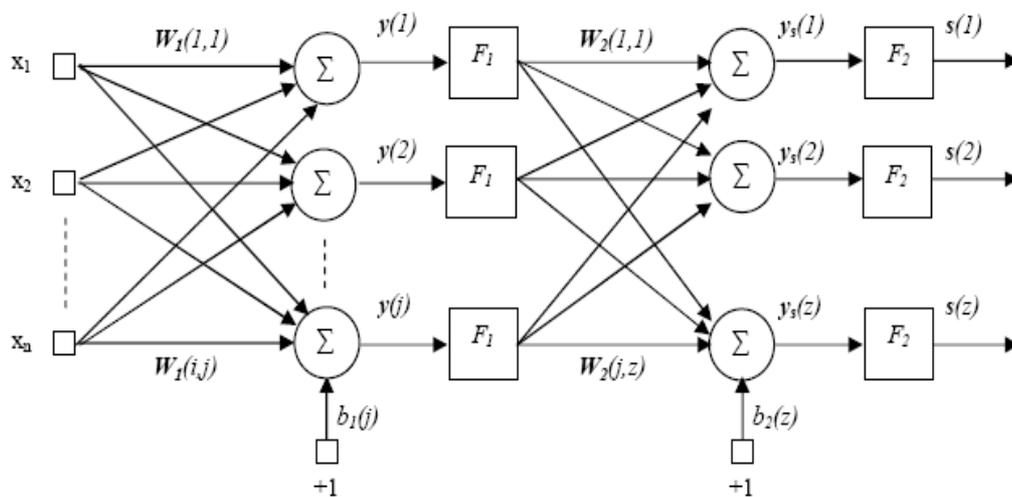


Figure 2.7 : Réseau de neurones multicouches.

### 2.4.1.3 Réseau de neurones linéaire

Cette classe de réseaux diffère du perceptron car elle possède un neurone dont la fonction d'activation est linéaire. L'une des règles d'apprentissage permet d'effectuer une descente de gradient de l'erreur sur une mesure d'erreur quadratique [9]. Les domaines d'application comprennent la commande, le contrôle et le traitement du signal. L'avantage de ce réseau est qu'il converge sur un seul minimum si la solution existe, sinon l'ajout de couches n'a aucun effet. Parmi ses inconvénients, il est limité à une couche de sortie et ne peut résoudre que les problèmes dont la relation entrées/sorties est linéaire.

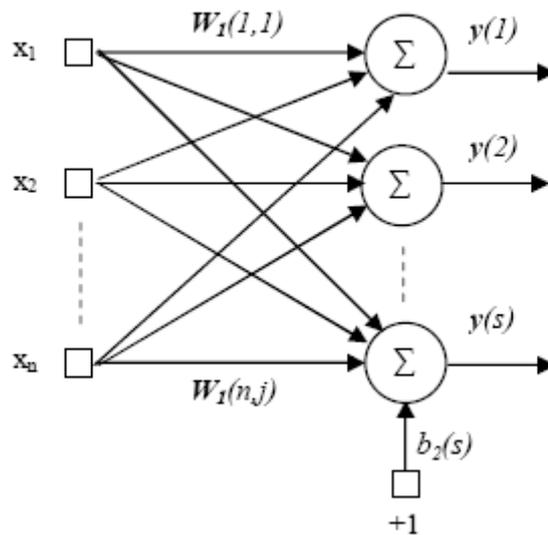


Figure 2.8 : Réseau de neurones mono-couche.

A La fin du cycle d'apprentissage, ces réseaux doivent aboutir à une structure optimale par rapport à la complexité du problème appris. Deux méthodes sont mises en oeuvre pour faire évoluer l'architecture du réseau de neurones :

- Une première méthode d'accroissement (growing network) consiste à ajouter des neurones à une architecture, minimale à la base, à chaque fois que l'erreur d'apprentissage ne décroît plus. La difficulté réside dans le choix d'un bon critère d'ajout de neurones.
- Une deuxième méthode est la méthode de simplification, de suppression, ou encore par élagage (pruning network). Cette fois-ci le réseau de neurones est sur-paramétré au départ. Des neurones sont ensuite supprimés au cours de l'apprentissage. Le critère de sélection des neurones à supprimer est plus facile à élaborer dans l'hypothèse ou la suppression intervient lorsque le réseau converge.

#### 2.4.2 Apprentissage des réseaux de neurones

L'apprentissage des réseaux de neurones est la phase de calcul de l'ensemble de ses poids par des algorithmes d'ajustement.

Trois classes d'apprentissage sont à distinguer.

### 2.4.2.1 Les différents types d'apprentissage

Les techniques d'apprentissage des réseaux de neurones se répartissent en trois grandes familles [9] :

- Apprentissage supervisé : Il se fait en présence d'un superviseur qui dirige le comportement du réseau en lui présentant les couples d'entrées et leurs sorties désirées. Les poids des connexions interneuronales sont alors modifiés de façon à minimiser une fonction appelée critère, qui évalue l'erreur entre la sortie du réseau et la sortie désirée. Ceci jusqu'à l'atteinte d'une bonne stabilisation du réseau. Nous considérons alors que, le réseau a appris et il est capable de distinguer entre les différents problèmes.
  
- Apprentissage non supervisé : L'apprentissage non supervisé nécessite des entrées seulement, sans l'intervention d'un superviseur. Le réseau neuronal s'organise de façon à créer des classes qui distinguent les différents exemples présentés. Après une certaine période, le réseau a appris et sera capable de ranger dans l'une des classes un futur exemple qui lui sera soumis.
  
- Apprentissage par renforcement : L'apprentissage est par renforcement quand le réseau de neurones interagit avec l'environnement. Contrairement à l'apprentissage supervisé, l'environnement ne fournit pas les réponses désirées, mais assigne une récompense quand la réponse du réseau est satisfaisante, ou assigne une pénalité si la réponse du réseau n'est pas satisfaisante. Le réseau doit ainsi découvrir les réponses qui lui donnent un maximum de récompenses.

Après le cycle d'apprentissage, le réseau de neurones est capable de répondre à n'importe quelle entrée. Il est intéressant de noter que le réseau neuronal n'utilise pas de modèle prédéterminé comme le font les approches statistiques. Il permet d'approximer avec précision n'importe quelle fonction linéaire ou non linéaire. Le choix du réseau, et par conséquent de l'algorithme, va essentiellement dépendre de l'application et de la disponibilité des informations.

#### 2.4.2.2 La procédure d'apprentissage

L'apprentissage d'un réseau se fait généralement dans le contexte d'une tâche ou d'un comportement à apprendre. Il se fait sur une période relativement longue, durant laquelle les patrons d'entrées (et éventuellement les patrons de sorties désirés) peuvent être présentés au réseau plusieurs fois.

Chaque apprentissage comprend quatre étapes de calcul :

- 1) Initialisation des poids synaptiques du réseau : La pratique courante est d'initialiser à des petites valeurs aléatoires.
- 2) Présentation du patron d'entrées et propagation de l'activation.
- 3) Calcul de l'erreur.
- 4) Calcul du vecteur de correction : à partir des valeurs de l'erreur, nous déterminons alors, la correction à apporter aux poids synaptiques des connexions et aux différents paramètres du neurone.

Les étapes 2,3 et 4 sont répétées jusqu'à la fin de l'apprentissage.

#### 2.4.2.3 Les règles d'apprentissage

Le choix d'un ensemble de poids synaptiques correct est un problème difficile, en raison de leur nombre et de la complexité de leur rapport avec le comportement global du réseau. C'est pourquoi, la majorité des réseaux de neurones disposent d'une règle d'apprentissage qui leur permet d'adapter leurs poids automatiquement.

Une règle d'apprentissage sera définie, d'une part, par la forme de la fonction coût et d'autre part, par la technique de minimisation. Elle est employée de manière à minimiser un critère « J » pour l'apprentissage. Un critère souvent utilisé est la somme des carrés des différences entre les sorties désirées et les sorties obtenues.

Donc, nous cherchons à résoudre :

$$\text{Min } (J (w)) \text{ avec } w \in R^n \quad (2..3)$$

Pour cela, plusieurs règles existent. Nous trouvons : la règle de HEBB, celle de WINDROW-HOFF sous autre nom « la règle delta » et les plus couramment utilisées, les règles qui se basent sur des procédures itératives [9].

- La règle de Hebb (ou règle de corrélation)

La première règle d'apprentissage a été grossièrement, formulée par Hebb en 1949. Il s'agissait de renforcer la connexion reliant deux neurones, à chaque fois qu'ils étaient actifs simultanément, dans le cas contraire, elle n'est pas modifiée.

Sa formalisation est la suivante :

$$W_{ij}(t+1) = W_{ij}(t) + \eta O_i O_j \quad (2.4)$$

Où,  $W_{ij}$  est le poids de la connexion reliant les neurones  $O_i, O_j$ .

$\eta$  : est un nombre compris entre 0 et 1, représentant le taux d'apprentissage.

t : représente l'étape d'apprentissage.

- La règle du Perceptron

A la fin des années cinquante Rosenblatt présenta le Perceptron qui fut le premier modèle solide. Sa fonction d'activation est une fonction discrète. Les sorties prennent des valeurs binaires (0 ou 1).

La règle d'apprentissage est la suivante :

$$W_{ij}(t+1) = W_{ij}(t) + \eta x_i \text{ si la sortie actuelle est 0 et doit être égale à 1.} \quad (2.5)$$

$$W_{ij}(t+1) = W_{ij}(t) - \eta x_i \text{ si la sortie actuelle est égale à 1} \quad (2.6)$$

et doit être égale à 0.

$$W_{ij}(t+1) = W_{ij}(t) \text{ si la sortie est correcte.} \quad (2.7)$$

Où,  $\eta > 0$  représente le coefficient d'apprentissage.

$x_i$  : L'entrée du neurone.

$W_{ij}$  : Le poids synaptique ou connexion entre i et j.

t : représente l'étape d'apprentissage.

L'inconvénient majeur du perceptron est qu'il ne peut s'appliquer que lorsque les classes sont linéairement séparables, donc il ne peut pas réaliser certaines fonctions de parité par exemple XOR.

- La règle de Windrow-Hoff ou règle delta

Après avoir étudié la règle du perceptron, Windrow - Hoff constatèrent que le Perceptron se limitait à des sorties binaires, ce qui les amena en 1960 à proposer une règle intéressante qui consistait à utiliser l'algorithme d'apprentissage du Perceptron, en considérant une fonction d'activation continue et différentielle.

Dans le Perceptron, le signal d'erreur utilisé, pour le calcul de la modification des poids, est égal à la différence entre la somme pondérée des entrées après le seuillage et le résultat attendu. Dans la règle de Windrow et Hoff le signal d'erreur est égal à la différence entre la somme pondérée des entrées non seuillées et le résultat attendu. Cette règle est aussi connue sous le nom de la méthode des moindres carrés dont le principe est le suivant:

- Calcul de l'erreur quadratique selon la formule :

$$E = \sum_{j=1}^n (d_j - y_j)^2 \quad (2.8)$$

Avec :

$$y = \sum_{i=1}^n x_i W_{ij} \quad (2.9)$$

- Minimiser cette erreur en modifiant les poids de chaque neurone suivant la règle :

$$W_{ij}(t+1) = W_{ij}(t) + \eta x_i (d_j - y_j) \quad (2.10)$$

$n$  : le nombre de neurones à la sortie

$d_j$  : est la sortie désirée

$y_j$  : est la sortie calculée

$x_i$  : l'entrée du  $i$  du neurone  $j$

$m$  : nombre de neurones à l'entrée

$\eta$  : coefficient d'apprentissage

La règle de WINDROW et HOFF pose le problème de l'apprentissage comme un problème de minimisation de l'erreur globale.

- Règle delta généralisée

La règle de delta généralisée ou encore appelée règle de la rétro-propagation du gradient est une généralisation de la règle Windrow et Hoff. Elle s'applique à un réseau multicouche utilisant des fonctions d'activation différentielles et un apprentissage supervisé. Afin de présenter cette règle, considérons un réseau à 3 couches.

L'apprentissage fonctionne comme suit :

On dispose d'un ensemble d'exemples qui sont des couples d'entrée /sortie désirée soit :

$X = (x_1; x_2; \dots; x_n)$  le vecteur d'entrées

$y = (y_1; y_2; \dots; y_n)$  le vecteur de sorties désirées.

Soit  $S = (s_1; s_2; \dots; s_n)$  le vecteur qui dénote les sorties réelles calculées.

Pour chaque étape, un exemple est présenté à l'entrée du réseau, une sortie réelle est calculée par la formule.

$$S_i = f\left(\sum_{i=1}^n W_{ji} x_i\right) \quad (2.11)$$

ou  $f$  dénote la fonction d'activation.

Ce calcul est effectué de proche en proche de la couche d'entrée à la couche de sortie. Cette phase est appelée propagation avant. Ensuite l'erreur quadratique,  $E$ , est calculée par la formule suivante :

$$E = \sum_{j=1}^n (y_j - s_j)^2 \quad (2.12)$$

Cette erreur est ensuite rétro-propagée dans le réseau, donnant lieu à une modification de chaque poids synaptique.

La règle de modification des poids synaptiques est :

$$\begin{aligned}
W_{ji}^s(t+1) &= W_{ji}^s(t) + \Delta W_{ji}^s(t) \\
W_{ji}^s(t) &= \eta \delta_j^s o_i^{[s-1]} + \alpha \Delta W_{ji}^s(t-1)
\end{aligned}
\tag{2.13}$$

avec  $\eta > 0$ ,  $0 \leq \alpha \leq 1$

Ou :  $s=1, 2,3$  (relativement à la couche d'entrée, la couche cachée, la couche de sortie).

$\delta_j^{[s]}$  : L'erreur locale

$\eta$  : le coefficient d'apprentissage

$\alpha$  : Momentun

Ce processus est répété, en présentant successivement chaque exemple. Si pour tous les exemples, l'erreur est inférieure à un seuil choisi, on dit alors que le réseau a convergé.

#### 2.4.2.4 Propriété des réseaux de neurones

L'intérêt principal porté aux réseaux de neurones tient sa justification dans les propriétés suivantes :

- La capacité d'apprentissage :

Comme l'être humain, un réseau de neurones apprend à résoudre des problèmes à partir d'exemples.

- La capacité de généralisation :

La capacité de généralisation se traduit par la capacité d'un système à apprendre et à retrouver, à partir d'un ensemble d'exemples des règles qui permettent de résoudre un problème donné non appris.

- Le parallélisme :

Cette notion se situe à la base de l'architecture des réseaux de neurones considérés comme un ensemble d'entités élémentaires qui travaillent simultanément. Le parallélisme permet une rapidité de calcul supérieure mais exige de penser et de poser les problèmes différemment.

#### 2.4.2.5 Les avantages des réseaux de neurones

Les principales qualités des réseaux de neurones sont leur capacité d'adaptabilité et d'auto-organisation et la possibilité de résoudre des problèmes non-linéaires avec une bonne approximation [9]. Ils ont une bonne immunité aux bruits et se prêtent bien à une implantation parallèle. La rapidité d'exécution est une qualité importante et elle justifie souvent à elle seule le choix d'implanter un réseau de neurones. Ces qualités ont permis de réaliser avec succès, plusieurs applications : classification, filtrage, compression de données, contrôleur, etc....

#### 2.4.2.6 Les inconvénients des réseaux de neurones

La difficulté d'interpréter le comportement d'un réseau de neurones est un inconvénient pour la mise au point d'une application. Il est souvent impossible d'utiliser les résultats obtenus pour améliorer ce comportement. Plusieurs paramètres doivent être ajustés et aucune méthode ne permet de choisir des valeurs optimales. Beaucoup d'heuristiques sont utilisées, mais elles se contredisent parfois et elles ne permettent pas toujours de trouver des valeurs optimales.

#### 2.4.2.7 Application des réseaux de neurones

On peut donc dresser une liste des domaines d'applications privilégiés :

- Robotique
- Classification et reconnaissance de formes
- Approximation de fonctions
- Optimisation
- Traitement du signal
- Modélisation

## 2.5 Conclusion

Nous avons décrit les fondements des réseaux de neurones artificiels à partir de la formulation mathématique d'un neurone formel. Nous avons vu plusieurs configurations de réseaux ainsi que plusieurs modèles d'apprentissage.

Le réseau MLP, il est difficile de plancher sur la meilleure configuration étant donné le caractère stochastique du nombre de couches nécessaires aussi du nombre de neurones nécessaire par couches permettant une configuration optimale.

Nous avons vu également les avantages et désavantages ainsi les applications des réseaux de neurones.

Finalement, pour conclure, nous en sommes encore à l'étape des essais et erreurs, pour avoir une meilleure configuration d'un réseau de neurone capable de résoudre de manière optimale un problème spécifique. Il est à espérer que dans un proche avenir les développements dans ce champ de recherche permettront de mieux saisir toute la complexité du cerveau humain et par le fait même nous permettre de découvrir de nouvelles possibilités d'utilisation des réseaux de neurones artificiels.

## CHAPITRE 3 METHODES DES PERTURBATIONS SINGULIERES ET LES CERCLES DE GERSCHGORIN

### 3.1 Introduction

Quand les grandeurs des modes du modèle sont très différentes, il est judicieux d'appliquer la méthode des perturbations singulières [11]. Le principe de cette méthode consiste à découpler les éléments qui ont des dynamiques distinctes en considérant leurs constantes de temps. Les éléments qui présentent une constante de temps petite ont des dynamiques rapides tandis que ceux qui ont une constante de temps élevée ont des dynamiques lentes. On dit que l'élément dynamique rapide se comporte comme un élément parasite qui augmente la dynamique du modèle [12]. Il faut noter que la séparation des dynamiques dépend des valeurs numériques du modèle. Par ailleurs, la méthode des perturbations singulières est valable pour les modèles linéaires ou non-linéaires, continus ou échantillonnés [30].

Le but principal de l'approche par la méthode des perturbations singulières est la diminution de la grande dimension et/ou du mal conditionnement des systèmes résultant de l'interaction entre les dynamiques lentes et rapides. Cette approche par la multi-échelles de temps est asymptotique. Elle est exacte à la limite lorsque le rapport  $\varepsilon$  des vitesses des dynamiques lentes et rapides tend vers zéro. Quand  $\varepsilon$  est petit, des approximations asymptotiques du modèle global sont obtenues à partir des modèles simplifiés d'ordre réduit et dans différentes échelles de temps séparées.

Son emploi dans le domaine de la modélisation des machines électriques pour différentes raisons, et notamment en contrôle est très récent. La mise en œuvre de cette théorie dans ce domaine a connu un développement très rapide [14][17].

L'utilisation de la technique des perturbations singulières pour la réduction des modèles de systèmes complexes exige l'identification et la séparation des variables

lentes et rapides. Ce problème disparaît par les méthodes géométriques pour certaines classes de modèles non linéaires.

Ceci nous a conduits à utiliser l'approche géométrique basée sur les cercles de Gerschgorin. Cette méthode donne une bonne sélection des variables lentes et rapides. Cependant, cette dernière nécessite une forme spéciale de la matrice d'état qui doit être diagonale dominante.

### 3.2 Méthode des perturbations singulières

Les systèmes singulièrement perturbés analysés par cette technique doivent avoir une forme spéciale dite standard. Cette forme standard s'écrit comme suit :

$$\begin{cases} \dot{x} = f(x, z, u, t, \varepsilon) & x \in \mathfrak{R}^n \\ \varepsilon \dot{z} = g(x, z, u, t, \varepsilon) & z \in \mathfrak{R}^m \end{cases} \quad (3.1)$$

Un modèle singulièrement perturbé d'un système dynamique est un modèle dans l'espace d'état dont les dérivés de quelques variables d'état sont multipliées par un petit paramètre positif  $\varepsilon$ . Et on suppose que les fonctions  $f$  et  $g$  sont suffisamment continues et dérivables par rapport à leurs arguments  $x, z, u, t, \varepsilon$  [14].

#### ▪ Cas d'un système linéaire singulièrement perturbé

Une démarche consiste, lorsque la matrice d'état possède des termes d'ordre de grandeur très différents, de réorganiser le vecteur d'état pour regrouper les termes en fonction de leur amplitude.

Supposons que le modèle  $\dot{X} = AX + BU$  se répartition en :

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} U, \quad (3.2)$$

où  $A_{21}, A_{22}, B_2$  sont très grands en module par rapport à ceux de  $A_{11}, A_{12}, B_1$ . Alors une normalisation peut être faite en introduisant un paramètre  $\varepsilon$ , tel que :

$$\begin{aligned} A_{21}^* &= \varepsilon A_{21} \\ A_{22}^* &= \varepsilon A_{22} \end{aligned}$$

$$B_2^* = \varepsilon B_2$$

et le modèle apparaît sous une forme singulièrement perturbée :

$$\begin{bmatrix} \dot{x} \\ \varepsilon \dot{z} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21}^* & A_{22}^* \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2^* \end{bmatrix} U, \quad (3.3)$$

Le paramètre  $\varepsilon$  peut être également calculé à partir des valeurs caractéristiques du modèle global obtenu après la diagonalisation en bloc, ainsi on peut citer :

$$\varepsilon = \frac{\min |\lambda(\hat{A}_1)|}{\max |\lambda(\hat{A}_2)|} \quad \text{ou} \quad \varepsilon = \|\hat{A}_1\| \|\hat{A}_2^{-1}\|, \quad \text{ou} \quad \varepsilon = \|A_{22}^{-1}\| (\|A_0\| + \|A_{12}\| \cdot \|L_0\|)$$

avec :

$$L_0 = -A_{22}^{-1} \cdot A_{21} \quad \text{et} \quad A_0 = A_{11} + A_{12}^{-1} \cdot L_0$$

Mise en œuvre sur un exemple :

Considérons l'exemple suivant [15], caractérisé par :

$$\dot{X} = AX + BU$$

avec :

$$A = \begin{bmatrix} -0.08 & 0.15 & 0 & 0.05 \\ 0 & -0.15 & 0.005 & 0 \\ 2.05 & 1.85 & -1.95 & 0 \\ 3.25 & 0 & 0 & -1.52 \end{bmatrix}, \quad B = \begin{bmatrix} 0.015 \\ 0.052 \\ 1 \\ 1.5 \end{bmatrix}$$

L'introduction du paramètre  $\varepsilon = 0.1$ , normalise les ordres de grandeur des termes de  $A$  et  $B$  :

$$A_{21}^* = \begin{bmatrix} 0.205 & 0.185 \\ 0.325 & 0 \end{bmatrix}, \quad A_{22}^* = \begin{bmatrix} -0.195 & 0 \\ 0 & -0.152 \end{bmatrix}, \quad B_2^* = \begin{bmatrix} 0.1 \\ 0.15 \end{bmatrix}$$

### 3.3 Méthode des perturbations singulières appliquée aux systèmes non-linéaires [14]

Un grand avantage de l'approche par la méthode des perturbations singulières sur les systèmes à deux échelles de temps est qu'elle n'est pas limitée aux systèmes linéaires contrairement à toutes les autres méthodes de simplification connues [15]. En effet, la méthode des perturbations singulières cause un comportement multi-échelles de temps et les dynamiques du système sont caractérisées par la présence des variables lentes et rapides.

Le modèle :

$$\begin{cases} \dot{x} = f(x, z, u, t, \varepsilon) & x \in \mathfrak{R}^n \\ \varepsilon \dot{z} = g(x, z, u, t, \varepsilon) & z \in \mathfrak{R}^m \\ y = h(x, z) \end{cases} \quad (3.4)$$

Avec :

$$\begin{aligned} x(t_0) &= x_{10} \\ z(t_0) &= x_{20} \\ \varepsilon &\in [0,1] \end{aligned}$$

Les fonctions  $f, g$  sont supposées suffisamment continues et dérivables par rapport à leurs arguments  $x, z, u, \varepsilon, t$ , et  $t_0 \in [0, T]$  et en notant  $x(t, \varepsilon)$  et  $z(t, \varepsilon)$  la solution du problème original (3.3).

Le modèle (3.3) est dit sous forme singulièrement perturbée car :

- L'introduction d'un petit, paramètre  $\varepsilon$  est considéré comme une perturbation.
- La valeur particulière  $\varepsilon = 0$  introduit une singularité, car  $m$  équations différentielles se transforment en équations algébriques.

$$g(x, z, u, t, 0) = 0 \quad (3.5)$$

Propriété:

On dit que le modèle (3.4) est sous la forme standard singulièrement perturbée si et seulement si l'équation (3.5) admet  $k$  racines réelles distinctes  $k \geq 1$  définies par :

$$z_s = q_i(x_s, u_s, t), \quad i = 1, 2, \dots, k \quad (3.6)$$

Pour tout  $(x, t) \in D_1 \times [t_0, T]$

Cette propriété assure qu'un modèle réduit bien défini de dimension  $n$  correspondra à chaque solution  $z_s$  de l'équation (3.6).

L'indice «  $s$  » sur les variables indique qu'elles appartiennent au modèle écrit avec  $\varepsilon = 0$ . En développant le modèle réduit caractérisé par  $\varepsilon = 0$ , on peut seulement spécifier  $n$  valeurs initiales parce que le modèle résultant est d'ordre  $n$ , donc on a retenu les états initiaux de  $x$ , soit le système d'état résultant :

$$\dot{x}_s = f(x_s, q(x_s, t), u_s, t, 0) \quad (3.7)$$

On note la solution de (3.7) par  $x_s(t, u_s)$ . La variable  $z$  est exclue du modèle réduit et elle est remplacée par son régime quasi-permanent  $q(x_s, t)$ . La seule information qu'on peut obtenir sur  $z$  est de calculer :

$$z_s = q(x_s(t), t) \quad (3.8)$$

qui décrit le comportement quasi-permanent de la variable  $z$  lorsque  $x = x_s$ .

La commande  $u$  est devenue  $u_s$ , car écrire  $\varepsilon = 0$  revient à considérer le système en régime quasi-permanent et à supposer que seules les composantes lentes de l'entrée sont encore actives. Les transitoires rapides ont donc disparus.

Pour obtenir le  $i^{\text{ème}}$  modèle réduit, il suffit de substituer (3.6) dans (3.4), soit :

$$\begin{aligned} \dot{x}_s &= f(x_s, q_i(x_s, u_s, t), u_s, t, 0) \\ y &= h(x_s, q_i(x_s, u_s, t)) \end{aligned} \quad (3.9)$$

En gardant la même condition initiale sur  $x_s$  que sur  $x$  :

$$x_s(t_0) = x_0$$

Dans la suite nous utiliserons la notation plus compacte :

$$\begin{aligned} \dot{x}_s &= f(x_s, u_s, t) \\ y &= h(x_s, u_s, t) \\ z_s &= q(x_s, u_s, t) \\ z_s(t_0) &= q(x_s(t_0), u_s(t_0), t_0) \end{aligned} \quad (3.10)$$

Ce modèle réduit est appelé modèle lent ou le modèle quasi-permanent car il donne l'évolution du système quand les transitoires rapides ont disparus.

Posons :

$$\varepsilon \frac{dz}{dt} = \frac{dz}{d\tau} \quad \text{avec} \quad \frac{d\tau}{dt} = \frac{1}{\varepsilon}$$

Et utilisons  $\tau = 0$  comme valeur initiale à  $t = t_0$  la nouvelle variable temporelle :

$$\tau = \frac{t - t_0}{\varepsilon}, \quad \tau = 0 \text{ pour } t = t_0 \quad (3.11)$$

est dilaté, et  $\tau$  tend vers l'infini quand  $\varepsilon$  tend vers 0, même pour un temps  $t$  légèrement supérieur à  $t_0$ . De plus, alors que  $z$  et  $\tau$  changent presque instantanément,  $x$  reste très proche de sa valeur initiale  $x_0$ .

Pour décrire le comportement de  $z$  comme fonction de  $\tau$ , on utilise la correction de couche limite :

$$z_f = z - z_s$$

Qui nous permet de représenter le système original comme suit :

$$\begin{aligned} \frac{dx}{dt} &= f(x, z_f + q(x, u_s, t), u, t, \varepsilon) \\ \varepsilon \frac{dz_f}{dt} &= g(x, z_f + q(x, u_s, t), u, t, \varepsilon) - \varepsilon \frac{\partial}{\partial x} q(x, u_s, t) \cdot \frac{\partial x}{\partial \tau} - \varepsilon \frac{\partial}{\partial t} q(x, u_s, t) \end{aligned}$$

Afin de profiter de l'ordre petit du terme  $\varepsilon$ , on exprime le modèle précédent dans l'échelle de temps rapide ; soit :

$$\begin{aligned} \frac{dx}{d\tau} &= \varepsilon f(x, z_f + q(x, u_s, t), u, t, \varepsilon) \\ \varepsilon \frac{dz_f}{d\tau} &= g(x, z_f + q(x, u_s, t), u, t, \varepsilon) - \frac{\partial}{\partial x} q(x, u_s, t) \cdot \frac{\partial x}{\partial \tau} - \varepsilon q(x, u_s, t) \end{aligned}$$

Le modèle réduit dans cette échelle de temps est obtenu en posant  $\varepsilon = 0$  et on suppose que  $\frac{\partial u_s}{\partial \tau} = O(\varepsilon)$ . Par conséquent, le système réduit vérifiant l'équation de

couche limite s'écrit :

$$\begin{aligned} \frac{dz_f}{d\tau} &= g(x_0, z_f(\tau), z_s(\tau), u(\tau), t_0) \\ y_f &= h(x_0, z_f(\tau) + z_s(\tau)) \\ z_f(0) &= z_0 - z_s(t_0) \end{aligned} \quad (3.11)$$

Cette équation comporte le terme  $u(\tau)$  qui, à cette échelle de temps  $\tau$ , correspond au comportement rapide de la variable  $u$ , défini par :  $u_f = u - u_s$ .

Avec  $z_s, z_f$  décrivant respectivement les composantes lentes et rapides de  $z$ .

### 3.4 Méthode des perturbations singulières appliquée aux modèles continus linéaires

Soit le modèle suivant sous la forme singulièrement perturbée

$$\begin{bmatrix} \dot{x} \\ \varepsilon \dot{z} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} U, \quad (3.12)$$

et 
$$y = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix}$$

avec :

$$\begin{aligned} x(t_0) &= x_0, \\ z(t_0) &= z_0 \end{aligned}$$

Il est dit sous forme explicite si la matrice  $A_{22}$  est inversible et sous forme implicite dans le cas contraire. Ce dernier cas suppose que le découplage en  $x(\text{lent})$  et  $z(\text{rapide})$  est mauvais et que les différentes dynamiques sont encore réparties entre  $x$  et  $z$ , il faut donc faire un changement de variables pour regrouper les dynamiques.

#### 3.4.1 Calcul des modèles réduits lent et rapide [14][15]

Le modèle réduit lent est obtenu en considérant que les variables rapides  $z$  ont atteint leur régime établi, ce qui correspond à écrire  $\varepsilon = 0$ .

Le modèle devient :

$$\begin{cases} \dot{x}_s = A_{11}x_s + A_{12}z_s + B_1u_s \\ 0 = A_{21}x_s + A_{22}z_s + B_2u_s \\ y_s = C_1x_s + C_2z_s \end{cases} \quad (3.13)$$

où  $x_s, z_s, u_s, y_s$  représentent les composantes lentes des variables  $x, z, u, y$ .

Après transformations, on obtient le système lent réduit d'ordre  $n$  :

$$\begin{cases} \dot{x}_s = A_s x_s + B_s u_s \\ y_s = C_s x_s + D_s u_s \\ z_s = -A_{22}^{-1}(A_{21}x_s + B_2u_s) \end{cases} \quad (3.14)$$

$$\begin{aligned}
A_s &= A_{11} - A_{12}A_{22}^{-1}A_{21} \\
B_s &= B_1 - A_{12}A_{22}^{-1}B_2 \\
C_s &= C_1 - C_2A_{22}^{-1}A_{21} \\
D_s &= -C_2A_{22}^{-1}B_2
\end{aligned}$$

avec

Si  $A_{22}$  est inversible, le modèle (3.14) fournit une bonne approximation du comportement du modèle global, qui sera valable pour  $t \geq t_0$ , si on fixe :

$$x_s(t_0) = x_0$$

Par contre, la valeur initiale des composantes lentes  $z_s$  vaut :

$$z_s(t_0) = -A_{22}^{-1}A_{12}x_s(t_0)$$

Ce qui est en général différent de  $z_0$ . Les variables rapides  $z$  ne peuvent donc pas être approchées par  $z_s$  sur tout l'intervalle de temps  $[0, T]$ .

Introduisant un terme correctif  $z_f$ , défini par :

$$z_f = z - z_s$$

qui représente les variations rapides de  $z$  donc, l'équation de couche limite, exprimée en temps dilaté  $\tau$  ( $\varepsilon \frac{dz_f}{dt} = \frac{dz_f}{d\tau}$ ) :

$$\begin{cases}
\frac{dz_f}{d\tau} = A_{22}z_f(\tau) + B_2u_f(\tau) \\
y_f = C_2z_f(\tau) \\
z_f(t_0) = z_0 + A_{22}^{-1}A_{21}x_0
\end{cases} \quad (3.15)$$

est obtenue par la différence entre le modèle initiale (3.12) et le modèle simplifié (3.14), en considérant immobiles les variables lentes pendant les transitoires rapides ( $\frac{dx_s}{d\tau} = 0$ ).

Remarque : la variable  $z_f(\tau)$  n'a pas de sens que durant un court intervalle  $[t_0, t_1]$  dit «domaine de couche limite».

Cette décomposition en modèles découplés lent et rapide est présentée par la figure (3.2) à partir de la forme initiale présentée par figure (3.1).

### 3.4.2 Application

Considérons le système d'état, correspondant à la forme singulièrement perturbée :

$$\begin{bmatrix} \dot{x} \\ \varepsilon \dot{z} \end{bmatrix} = \begin{bmatrix} -0.08 & 0.04 & -0.01 \\ -0.01 & -0.1 & -0.14 \\ 0.011 & 0.002 & -0.09 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0.1 \end{bmatrix} u$$

$$y = [1.0 \quad 0.0 \quad -0.5] \begin{bmatrix} x \\ z \end{bmatrix}$$

$$x(t_0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, z(t_0) = [1]$$

Conformément aux équations (3.14) à (3.15), nous obtenons les deux sous modèles suivants :

- Sous modèle lent

$$\frac{d}{dt} x_s = \begin{bmatrix} -0.0812 & 0.0398 \\ -0.0271 & -0.1031 \end{bmatrix} x_s + \begin{bmatrix} 0.9889 \\ -0.1556 \end{bmatrix} u_s$$

$$y_s = [0.9389 \quad -0.0111] x_s + [-0.5556] u_s$$

$$x_s(t_0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$z_s = [0.1222 \quad 0.0222] x_s + [1.1111] u_s$$

- Sous modèle rapide

$$\frac{d}{dt} z_f = \left[ -\frac{1}{\varepsilon} \cdot 0.09 \right] z_f + \left[ \frac{1}{\varepsilon} \cdot 0.1 \right] u_f$$

$$y_f = z_f$$

$$z_f(t_0) = 0.8778$$

### 3.4.3 Interprétation

La validité de la méthode des perturbations singulières dépend du terme parasite  $\varepsilon$ , c'est-à-dire du taux de séparabilité des dynamiques.

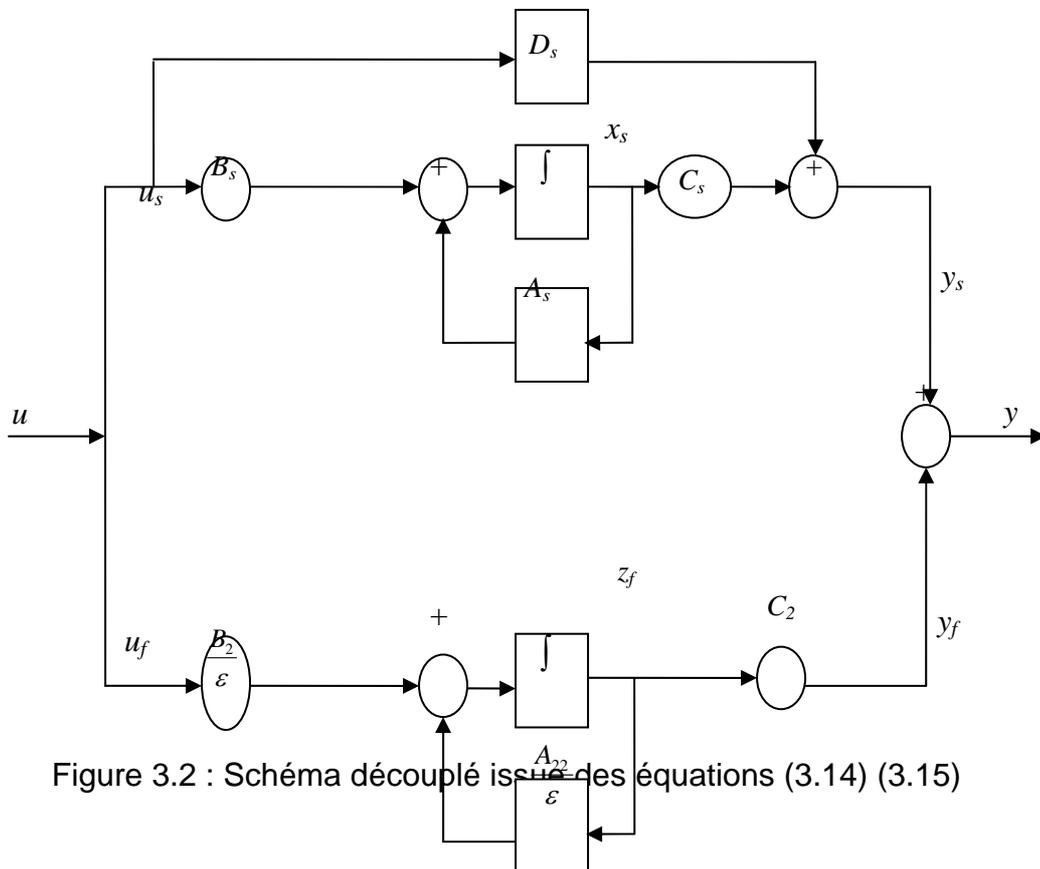
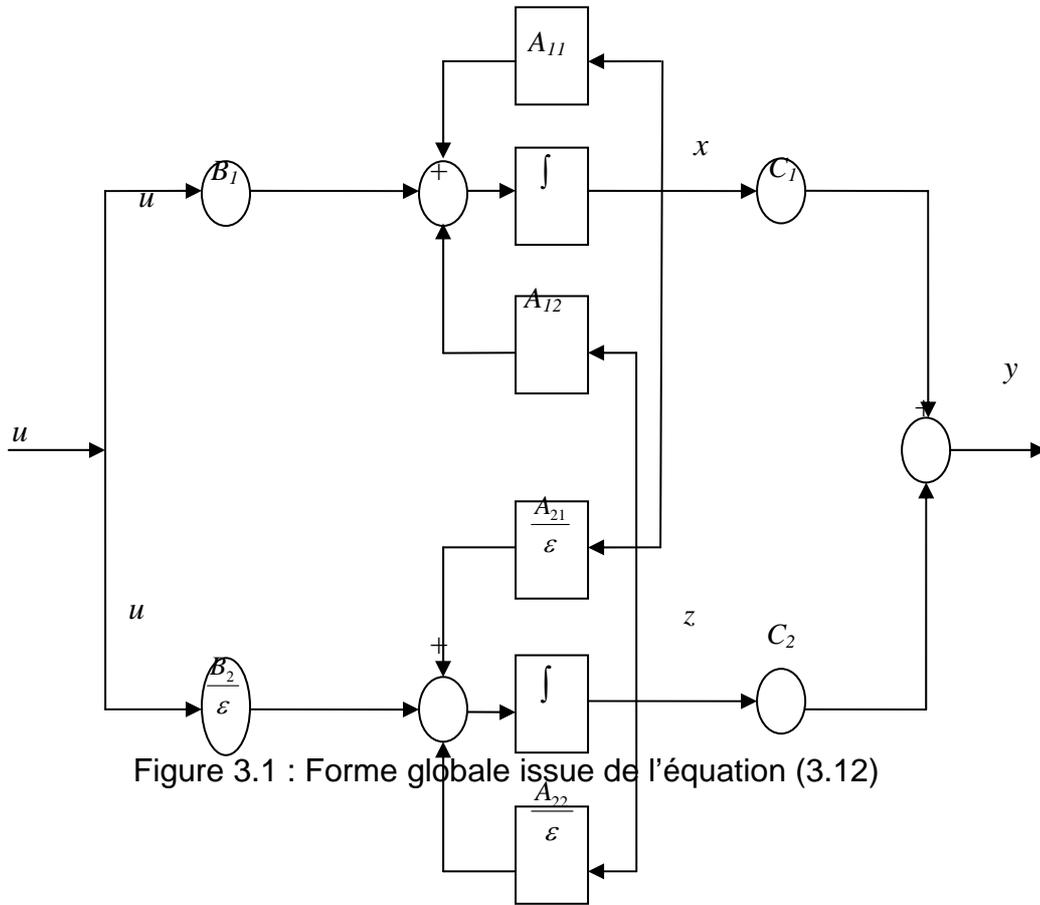
Les valeurs propres de la matrice d'état pour  $\varepsilon = 0.1$  sont :

$$\lambda(A_1) \quad \{(-0.0920 + j0.0322), (-0.0920 - j0.0322), (-0.8961)\}$$

Elles sont approchées de façon satisfaisante par les valeurs propres des  $A_s$  et  $A_f$  :

$$\lambda(A_{1s}) \quad \{(-0.0922 + j0.0310), (-0.0922 - j0.0310)\}$$

$$\lambda(A_{1f}) \quad \{(-0.9000)\}$$



### 3.5 Identification géométrique des dynamiques [ 14] [17][16]

Notons  $(a_{ij}, i, j = 1, \dots, n)$  les éléments de la matrice carrée d'état A est :

$$p_i = \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|, \quad i = 1, 2, \dots, n \quad (3.16)$$

$$Q_i = \sum_{\substack{i=1 \\ j \neq i}}^n |a_{ij}|, \quad j = 1, 2, \dots, n \quad (3.17)$$

#### 3.5.1 Cercles de Gerschgorin

Les cercles de Gerschgorin sont généralement utilisés pour développer des algorithmes plus simples destinés à identifier les différentes échelles de temps dans les systèmes linéaires. En effet, après des transformations algébriques que subit la matrice d'état, cette dernière se présente sous forme diagonale dominante et dont les valeurs propres correspondent aux différentes échelles de temps. Des conditions nécessaires et suffisantes sont énoncées pour que cette technique soit correctement applicable.

- Théorème 1: [14][17]

Les valeurs propres de la matrice A appartiennent au domaine  $D_1$  du plan complexe formé par la réunion des régions intérieures des  $n$  cercles de Gerschgorin  $C_i$  centrés en  $a_{ii}$  et de rayon  $R_{li}$  :

$$R_{li} = p_i$$

et dans le domaine  $D_2$  formé par la réunion des  $n$  interférences de centres  $a_{ii}$  et de rayon  $R_{ci}$  :

$$R_{ci} = Q_i$$

- Théorème 2 : [14][17]

1- Chaque valeur propre de la matrice carrée A appartient à au moins un des cercles  $C_i$ , de centre  $a_{ii}$  et de rayon :

$$R_i = R_{li} \quad \text{ou} \quad R_i = R_{ci}$$

2- Si  $m$  cercles  $C_i(a_{ii}, R_i)$  forment un domaine séparé des autres cercles, alors il y'a exactement  $m$  valeurs propres situées dans ce domaine.

- Théorème 3: [14] (Système à deux dynamiques).

Si on peut définir deux ensembles d'indices  $I$  et  $K$ , avec  $I \cap K = \emptyset$  et  $I \cup K = \{1, \dots, n\}$ ,

Tels que :

$\forall (i, j) \in I \times K$ , les cercles  $C_i(a_{ii}, R_i)$  et  $C_k(a_{kk}, R_k)$  vérifient :

$$|a_{ii} - a_{kk}| \geq (R_i + R_k) \quad \forall i \in I \quad \text{et} \quad \forall k \in K,$$

Alors la matrice  $A$  possède deux ensembles de valeurs propres séparés. Si  $|a_{ii}|$  est supposé plus grand que  $|a_{kk}|$  les variables  $x_i$ ,  $i \in I$  sont alors rapides et  $x_k$ ,  $k \in K$  sont lentes.

-Taux de séparabilité [11]

Le taux de séparabilité des dynamiques peut être estimé par la relation :

$$\varepsilon_s = \sup(\varepsilon_1, \varepsilon_2), \quad 0 < \varepsilon_s < 1.$$

avec :

$$\varepsilon_1 = \max_{\substack{i \in I \\ k \in K}} \left( \frac{R_i + R_k}{|a_{ii} - a_{kk}|} \right) \quad \text{et} \quad \varepsilon_2 = \max_{\substack{i \in I \\ k \in K}} \left( \frac{|a_{kk}|}{|a_{ii}|} \right) \quad (3.18)$$

Remarque :

1. Les deux paramètres  $\varepsilon_1$  et  $\varepsilon_2$  ont des rôles complémentaires :  $\varepsilon_1$  indique le degré de couplage entre les différentes variables, et  $\varepsilon_2$  correspond au rapport entre les différentes échelles de temps.

2. La valeur de  $\varepsilon_s$  dépend du conditionnement de la matrice  $A$ . Pour diminuer les rayons des cercles, il est parfois utile d'effectuer un calibrage des termes hors diagonale par un changement de base, afin de rendre la matrice à diagonale dominante.

3. La condition nécessaire et suffisante pour l'application correcte du théorème des cercles de Gerschgorin est que la matrice d'état soit diagonale dominante. En plus, il faut noter que chaque matrice linéaire peut se transformer en une autre à diagonale dominante [14][17].

### 3.5.2 .application

- Cas d'un modèle linéaire

Considérons le modèle suivant :

$$\dot{\tilde{x}} = \begin{bmatrix} -2 & 0 & 0 & 0 & -0.5 \\ 3.493 & -5 & 0 & 0 & 0 \\ 0 & 0.908 & -0.167 & 0 & 0 \\ 0 & 0 & 0.5 & -2 & 0 \\ 0 & 0.273 & 0.047 & 0.28 & -0.1125 \end{bmatrix} \cdot \tilde{x}, \quad (3.19)$$

Description :

- Le tracé des cercles de Gerschgorin est représenté à la figure (3.3).
- Les cercles de Gerschgorin indiquent deux différentes échelles de temps, avec les sous-ensembles de variables d'état  $(x_3, x_5)$  et  $(x_1, x_2, x_4)$ , respectivement lent et rapide correspondant au rapport  $\varepsilon_s = 0.747$ .

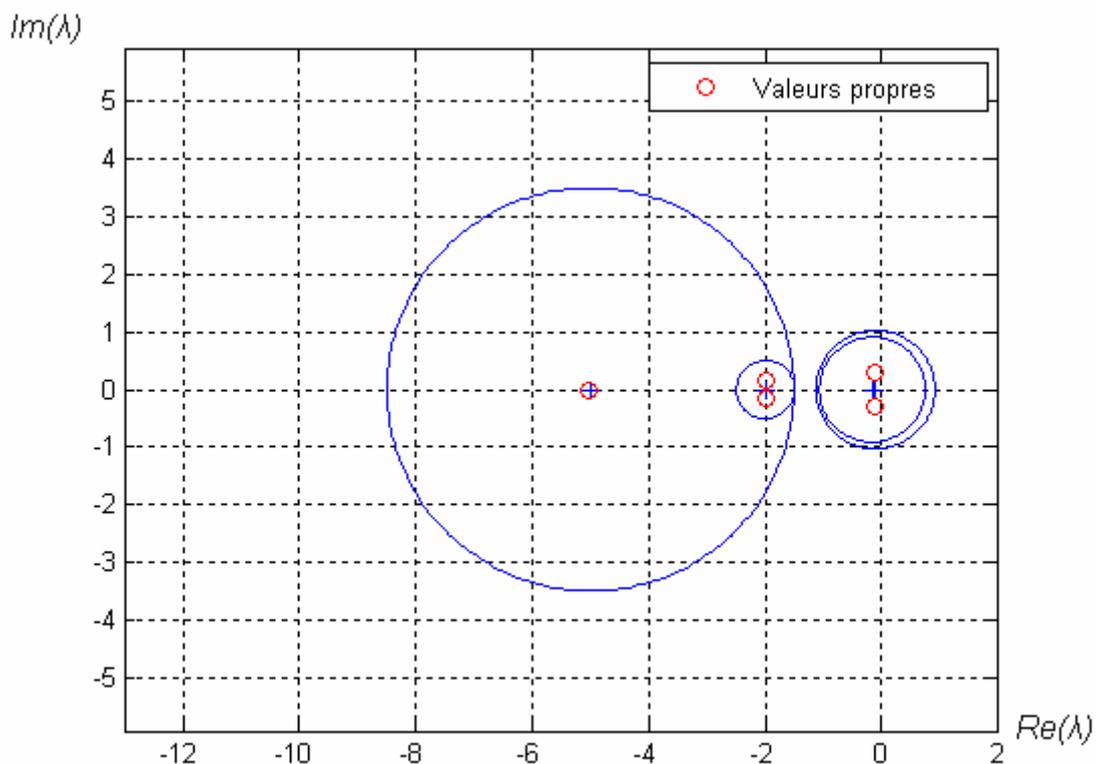


Figure 3.3 : Cercles de Gerschgorin pour un modèle linéaire 3.19

### 3.5.3 Transformations modales

La technique des perturbations singulières peut réduire l'ordre des modèles complexes pour permettre leur analyse et contrôle. Le problème majeur de cette réduction est l'écriture du modèle mathématique sous la forme standard singulièrement perturbée [17].

Une solution pour remédier à ce problème est de trouver une matrice équivalente de la matrice de base sous forme bloc diagonale ou à diagonale dominante pour garantir la séparation correcte des dynamiques lentes et rapides.

Des algorithmes dans ce sens ont été développés par [16] [14][17]. Ils traitent le cas des systèmes linéaires singulièrement perturbés dont il est possible de faire une extension sur certaines classes de modèles non linéaires. Cependant, la convergence de ces algorithmes est conditionnée par la bonne sélection des dynamiques lente et rapide du système. En effet, le théorème de cercles de Gerschgorin est un bon test de la convergence.

Pour un système linéaire présentant deux échelles de temps et modélisé sous la forme implicite(3.1), il est difficile de le poser sous la forme standard singulièrement perturbée le problème se résume à :

1. l'évaluation de la dimension  $n$  et  $m$  des vecteurs lent et rapide.
2. l'identification et l'arrangement correct, parmi les composantes de  $\underline{X}$ , des modes lents et rapides.
3. l'évaluation du terme  $\varepsilon$ .

Pour traiter les points 1 et 2, il n'y pas des méthodes systématiques, par contre, le terme  $\varepsilon$  peut être estimé par le rapport entre les valeurs absolues de la petite valeur propre des composantes lentes et la grande valeur propre des composantes rapides.

### 3.5.3.1 Changement de taille des rayons des cercles [17]

Soit la matrice :

$$S_k = \text{diag}(1, \dots, 1, \alpha_k, 1, \dots, 1), \quad k = 1, 2, \dots, n \quad (3.20)$$

Le changement de base  $\underline{X}' = S_k \underline{X}$  fait apparaître la nouvelle matrice d'état sous la forme :

$$\begin{bmatrix} a_{1,1} & \cdot & \cdot & \cdot & a_{1,k-1} & a_{1,k}/\alpha_k & a_{1,k+1} & \cdot & \cdot & \cdot & a_{1n} \\ \cdot & & & & \cdot & \cdot & \cdot & & & & \cdot \\ \cdot & & & & \cdot & \cdot & \cdot & & & & \cdot \\ \cdot & & & & \cdot & \cdot & \cdot & & & & \cdot \\ a_{k-1,1} & \cdot & \cdot & \cdot & a_{k-1,k-1} & a_{k-1,k}/\alpha_k & a_{k-1,k+1} & \cdot & \cdot & \cdot & a_{k-1,n} \\ a_{k-1}\alpha_k & \cdot & \cdot & \cdot & a_{k,k-1}\alpha_k & a_{k,k} & a_{k,k+1}\alpha_k & \cdot & \cdot & \cdot & a_{k,n}\alpha_k \\ a_{k+1,1} & \cdot & \cdot & \cdot & a_{k+1,k-1} & a_{k+1,k}/\alpha_k & a_{k+1,k+1} & \cdot & \cdot & \cdot & a_{k+1,n} \\ \cdot & & & & \cdot & \cdot & \cdot & & & & \cdot \\ \cdot & & & & \cdot & \cdot & \cdot & & & & \cdot \\ \cdot & & & & \cdot & \cdot & \cdot & & & & \cdot \\ a_{n1} & \cdot & \cdot & \cdot & a_{n,k-1} & a_{n,k}/\alpha_k & a_{n,k+1} & \cdot & \cdot & \cdot & a_{nn} \end{bmatrix} \quad (3.21)$$

Les rayons  $R_{1k}$  et  $R_{ck}$  deviennent respectivement  $R_{1k}\alpha_k$  et  $R_{ck}/\alpha_k$ . Un choix judicieux de  $\alpha_k$  peut introduire une séparation entre les cercles.

Si l'opération est répétée plusieurs fois, la transformation obtenue est :

$$\begin{cases} \underline{X} = S \underline{X} \\ \underline{A} = S \underline{A} S^{-1} \end{cases}$$

avec :  $S = \prod_k S_k$

S'il y'a deux ensembles disjoints de cercles, la matrice de permutation est donc trouvée :

$$\begin{cases} \underline{X} = P \underline{X} \\ \underline{A} = P \underline{A} P^{-1} \end{cases} \quad (3.22)$$

### 3.5.3.2 Déplacement des centres des cercles[17]

Il est parfois nécessaire, pour améliorer la séparation des dynamiques par les cercles de Gerschgorin d'introduire un déplacement des cercles. Ce déplacement est caractérisé par la transformation suivante :



$$\begin{cases} X = P\underline{X} \\ A = P\underline{A}P^{-1} \end{cases} \quad (3.27)$$

### 3.5.3.3 Application

Considérons le modèle suivant :

$$\dot{x} = \frac{1}{2} \begin{bmatrix} -11 & 9 & -9 \\ 1 & -3 & -1 \\ -8 & 8 & -12 \end{bmatrix} \cdot x + \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \cdot u \quad (3.28)$$

Les rayons de la matrice de base

		<i>rayons</i>	
	$A = \frac{1}{2} \begin{bmatrix} -11 & 9 & -9 \\ 1 & -3 & -1 \\ -8 & 8 & -12 \end{bmatrix}$	9 1 8	
<i>rayons</i>	4.5   4.5   5		(3.29)

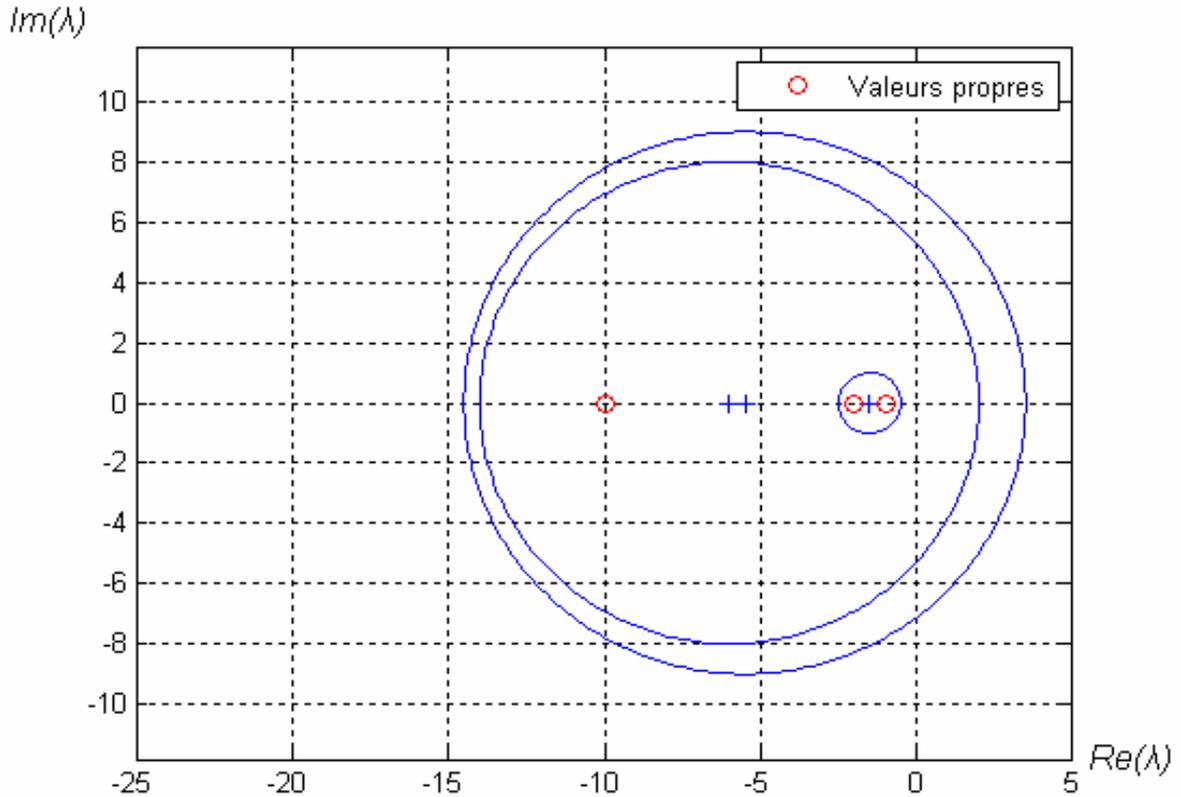


Figure 3.4 Cercles de Gerschgorin pour le modèle 3.28

Les cercles ne sont pas disjoints. Les termes diagonaux (-11) et (-12) sont proches, donc on utilise une transformation de type  $T$  pour déplacer les centres des cercles 1 et 3. La matrice  $T$  utilisée porte sur la ligne 1 et sur la colonne 3.

L'équation :  $X_{13} = -9 - B_1 + 8.B_1^2 = 0$  est vérifiée pour  $B_1 = -1$ . Soit la matrice sous forme standard :

$$A^* = \frac{1}{2} \begin{matrix} & \begin{matrix} \text{rayons} \\ \text{rayons} \end{matrix} \\ \begin{matrix} \text{rayons} \\ \text{rayons} \end{matrix} & \begin{bmatrix} -3 & 1 & 0 \\ 1 & -3 & 0 \\ -8 & 8 & -20 \end{bmatrix} \end{matrix} \begin{matrix} 0.5 \\ 0.5 \\ 8 \end{matrix} \quad (3.30)$$

Description :

- Le tracé des cercles de Gerschgorin après la transformation est représenté à la figure (3.5).
- Les cercles de Gerschgorin indiquent deux différentes échelles de temps, avec les sous-ensembles de variables d'état  $(x_1, x_2)$  et  $(x_3)$ , respectivement rapide et lent pour laquelle les cercles sont disjoints.

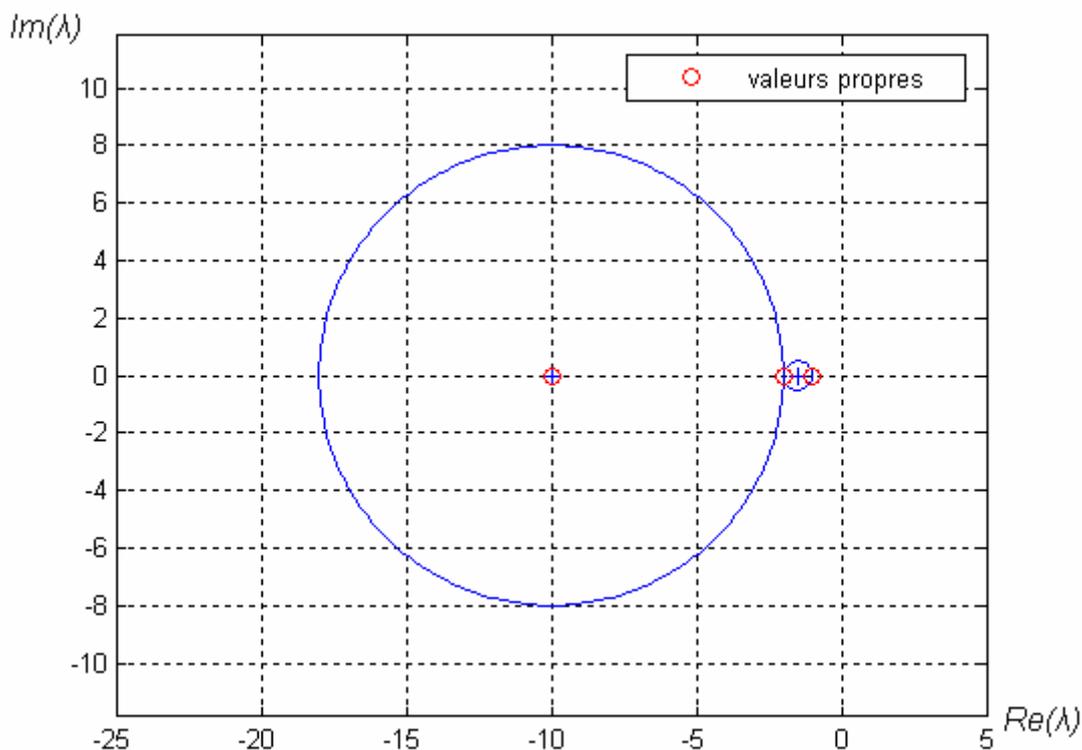


Figure 3.5. Cercles de Gerschgorin pour le modèle 3.30

### 3.5.4 Regroupement et séparation des dynamiques d'un système linéaire

Dans ce contexte, on présente un algorithme simple pour le regroupement et la séparation des différentes échelles de temps d'un système linéaire. Cet algorithme basé sur les cercles de Gerschgorin suppose que la matrice d'état est à diagonale dominante [16].

- Etape 1 (Réorganisation du vecteur d'état)

Comme la localisation des valeurs propres est liée aux lignes de telle sorte à regrouper les termes de la diagonale en fonction de leur amplitude d'ordre croissant. Cette réorganisation permet l'utilisation correcte de la technique des perturbations singulières.

Ce processus nécessite la transformation suivante :

$$\begin{cases} \tilde{A} = P^T A P \\ \tilde{B} = P^T B \\ \tilde{C} = C P^T \end{cases} \quad (3.31)$$

La matrice  $P$  caractérise le changement de lignes correspondant à l'ordre croissant des valeurs propres de  $A$  et  $P^t$  sa transposée. Les éléments de la matrice  $P$  seront déterminés comme suit :

$$\begin{cases} P_{ij} = 1, & \text{si la } i^{\text{ième}} \text{ ligne sera transférée à la } j^{\text{ième}}, \\ P_{ii} = 1, & \text{si la } i^{\text{ième}} \text{ ligne ne sera pas à permuter,} \\ P_{ij} = 0, & \text{autrement.} \end{cases} \quad (3.32)$$

- Etape 2 (Séparation des dynamiques)

L'utilisation des cercles de Gerschgorin met en évidence les différentes dynamiques du système. Si les cercles présentent  $q$  ensembles disjoints, alors le système est à  $q$  échelles de temps.

Par exemple, un système à deux échelles de temps sera partitionné comme suit :

$$\tilde{A} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} B_{11} \\ B_{22} \end{bmatrix}, \quad \tilde{C} = [C_{11} \quad C_{22}] \quad (3.33)$$

Le bloc de ligne  $C = [A_{11} \ A_{22}]$  est associé à la réunion des cercles qui contient les petites valeurs propres en valeurs absolues et correspond au modèle dit lent. Si les dynamiques sont suffisamment séparées. Les modèles réduits lent et rapide peuvent se calculer facilement en utilisant la technique des perturbations singulières.

### 3.5.5 Application

- Séparation des dynamiques

Considérons le modèle suivant d'ordre 5[16] :

$$\begin{aligned} \frac{d}{dt} \underline{X} &= \underline{A} \cdot \underline{X} + \underline{B} \cdot U \\ Y &= \underline{C} \cdot \underline{X} \end{aligned} \quad (3.34)$$

avec :

$$\underline{A} = \begin{bmatrix} -17.0 & 0.2 & 0.0 & -0.5 & -0.5 \\ -0.1 & -16.5 & -0.4 & 0.0 & -1.0 \\ 0.5 & -0.03 & -1.5 & -0.02 & -0.05 \\ -0.3 & 0.0 & -1.1 & -16.0 & -0.2 \\ -0.01 & -0.4 & 0.0 & 0.04 & -1.2 \end{bmatrix}, \quad \underline{B} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\underline{C} = [1 \ 0 \ 1 \ 0 \ 0] \quad \text{et} \quad \underline{X}_0 = [0 \ 0 \ 0 \ 0 \ 0]^T$$

Pour partitionner le vecteur d'état correctement, on utilise l'algorithme précédent (étape 1). En effet, la matrice P est :

$$P = \begin{bmatrix} 0 & 0 & 0. & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (3.35)$$

et la nouvelle représentation d'état est :

$$\underline{A} = \begin{bmatrix} -1.2 & 0.0 & 0.04 & -0.4 & -0.01 \\ -0.05 & -1.5 & -0.02 & 0.03 & 0.5 \\ -0.2 & -1.1 & -16.0 & 0.0 & -0.3 \\ -1.0 & -0.4 & 0.0 & -16.5 & -0.1 \\ -0.5 & 0.0 & -0.5 & 0.2 & -17.0 \end{bmatrix}, \quad \underline{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad (3.36)$$

$$\underline{C} = [1 \ 1 \ 0 \ 0 \ 1]$$

Le tracé des cercles est représenté sur la figure (3.5). On remarque deux ensembles disjoints des cercles, donc il y'a deux échelles de temps et les sous-vecteurs  $x = [x_1 \ x_2]^T$  et  $z = [x_3 \ x_4 \ x_5]^T$  sont respectivement lent et rapide et leurs valeurs initiales sont  $x_0 = [0 \ 0]^T$  et  $z_0 = [0 \ 0 \ 0]^T$ .

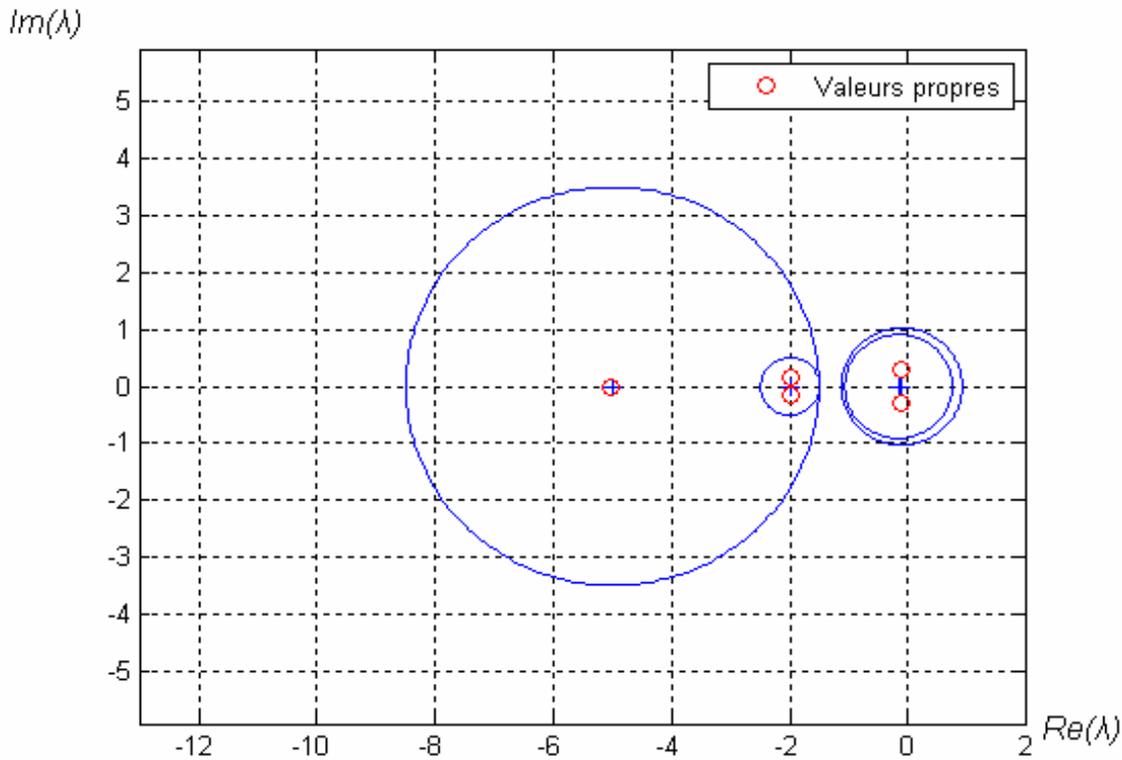


Figure 3.6 : Cercles de Gerschgorin du modèle 3.36

- Application de la technique des perturbations singulières

En appliquant les équations (3.14) à (3.15), les deux sous systèmes sont :

-Modèle réduit lent :

$$\frac{d}{dt} x_s = \begin{bmatrix} -1.1760 & 0.0069 \\ -0.0628 & -1.4970 \end{bmatrix} x_s + \begin{bmatrix} -0.9758 \\ -0.0015 \end{bmatrix} u_s,$$

$$y_s = [0.9702 \ 1.0013] x_s + [7.1336 * 10^{-4}] u_s \quad (3.37)$$

$$z_s = \begin{bmatrix} -0.0119 & -0.0688 \\ -0.0604 & -0.0243 \\ -0.0298 & 0.0017 \end{bmatrix} x_s + \begin{bmatrix} 0.00000 \\ 0.0606 \\ 0.0007 \end{bmatrix} u_s$$

avec :  $x_{s0} = [0 \ 0]^T$

-Modèle réduit rapide :

$$\frac{d}{dt} z_f = \begin{bmatrix} -16.0000 & 0.0000 & -0.3000 \\ 0.00000 & -16.5000 & -0.1000 \\ -0.5000 & 0.2000 & -17.0000 \end{bmatrix} z_f(\tau) + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u_s(\tau)$$

$$y_f(\tau) = [0 \ 1 \ 1] z_f(\tau)$$

(3.38)

$$z_{f0} = z_0 + \begin{bmatrix} 0.0119 & 0.0688 \\ 0.0604 & 0.0243 \\ 0.0298 & -0.0017 \end{bmatrix} x_0$$

### 3.6 Conclusion

Dans ce chapitre, nous avons rappelé quelques notions de la décomposition multi échelles de temps basée sur la technique des perturbations singulières. Son application suppose en préambule une mise sous forme plus ou moins proche de la forme standard singulièrement perturbée. Dans le cas des systèmes à multi-échelles de temps, les décompositions des modèles en sous-systèmes lents et rapides permettent d'analyser et de traiter ces modèles avec moins de complexités et de calculs.

En outre, nous avons présenté une approche géométrique basée sur les cercles de Gerschgorin pour l'identification et la séparation des dynamiques. Pour améliorer la séparation des dynamiques, nous avons procédé à des transformations modales afin de remédier à quelques problèmes d'identification des modes lents et rapides.

En effet, nous avons présenté des algorithmes de changement de tailles des rayons et de déplacement des centres des cercles. Nous avons mis en évidence l'utilisation d'un algorithme efficace pour le regroupement et la séparation correcte des dynamiques.

L'exploitation de cette approche pour les systèmes singulièrement perturbés permet de développer des modèles réduits qui représentent de manière suffisante le comportement global du système avec moins de complexité. Ces modèles peuvent être utilisés pour des raisons de simulation, d'analyse et de contrôle.

## **CHAPITRE 4 ARCHITECTURE NEUROGLIALE**

### 4.1 Introduction

Aujourd'hui les réseaux de neurones artificiels montrent leurs limites dans le paradigme de leur processus d'information. Paradigme désignant l'ensemble des éléments qui constitue un champ d'interprétation d'une réalité donnée à un moment donné [5]. Les modèles actuels se basent sur les connaissances du passé et certains éléments nouveaux devraient être intégrés dans les nouveaux modèles pour corriger ce paradigme. Depuis que les RNA(réseaux de neurones artificiels) essayent d'émuler le cerveau, les chercheurs n'ont cessé de porter leur attention sur l'importance des neurones dans le système nerveux [5].

Néanmoins depuis les dernières décennies, l'importance du système glial a été observé et l'on pense que le GS (système glial) intervient dans le processus d'informations du système nerveux d'une manière beaucoup plus appuyée qu'auparavant [5] et que ces cellules gliales communiquent également, formant un réseau séparé, parallèle au réseau neuronal [3]. Ces deux réseaux ont deux vitesses différentes.

Dans ce cas, il peut être utile d'intégrer dans les réseaux de neurones artificiels des connaissances supplémentaires, soit en concevant des neurones plus complexes, soit en élaborant des architectures plus complexes. L'élaboration des neurones plus complexes, proche de la réalité biologique fait appel à toutes sortes de techniques, telles que rajouter des paramètres ou utiliser des fonctions plus complexes. Cette complexification rend l'apprentissage plus difficile, la convergence plus lente.

Notre objectif n'est pas d'obtenir des modèles neuronaux forcément plus complexes, mais d'aboutir à des modèles des neurones plus riches c'est-à-dire capable de mémoriser plus d'informations. Une alternative, pour apprendre mieux

sans complexifier les neurones, consiste à mieux les organiser en utilisant des approches neuronales modulaires.

La biologie est également basée sur une structure modulaire.

L'article « Design and Evolution of Modular Neural Network Architectures » [18] examine la structure des preuves pour une architecture modulaire dans le cerveau humain qui est donnée par différents psychologues, des biologistes et des neurologues. Plusieurs niveaux de la modularité dans le cerveau sont décrits. Multitâche, capacités humaines et les handicaps sont expliqués avec la structure modulaire et la structure parallèle du cerveau [18].

Happle et Murre concluent que la nature du traitement de l'information dans le cerveau est modulaire [18]. Différentes fonctions sont réparties dans les sous-réseaux qui peuvent être exécutés dans des modules séparés sans interférence. Ils ont, en outre, montré qu'il existe un processus de subdivision en sous modules et les tâches en sous-tâches à un niveau très élémentaire, et que l'architecture modulaire du cerveau, qui s'est développée dans un long processus évolutif, est le principal sujet de cette division des tâches complexes en sous-tâches.

Les travaux récents suggèrent une construction plus modulaire des réseaux neuronaux artificiels à réseau glial qui sont similaires à la structure du cerveau. Ces nouvelles architectures ont pour rôle d'augmenter la capacité du réseau à résoudre des problèmes plus complexes du monde réel.

Sur la base de ces connaissances une nouvelle approche modulaire Neurogliale a été développée dans ce chapitre.

#### 4.2 Le réseau artificiel Neuroglial

Ce réseau inclut des neurones artificiels et des éléments de contrôle qui représentent les astrocytes. Le fonctionnement du réseau Neuroglial suit les différentes étapes qui ont été appliquées avec réussite dans la construction et l'utilisation du système de connectivité: conception, entraînement, tests et exécution.

Nous proposons une nouvelle structure qui inclut deux réseaux de neurones distincts : l'un correspond à un réseau rapide qui est constitué des sous réseaux et un autre correspond à un réseau lent.

Deux formes de modularité dans la conception des architectures neuronales utilisent plusieurs réseaux, deux approches opposées existent [19].

- La modularité à modèles multiples ou modularité itérative, désignée en anglais par « Ensemble based approach ». Lorsque plusieurs modules traitent tous le même problème et que leurs sorties sont combinées pour calculer la sortie globale [19].
- La modularité vraie, appelée aussi modularité différenciée et en anglais « Neural net ensemble », lorsque un regroupement de réseau est composé de modules qui apprennent séparément les sous problèmes d'un problème complet complexe [19].

La conception de la nouvelle approche Neurogliale implique l'utilisation de la modularité vraie et l'exécution des 4 étapes suivantes :

1. Décomposition de la tâche en plusieurs sous-tâches.
2. l'organisation de l'architecture modulaire.
3. La nature de l'apprentissage.
4. La communication entre les modules.

#### 4.2.1 La décomposition

La décomposition d'une tâche en sous-tâches peut se faire, soit explicitement soit automatiquement.

La décomposition explicite doit être faite par le concepteur du réseau [20] [21]. Elle est un excellent moyen d'insérer dans l'architecture neuronale de la connaissance à priori sur le problème. La décomposition neuronale est alors directement déduite de la structure du problème. Pour un problème donné, la décomposition est particulière et ne sera pas transposable à un autre problème. La

décomposition doit être significative selon les propriétés mécaniques et physiques du système [20] [21].

Lorsqu'on ne dispose d'aucune information préalable sur le problème, on opte généralement pour une décomposition automatique [20] [21]. Pour ce type de décomposition, une partition des données de la base d'apprentissage est souvent plus performante qu'un apprentissage sur tout un espace d'entrée. Il existe deux façons de procéder, la décomposition automatique peut ce faire, soit avant, soit pendant la phase d'apprentissage.

- Avant l'apprentissage :

Cette méthode consiste dans un premier temps à partitionner la base d'apprentissage; puis à faire apprendre chaque module séparément sur un sous-ensemble de la base d'apprentissage.

- pendant l'apprentissage :

Un partitionnement pendant la phase d'apprentissage signifie que les modules se répartissent et s'organisent en cours d'apprentissage. L'exemple le plus connu est l'approche des Mélanges d'experts (ME) ou un module annexe (gating network, parfois appelé module auxiliaire) supervise la répartition des experts au cours d'apprentissage et contrôle la décomposition en sous- problèmes [20] [22].

Deux genres de décomposition peuvent être réalisés :

Une décomposition peut être exécutée sur l'espace d'entrée ou sur les variables d'entrée [21].

- Décomposition horizontale :

Cette décomposition concerne l'espace d'entrée :

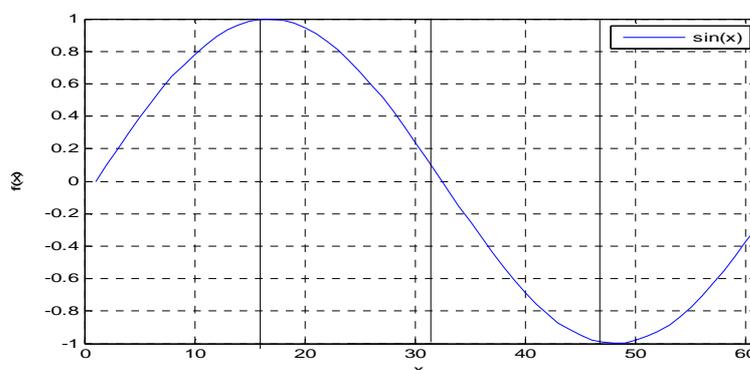


Figure 4.1: Exemple de décomposition horizontale

La fonction  $f(x)$  est décomposée en 4 parties homogènes à partir de l'axe des  $x$ .

- Décomposition verticale :

Cette décomposition concerne la variable d'entrée. Les entrées sont regroupées en paquets. Chaque paquet est lié à une fonction différente. Chacune des fonctions est estimée par différents modules. La combinaison des sorties de ces modules mène à estimer des fonctions plus complexes [21]. L'exemple illustré en figure 4.2 montre une fonction simple décomposée en deux sous fonctions d'une variable chacune.

Toutes les sorties sont combinées pour fournir une réponse globale.

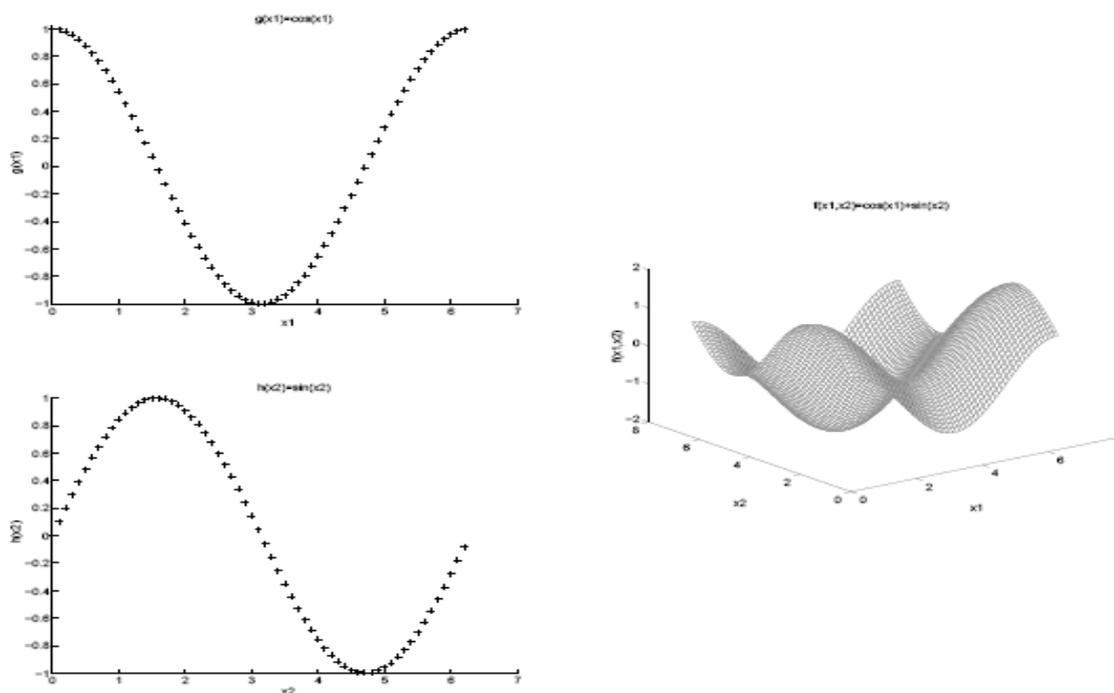


Figure 4.2 : Exemple de décomposition verticale

La fonction de deux entrées  $f(x_1, x_2)=\cos(x_1) + \sin(x_2)$  est décomposée en deux fonctions secondaires d'une variable chacune.

La décomposition d'un système totalement inconnu n'est pas toujours possible. Cependant, les connaissances, à priori, que nous avons sur les systèmes physiques, que nous cherchons à simuler, permettent de dégager une décomposition.

Une autre façon d'analyser les formes de modularité revient à étudier la nature de la décomposition, c'est à dire l'agencement des modules les uns par rapport aux autres.

#### 4.2.2 L'organisation de l'architecture modulaire

Nous distinguerons deux manières d'inter-connecter les modules entre eux. Elles peuvent être parallèles ou séquentielles (séries). Nous parlerons d'architecture parallèle lorsque chacun des modules est indépendant et d'architecture série lorsque qu'il existe une dimension temporelle entre les modules.

La combinaison de décompositions séquentielles et parallèles conduit à des structures hiérarchiques.

- Les architectures séquentielles

Dans une architecture séquentielle (ou série), une tâche est décomposée en sous-tâches successives. L'entrée est transformée au fur et à mesure qu'elle traverse les différents modules qui composent l'architecture. Une architecture séquentielle est présentée en figure 4.3. Les sorties du  $i^{ième}$  module servent d'entrées au module  $i^{ième} + 1$ . Cette forme de décomposition est très répandue.

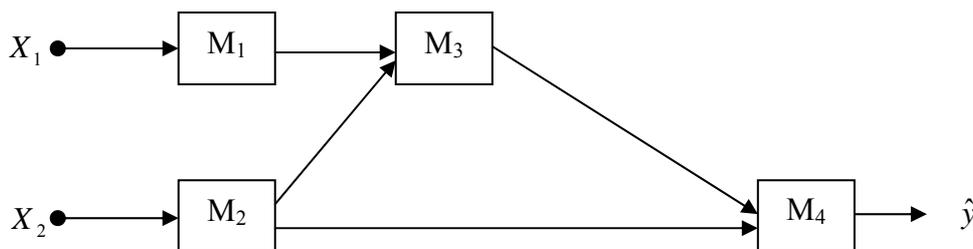


Figure 4.3 : Architecture séquentielle

- Les architectures parallèles

Dans les architectures parallèles, tous les modules travaillent et traitent leurs informations simultanément (figure 4.4).

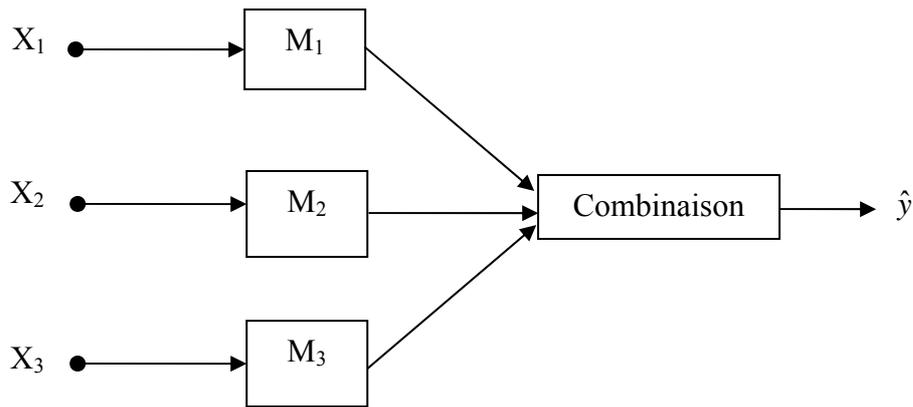


Figure 4.4 : Architecture parallèle

La sortie globale fait intervenir ou non chacun des modules, tout dépend des applications.

Il y a deux décompositions parallèles distinctes, la relation de coopération et celle de surveillance.

#### - La coopération

Il existe au moins deux façons d'établir une relation de coopération entre modules la décomposition ET et la décomposition OU [8].

Une décomposition est appelée décomposition ET lorsque la résolution d'une tâche implique la résolution de toutes les sous-tâches. Une décomposition est qualifiée de décomposition OU lorsque la résolution d'une tâche revient à résoudre une des sous-tâches.

Par conséquent, les modules d'une décomposition ET ne traitent qu'une partie du vecteur d'entrée global et la sortie globale nécessite la réponse de tous les modules.

Inversement, les modules d'une décomposition OU prennent le vecteur d'entrée en entier et la réponse d'un seule module est suffisante pour calculer la réponse globale alors que dans le cas de la décomposition ET chaque module apprend à traiter une partie du vecteur d'entrée.

#### - La surveillance

On parle aussi de supervision quand un module contrôle un autre. Ce principe permet de coordonner les activités du système d'apprentissage ; le travail peut être réparti parmi différents réseaux, les sorties de chacun sont fusionnées pour le calcul de la réponse globale. Comme exemple, on peut citer le réseau sélecteur de l'approche ME qui supervise les modules et pondère leurs sorties [22] [24].

#### ▪ Les architectures hiérarchiques

L'architecture hiérarchique est une structure en arbre. Les modules d'un niveau sont connectés à un nombre fini de modules du niveau inférieur. Un exemple d'une telle architecture est illustré par la figure 4.5. Cette architecture combine des modules en séries et des modules en parallèles. Les modules au sein d'un niveau sont parallèles, alors que les modules qui lient une entrée à une sortie sont en série. Ces décompositions sont les plus courantes et parmi elles, sont rangés les modèles HMES, autrement dit les mélanges d'experts hiérarchiques [24].

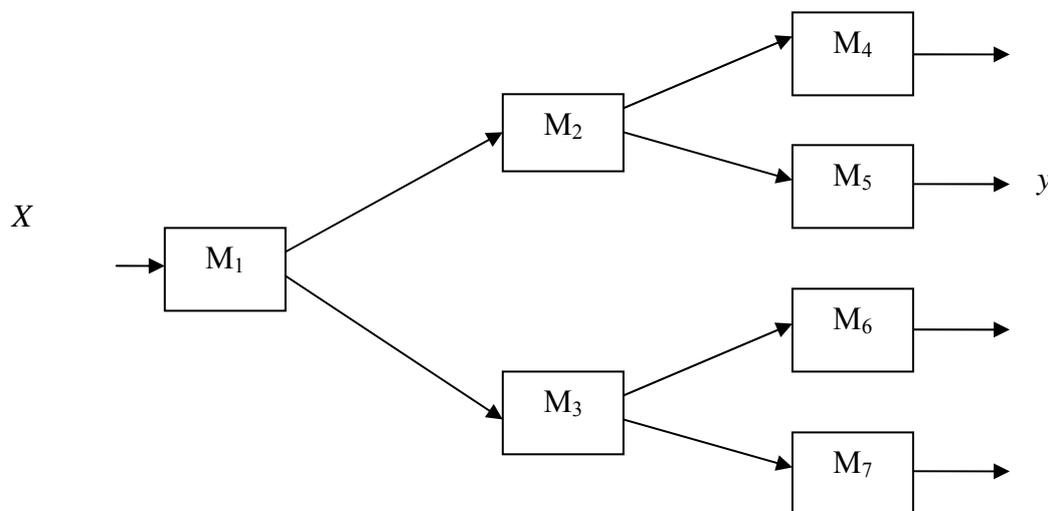


Figure 4.5 : Architecture hiérarchique

#### 4.2.3 Nature de l'apprentissage

Dans le principe d'apprentissage d'un système modulaire, trois types d'apprentissage peuvent être associés :

#### - L'apprentissage indépendant

L'apprentissage d'un module est réalisé de façon indépendante lorsque les autres modules n'interviennent et n'influent pas pendant la phase d'apprentissage. Les modules ne sont reliés que pour la phase d'utilisation/restitution. Ce principe est doté d'un avantage intéressant : il est possible d'apprendre des sous-problèmes plus simples [25]. L'architecture est fixe et déterminée au départ.

#### - L'apprentissage coopératif

Cette forme d'apprentissage ne s'applique qu'aux architectures fixes et déterminées dès le départ. Mais la fonction de chacun des modules est déterminée au cours de l'apprentissage. Autrement dit, le nombre de modules ainsi que leur topologie est prédéterminée et chaque module apprend par lui-même à se spécialiser sur une sous-tâche. L'apprentissage est réalisé sur tous les modules en même temps, à l'aide des entrées et des sorties globales. Un exemple est donné par les Mélanges d'experts [24], où les différents experts se voient attribuer un rôle par le superviseur en cours d'apprentissage. Ils se spécialisent alors sur une portion de l'espace des entrées.

#### - L'apprentissage incrémental

Cet apprentissage supporte une architecture évolutive. Lorsque le nombre de modules, leurs connectivités et leurs fonctions changent au cours de la phase d'apprentissage. L'apprentissage est appelé incrémental.

Les réseaux neuronaux modulaires sont de toutes évidences une avancée significative dans la théorie de l'apprentissage.

La nature de l'apprentissage dépend de la nature de la décomposition, des connaissances, à priori, mais aussi de la disponibilité de certaines grandeurs internes du système à apprendre.

#### 4.2.4 La communication entre les modules

Les techniques pour calculer la sortie globale d'une assemblée de réseaux sont diverses et variées. Il y a entre autres, les techniques de votes, la moyenne et la moyenne pondérée et la combinaison non linéaire. Parmi les techniques de votes, on a les votes pondérés [26]. Un poids est alors associé à chaque classificateur, il

représente une mesure de performance. Concernant la moyenne, la sortie est donnée par la combinaison linéaire de sorties des réseaux. On a aussi la technique qui minimise l'erreur quadratique moyenne [27] de la sortie globale.

Utiliser les sorties combinées est une alternative efficace à la méthode qui consiste à choisir le réseau qui présente les meilleures performances. Or, les sorties de plusieurs réseaux délivrent des sorties différentes, des temps de convergences différents, ou même des architectures différentes. L'élément déterminant l'efficacité d'un regroupement d'un réseau de neurones est indépendant de ses membres en termes d'erreurs [28]. Pour qu'un ensemble donne des performances supérieures à celles d'un réseau unique, il faut faire en sorte que ces réseaux restent indépendants.

Nous avons présentés les démarches et les références sur lesquelles nous appuyons pour proposer une architecture modulaire et autonome. Cette approche utilise le principe de <<diviser pour mieux régner >>.

Un problème donné est divisé en sous problèmes plus simples. Ce système modulaire apprend à décomposer la tâche globale dans le sens où il met en compétition des réseaux de neurones en différentes régions de l'espace d'entrée. Le résultat de cette division se traduit par le fait que les différents réseaux apprennent des exemples différents, ils apprennent des fonctions différentes.

#### 4.3 L'architecture du réseau Neuroglial

L'architecture de cette nouvelle approche est composée de K modules de réseaux rapides et d'un réseau lent (appelé aussi réseau glial ou glial network en Anglais). Le principe est illustré par la figure 4.6.

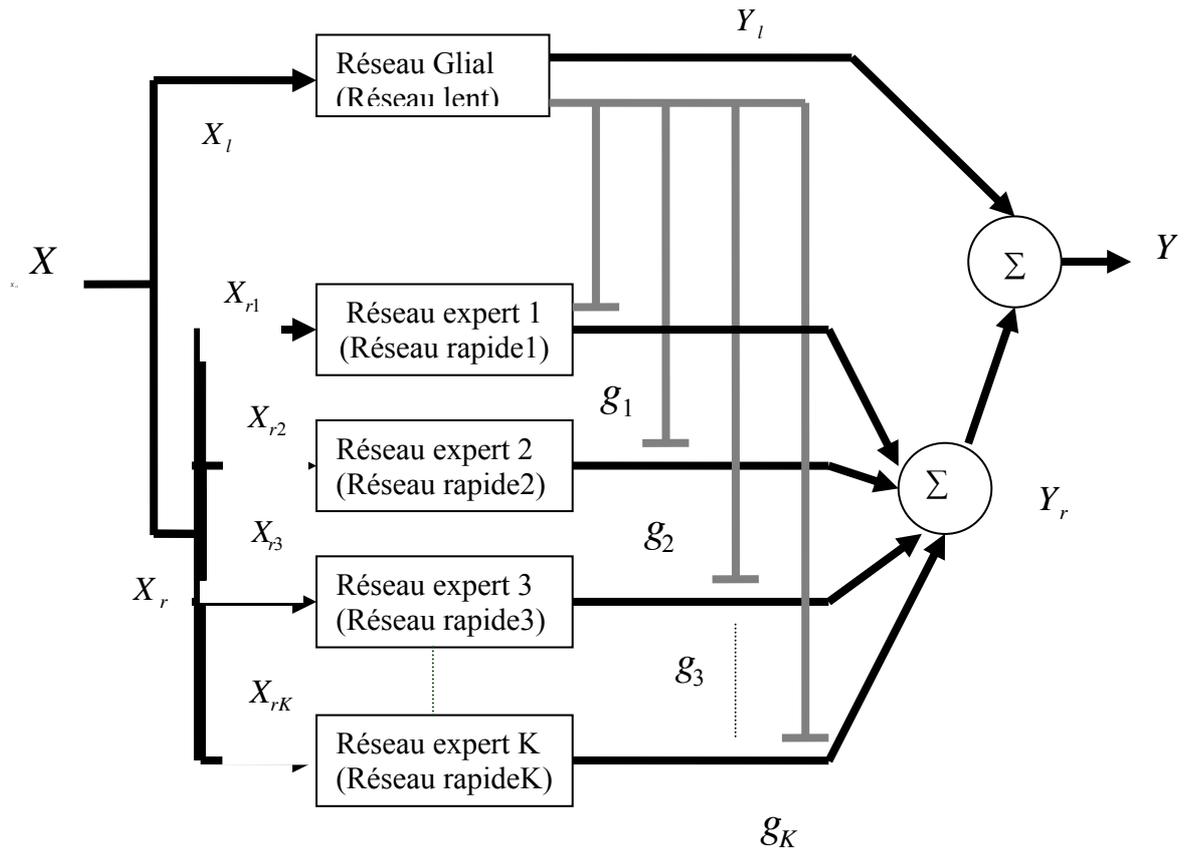


Figure : 4.6 Architecture Neurogliale

Par la suite, nous n'emploierons plus les termes "rapide" et "lent" et nous nommerons les modules des réseaux rapides et réseau lent simplement « experts » et « glial » respectivement. Experts et réseau glial ont accès aux variables d'entrées  $X_l$  et  $X_r$ . Ces entrées  $X_l$  et  $X_r$  sont les entrées lentes et rapides respectivement.

Le réseau glial possède deux vecteurs de sortie : le 1<sup>er</sup> vecteur de sortie est utilisé pour le contrôle et la supervision des réseaux experts.

En outre, le 2<sup>ème</sup> vecteur de sorties représente la réponse du réseau glial (réponse lente), cette réponse doit être soumise à la réponse du réseau expert activé pour avoir la réponse globale du réseau Neuroglial à chaque vecteur d'entrée présenté.

Le réseau glial possède autant de sorties qu'il y a d'experts plus le nombre de sorties qui seront combinées aux sorties des experts.

Dans la discussion qui suit, la dimension du vecteur de sortie de chaque expert est désignée par  $m$ , le nombre d'experts par  $K$  et le nombre de sortie du réseau glial par  $t$  donc on a :  $t = K + m$

Dans notre approche, le vecteur d'entrée  $X_l$  du réseau glial est différent du réseau expert  $X_r$ , mais de même dimension.

Suivant les étapes citées ci dessus, nous disons que cette structure est une forme de modularité vraie puisque chaque réseau expert se spécialise sur un sous problème. Elle est basée sur une décomposition verticale de variables d'entrée : les variables d'entrée sont décomposées aux variables lentes  $X_l$  et variables rapides  $X_r$  et une décomposition horizontale de l'espace d'entrée  $X_r$  :

- Pour le 1<sup>er</sup> algorithme : chaque partie du vecteur d'entrée est connectée à un seul réseau expert.
- Pour le 2<sup>ème</sup> algorithme : chaque partie du vecteur d'entrée est connectée à tous les réseaux experts.

Cette structure est parallèle du type ET entre le réseau glial et les experts : les réponses de ces réseaux doivent être combinées pour avoir la réponse globale et une structure parallèle du type OU entre les experts : car un réseau expert est activé parmi les K réseaux experts pour chaque vecteur d'entrée présenté.

L'apprentissage est indépendant : deux apprentissages différents dans le réseau glial pour deux sorties différentes. L'apprentissage, pour avoir la réponse désirée du réseau glial, est réalisé d'une façon indépendante de l'apprentissage pour le contrôle des experts. Les experts n'interviennent pas et n'influent pas pendant la phase d'apprentissage, donc les réponses lentes et rapides ne sont reliées que pour la phase d'utilisation/restitution.

L'apprentissage est également coopératif : ce type d'apprentissage ne concerne que le 2<sup>ème</sup> algorithme. Les réseaux experts apprennent en même temps et se répartissent la tâche au cours de l'apprentissage.

Il y a également une forme d'apprentissage supervisé: un réseau glial contrôle et supervise les réseaux experts et coordonne les activités de l'ensemble de l'architecture.

#### 4.4 Apprentissage du réseau Neuroglial

Nous avons proposé deux algorithmes différents pour l'adaptation de ce réseau :

- 1<sup>er</sup> Algorithme

Considérons l'architecture du réseau Neuroglial représentée par la figure 4.6.

Chaque réseau expert est connecté à une seule partie du vecteur d'entrée. Par exemple la 1<sup>ère</sup> partie connectée au 1<sup>er</sup> réseau, la 2<sup>ème</sup> partie connectée au 2<sup>ème</sup> réseau... etc. Le réseau glial, suivant la partie d'entrée présentée, active le réseau expert correspondant et désactive les autres réseaux experts.

Le réseau glial sélectionne les réseaux experts sans tester les performances d'adaptation de chaque réseau. Il n'y a pas de compétition entre les réseaux experts.

Tous les réseaux experts et réseau glial sont des réseaux perceptron monocouche.

l : Désigne lent

r : Désigne rapide

K : Nombre experts

k : Désigne itération

1-Répartition de l'espace d'entrée rapide en  $K$  parties  $X_1, \dots, X_K$

2-Initialiser les poids synaptiques des réseaux experts et réseau glial

3 -Calculer pour  $i = 1, 2, \dots, K$  et pour  $m = 1, 2, \dots, q$

Pour chaque partie d'entrée  $X_{li}$

$$y_{li}^m = X_{li}^T W_{li}^m(k)$$

$$e_{li}^m(k) = d_l^m(k) - y_{li}^m(k)$$

$$W_{li}^m(k+1) = W_{li}^m(k) + \eta e_{li}^m(k) X_{li}$$

$$y_{ri}^m = X_{ri}^T W_{ri}^m(k)$$

$$y_{ri}(k) = [y_{ri}^1, y_{ri}^2, \dots, y_{ri}^q]^T$$

$$e_{ri}^m(k) = d_r^m(k) - y_{ri}^m(k)$$

$$w_{ri}^m(k+1) = w_{ri}^m(k) + \eta e_{ri}^m(k) X_{ri}$$

Fin de  $m$

4 - Répéter l'étape 3 pour toutes les données d'apprentissage

5 - Répéter l'étape 3 et 4 jusqu'à la convergence de l'algorithme.

Si  $X = X_i$  alors  $g_i = 1$

Sinon  $g_i = 0$

Fin de  $i$

$$y_r = \sum_{i=1}^K g_i y_{ri}$$

$$y = y_{li} + y_r$$

Table 4.1: Le 1<sup>er</sup> algorithme de l'architecture Neurogliale

▪ 2<sup>ème</sup> Algorithme :

Considérons l'architecture du réseau Neuroglial représentée par la figure 4.6. Chaque vecteur d'entrée présenté est connecté à tous les réseaux experts. Les  $K$  réseaux experts apprennent des exemples différents de la base d'apprentissage. Ils sont mis en compétition pour chaque vecteur d'entrée présenté.

Chaque réseau rapide devient un expert. Les différents experts se spécialisent pour un groupe de réponses parmi l'ensemble des réponses possibles. Le réseau dont la réponse est la plus proche de la réponse désirée (c'est-à-dire le gagnant de la compétition) reçoit plus de poids, alors que les autres reçoivent très peu. Le réseau lent pondère les sorties des experts de tel sorte que pour une entrée donnée, l'expert ayant la plus grande chance de fournir la bonne réponse obtient le poids le plus élevé (pour notre application le poids le plus élevé =1).

L'approche Neurogliale, en utilisant le 2<sup>ème</sup> algorithme, présente certaines similitudes avec l'architecture des mélanges d'experts développée par Jacob [24] [22], concernant la supervision et la sélection des experts. On peut citer 4 différences :

- La première différence

Dans l'architecture du mélange d'experts le superviseur et les experts ont les mêmes entrées [9] [7]. Par contre dans notre architecture, elles sont différentes.

- Deuxième différence

La tâche du réseau superviseur dans l'approche de Jordan [24] [22] supervise et contrôle la compétition des experts, alors que dans notre approche le réseau glial supervise, contrôle la compétition des experts et en plus donne des informations : c'est la réponse du réseau glial (la réponse lente). Sachant que dans notre application on doit combiner deux réponses (lente et rapide) pour avoir la réponse globale du système.

- Troisième différence

Les poids de sélection des experts doivent être binaire 0 ou 1 par contre dans Jacob les poids sont des probabilités entre (0,1).

- Quatrième différence

Notre approche s'adapte bien aux systèmes présentant deux échelles de temps, c'est-à-dire les systèmes qui peuvent être décomposés en sous système lent et sous système rapide, comme les systèmes singulièrement perturbés. Par contre, l'approche des mélanges d'experts présente une seule réponse.

L'apprentissage des experts et le réseau glial se font simultanément en suivant les étapes suivantes :

1. Choisir le vecteur d'entrée  $X_r$ .
2. Faire apprendre le réseau glial pour obtenir la réponse lente désirée correspondant au vecteur d'entrée choisi.
3. Le choix d'expert se fait sous forme de probabilité d'avoir l'expert  $i$  sachant l'entrée  $X_r$   $p(i / X_r)$ .

4. La sortie de l'expert  $i$  représente la moyenne conditionnelle de la réponse désirée sachant l'entrée et le réseau expert.

#### 4.4.1 L'apprentissage des experts (réseaux rapides)

Supposons que chaque réseau expert est représenté par la (figure 4.7)

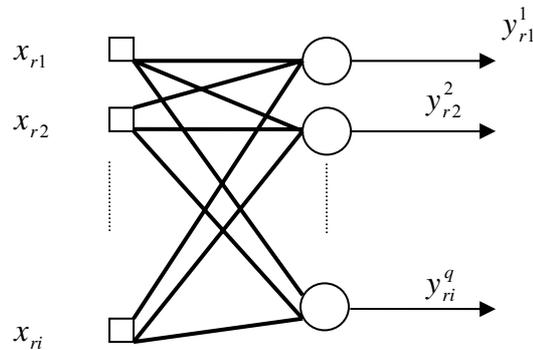


Figure 4.7 : Un réseau expert

On suppose que la dimension du vecteur d'entrée  $X_r$  est  $p$  et celle de la sortie est  $q$ , chaque expert  $i$  « apprend » une fonction de  $\mathfrak{R}^p$  dans  $\mathfrak{R}^q$ .

$$X_r \mapsto Y_{ri}(X_r) = F(x_r, w_{ri})$$

Un expert est un perceptron monocouche avec un neurone de sortie de type linéaire.

D'après la (figure 4.7), la sortie du neurone  $m$  est donnée par :

$$Y_{ri}^m = X_r^T w_{ri}^m \quad (4.1)$$

ou  $X_r = [x_1, x_2, \dots, x_p]^T$  et  $W_{ri}^m = [w_{i1}^m, w_{i2}^m, \dots, w_{ip}^m]^T$

Notre but est de déterminer les poids  $W_{ri}^m$  qui maximisent la fonction  $L(W, g)$ .

Ces poids seront calculés récursivement selon l'équation suivante :

$$W_{ri}^m = W_{ri}^m(k) + \Delta W_{ri}^m(k) \quad (4.2)$$

ou

$$\Delta W_{ri}^m(k) = \eta \frac{\partial L(W, g)}{\partial W_{ri}^m(k)} \quad (4.3)$$

$\eta$  : Le pas d'apprentissage.

Pour le calcul de  $\frac{\partial L(W, g)}{\partial W_{ri}^m(k)}$ , on a :

$$\frac{\partial L}{\partial W_{ri}^m} = \frac{\partial L}{\partial y_{ri}^m} \frac{\partial y_{ri}^m}{\partial W_{ri}^m} \quad (4.4)$$

Après des simplifications, nous obtenons :

$$\frac{\partial L}{\partial y_{ri}^m} = h_i(d_r^m - y_{ri}^m) \quad (4.5)$$

D'après l'équation ( 4.1 ), nous avons :

$$\frac{\partial y_{ri}^m}{\partial W_{ri}^m} = X_r \quad (4.6)$$

En utilisant les équations (4.2) à (4.6) et en posant  $e_{ri}^m = d_r^m - y_{ri}^m$ , nous aurons :

$$W_{ri}^m(k+1) = W_{ri}^m(k) + \eta h_i e_{ri}^m(k) X_r \quad (4.7)$$

#### 4.4.2 L'apprentissage du réseau glial

On considère le réseau glial représenté par la (figure 4.8).

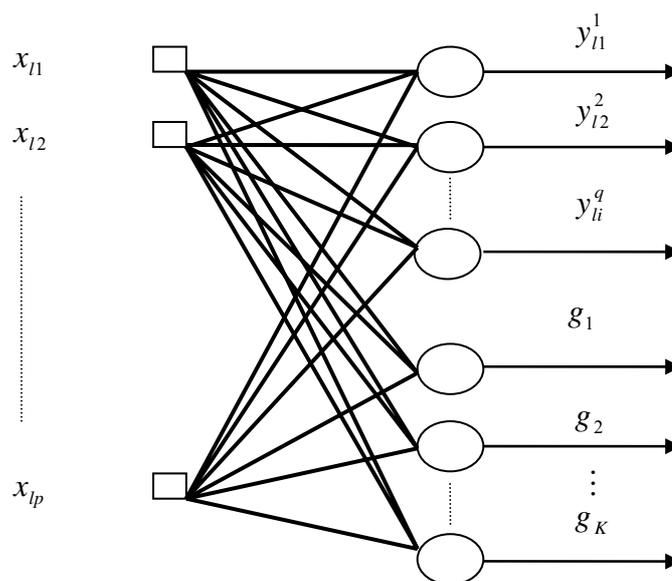


Figure 4.8 : Réseau glial

Le réseau glial est un réseau à couche unique avec deux vecteurs de sortie : Vecteur de sélection avec  $K$  neurones et le vecteur de la réponse lente avec  $q$  neurones. Chacun des  $K$  neurones étant assigné à un expert. Ces derniers sont non-linéaires, leur sortie est scalaire et les  $q$  sorties sont linéaires.

On suppose que la dimension du vecteur d'entrée  $X_l$  est  $p$ . Celle de la sortie du vecteur de sélection est  $K$  et celle la sortie de la réponse est  $q$ .

Le réseau glial définit deux fonctions de  $\mathfrak{R}^p$  dans  $\mathfrak{R}^q$  et de  $\mathfrak{R}^p$  dans  $\mathfrak{R}^K$ .

$$X_l \mapsto Y_{li}(X_l) = F(x_l w_{li})$$

$$X_l \mapsto g_i(X_{li}) = F(x_{li}, a_{li})$$

▪ Apprentissage du vecteur d'activation :

Soit  $a_i = [a_{i1}, a_{i2}, \dots, a_{ip}]^T$  le vecteur poids du  $i^{\text{ème}}$  neurone du réseau glial.

$i = 1, 2, \dots, K$

D'après la (figure 4.8), on a :

$$u_i = X_l^T a_i \quad (4.8)$$

La  $i^{\text{ème}}$  sortie du réseau glial représente l'estimation de la probabilité que le  $i^{\text{ème}}$  expert soit activé sachant l'entrée. Pour cela et puisque ces probabilités sont de somme égale à 1, on utilise une fonction « exponentielle normalisée » aussi appelée « softmax ».

$$g_i = \frac{\exp(u_i)}{\sum_j^K \exp(u_j)} \quad (4.9)$$

Le vecteur  $u_i$  est alors transformé par la fonction « softmax » afin de fournir des valeurs positives, de somme égale à 1 :

Les sorties  $g_i$  du réseau glial représentent les probabilités conditionnelles, à priori, que les sorties des experts estiment correctement une solution désirée  $d_r$ , pour un vecteur  $X_l$  connu. Pour ces probabilités la fonction (4.9) doit satisfaire les conditions suivantes :

$$\begin{cases} 0 \leq g_i \leq 1 \\ \sum_{i=1}^K g_i = 1 \end{cases} \quad (4.10)$$

La sortie globale  $y_r$  est donnée par la combinaison linéaire des sorties des  $K$  experts du réseau Neuroglial.

$$y_r = \sum_{i=1}^K g_i y_{ri} \quad (4.11)$$

Chaque sortie  $y_{ri}$  est pondérée par la sortie  $g_i$  du réseau glial qui lui est associé au même instant. Ces pondérations sont une solution optimale d'une fonction d'erreur. Dans l'hypothèse où chaque exemple  $(x_r, y_r)$  de la base d'apprentissage soit généré par un et un seul expert, les coefficients  $g_i$  seraient devenus binaires au cours de l'apprentissage.

Dans ce cas, la sortie globale du réseau rapide correspond à la sortie de l'un des experts du réseau et la valeur moyenne a un sens. (Comme dans notre application).

- Fonction de coût

La phase d'apprentissage permet de déterminer les poids de chaque expert et du réseau glial en maximisant une fonction coût désignée par le logarithme de la vraisemblance suivante :

$$L(W, g) = \ln \sum_{i=1}^k g_i \exp\left(-\frac{1}{2} \|d_r - y_{ri}\|^2\right) \quad (4.12)$$

Pour l'approximation de cette fonction, une probabilité, à posteriori, est introduite pour chaque expert :

$$h_i = \frac{g_i \exp\left(-\frac{1}{2} \|d_r - y_{ri}\|^2\right)}{\sum_{j=1}^k g_j \exp\left(-\frac{1}{2} \|d_r - y_{rj}\|^2\right)} \quad (4.13)$$

Les probabilités  $h_i$  sont conditionnelles. Elles doivent satisfaire les conditions suivantes :

$$\begin{cases} 0 \leq h_i \leq 1 \\ \sum_{i=1}^K h_i = 1 \end{cases}$$

L'adaptation se fait récursivement selon les équations suivantes :

$$a_i(k+1) = a_i(k) + \Delta a_i(k) \quad (4.14)$$

$$\Delta a_i(k) = \eta \frac{\partial L(W, g)}{\partial a_i(k)} \quad (4.15)$$

ou

$$\frac{\partial L}{\partial a_i} = \frac{\partial L}{\partial u_i} \frac{\partial u_i}{\partial a_i} \quad (4.16)$$

Après des simplifications nous avons :

$$\frac{\partial L}{\partial a_i} = h_i - g_i \quad (4.17)$$

$$\frac{\partial u_i}{\partial a_i} = X_i \quad (4.18)$$

$$a_i(k+1) = a_i(k) + \eta(h_i(k) - g_i(k))X_i \quad (4.19)$$

▪ Apprentissage de la réponse

L'apprentissage du réseau glial, permettant l'obtention de la réponse lente, n'est pas influencé par l'apprentissage du vecteur de sélection. Le réseau glial est un perceptron monocouche. L'adaptation de ce réseau pour avoir la réponse désirée se fait en utilisant l'algorithme du perceptron monocouche. A chaque exemple dans la base d'apprentissage, l'erreur instantanée entre la réponse désirée  $d_i$  et la réponse estimée  $y_i$  est définie par :

$$\varepsilon(k) = d(k) - y_i(k) \quad (4.20)$$

L'algorithme du perceptron consiste à ajuster les poids selon la règle suivante :

$$W_i(k+1) = W_i(k) + \mu \varepsilon(k) X_i(k) \quad (4.21)$$

Où  $\mu$  représente le taux d'apprentissage (« Learning rate »)  $0 < \mu < 1$ .

A la présentation de  $k^{ième}$  exemple, si la réponse désirée et la réponse estimée sont les même, les poids sont inchangés. Sinon les poids seront modifiés.

La procédure est répétée pour tous les exemples d'apprentissage.

## Algorithme d'apprentissage du réseau Neuroglial

1-Initialiser les poids synaptiques des réseaux experts et réseau glial

2- Pour chaque partie d'entrée  $X_{li}$

2.1- Calculer pour  $i=1,2,\dots, k$  et  $m= 1,2,\dots, q$

$$u_i(k) = X_i^T a_i(k)$$

$$g_i(k) = \frac{\exp(u_i(k))}{\sum_{j=1}^k \exp(u_j(k))}$$

$$y_{ri}^m = X_r^T W_{ri}^m(k)$$

$$y_{ri}(k) = [y_i^1, y_i^2, \dots, y_i^q]^T$$

$$h_i(k) = \frac{g_i(n) \exp\left(\frac{-1}{2} \|d_r - y_{ri}\|^2\right)}{\sum_{j=1}^k g_j \exp\left(\frac{-1}{2} \|d_r - y_{ri}\|^2\right)}$$

$$e_{ri}^m(k) = dr^m - y_{ri}^m(k)$$

$$W_{ri}^m(k+1) = W_{ri}^m(k) + \eta h_i(k) e_{ri}^m(k) X_r$$

$$a_i(k+1) = a_i(k) + \eta (h_i(k) - g_i(k)) X_i$$

$$y_{li}^m = X_{li}^m W_{li}^m(k)$$

$$e_{li}^m(k) = d_l^m(k) - y_{li}^m(k)$$

$$W_{li}^m(k+1) = W_{li}^m(k) + \eta e_{li}^m(k) X_l$$

2.2-Répéter l'étape 2.1 pour toutes les données d'apprentissage.

2.3-Répéter 2.1 à 2.2 jusqu'à la convergence de l'algorithme.

Si  $\max(g_i)$  alors  $g_i = 1$

Sinon  $g_i = 0$

Fin  $i$

$$y_r = \sum_{i=1}^K g_i y_{ri}$$

$$y = y_l + y_r$$

Table 4.2 : 2<sup>ème</sup> algorithme de l'architecture Neurogliale

#### 4.5 Conclusion

Selon la complexité et les caractéristiques physiques du problème à résoudre, nous avons décomposé le modèle en un sous modèle lent et un sous modèle rapide.

Le puissant concept de la modularité permet de fournir de l'information dans la structure du réseau de neurones, sans complexifier ni son architecture ni celle du neurone lui-même.

Nous avons présenté notre démarche et les références sur lesquelles nous sommes appuyés pour proposer une architecture Neurogliale intelligente, modulaire et autonome.

Deux algorithmes ont été traités pour l'adaptation de cette structure :

- Le 1<sup>er</sup> algorithme consiste à faire activer automatiquement, selon l'entrée présentée, un expert parmi tous les experts sans tester leurs performances.
- En outre, le 2<sup>ème</sup> algorithme évalue les performances des experts. Il consiste à sélectionner les experts, en activant l'expert le plus performant pour une entrée donnée.

Afin de valider nos idées, nous prévoyons de les mettre en œuvre par simulation. Le domaine d'application qui nous concerne présente deux échelles de temps, ce sont les systèmes singulièrement perturbés. En raison de ses caractéristiques, nous avons choisi la machine asynchrone.

## **CHAPITRE 5**

### **APPLICATION DU RESEAU NEUROGLIAL AU MODELE DE LA MACHINE ASYNCHRONE ET SIMULATION**

#### 5.1 Introduction

Dans ce chapitre, nous allons valider une nouvelle approche modulaire inspirée récemment de la biologie, appelée Neurogliale. La mise on œuvre est basée sur les modèles réduits de la machine asynchrone.

Les équations des machines asynchrone sont non linéaires et fortement couplées. Cependant, ces modèles peuvent être considérés à plusieurs échelles de temps (dynamiques électriques, électromagnétiques et mécaniques, hormis l'échelle de temps thermique).

Afin d'en réduire la complexité et de faciliter l'analyse et l'élaboration de stratégies du contrôle de ces machines, nous proposons d'utiliser des connaissances, à priori, en découplant le système en deux sous systèmes, en utilisant la techniques des perturbations singulières. Le calcul des modèles réduits suppose, au préalable, un regroupement adéquat des modes lent et rapide des systèmes (système modélisé sous la forme singulièrement perturbée).

Plutôt que d'utiliser un seul réseau de neurones pour simuler la machine asynchrone, nous avons choisi de développer une architecture modulaire contenant plusieurs sous réseaux.

Le choix d'utiliser l'architecture modulaire Neurogliale est une réponse aux contraintes liées aux problèmes engendrés par la complexité de la tâche à accomplir (dimensionnalité et aux problèmes de non-linéarité).

Les modèles que nous allons utiliser vont respectivement estimer le sous système lent et le sous système rapide.

Dans un premier temps, nous allons présenter le modèle de la machine asynchrone ainsi qu'une procédure itérative qui utilise l'approche par les cercles de

Gerschgorin pour permettre de séparer les modes lents et rapides, suivi par l'application de la technique des perturbations singulières pour l'obtention d'une forme standard singulièrement perturbée.

Dans un deuxième temps, nous présentons plusieurs résultats de tests de différentes architectures modulaires effectués sur les modèles réduits de la machine asynchrone.

- Architecture modulaire à deux réseaux de neurones.
- L'architecture Neurogliale.  
Deux types de réseaux gliaux sont développés :
- Réseau glial automatique.
- Réseau glial basé sur une fonction softmax.

Dans le but d'avoir une erreur optimale, une meilleure vitesse de convergence et une configuration neuronale réduite, une étude comparative des performances serait effectuée.

En effet :

- Etude comparative entre le modèle de la machine à réseau de neurones unique et le modèle modulaire à deux réseaux de neurones.
- Etude comparative entre le modèle modulaire à deux réseaux de neurones et le modèle Neuroglial\_MAS basé sur le 1<sup>er</sup> algorithme.
- Enfin, une étude comparative entre le modèle Neuroglial\_MAS basé sur le 1<sup>er</sup> algorithme et le modèle Neuroglial\_MAS basé sur le 2<sup>ème</sup> algorithme.

## 5.2 Mise sous forme standard de la machine asynchrone

### 5.2.1 Les modes électromagnétiques

Les expressions mathématiques décrivant le comportement dynamique de la machine asynchrone peuvent être mises sous la forme d'une équation différentielle généralisée :

$$\begin{aligned}
 [U] &= [R][I] + [L] \frac{d}{dt} [I] & (5.1) \\
 [U] &= [v_{ds} \quad v_{qs} \quad v_{dr} \quad v_{qr}]^T \\
 [I] &= [i_{ds} \quad i_{qs} \quad i_{dr} \quad i_{qr}]^T
 \end{aligned}$$

Mise sous forme d'équation d'état, cette équation devient :

$$\frac{d}{dt}[I] = [A][I] + [B][U] \quad (5.2)$$

avec :  $[A] = -[L]^{-1}[R]$ , et  $[B] = [L]^{-1}$

Soit la matrice d'évolution :

$$[A] = \begin{bmatrix} [A_{11}] & [A_{12}] \\ [A_{21}] & [A_{22}] \end{bmatrix}$$

Pour laquelle :

$$[A_{11}] = \begin{bmatrix} -\frac{1}{\sigma T_s} & \omega_s + \frac{1-\sigma}{\sigma} \omega \\ -\left(\omega_s + \frac{1-\sigma}{\sigma} \omega\right) & -\frac{1}{\sigma T_s} \end{bmatrix} = -\frac{1}{\sigma T_s} I_2 - \left(\omega_s + \frac{1-\sigma}{\sigma} \omega\right) J_2$$

$$[A_{12}] = \begin{bmatrix} \frac{M}{\sigma L_s T_r} & \frac{M}{\sigma L_s} \omega \\ -\frac{M}{\sigma L_s} \omega & \frac{M}{\sigma L_s T_r} \end{bmatrix} = \frac{M}{\sigma L_s T_r} I_2 - \frac{M}{\sigma L_s} \omega J_2$$

$$[A_{21}] = \begin{bmatrix} \frac{M}{\sigma L_r T_s} & \frac{M}{\sigma L_r} \omega \\ -\frac{M}{\sigma L_r} \omega & \frac{M}{\sigma L_r T_s} \end{bmatrix} = \frac{M}{\sigma L_r T_s} I_2 - \frac{M}{\sigma L_r} \omega J_2$$

$$[A_{22}] = \begin{bmatrix} -\frac{1}{\sigma T_r} & \left(\omega_s - \frac{1}{\sigma} \omega\right) \\ -\left(\omega_s - \frac{1}{\sigma} \omega\right) & -\frac{1}{\sigma T_r} \end{bmatrix} = -\frac{1}{\sigma T_r} I_2 - \left(\omega_s - \frac{1}{\sigma} \omega\right) J_2$$

et la matrice d'application des entrées :

$$[B] = \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \\ -\frac{1}{\sigma L_s L_r} & 0 \\ 0 & -\frac{1}{\sigma L_s L_r} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sigma L_s} I_2 \\ -\frac{M}{\sigma L_s L_r} I_2 \end{bmatrix}$$

avec :

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, J_2 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

et  $J_2$  une matrice de notation de  $90^\circ$ .

### 5.2.2 Modèle d'état des flux de la machine en $\alpha, \beta$

En utilisant les équations (5.1) et (5.2), le modèle d'état s'écrit :

$$\frac{d}{dt} \begin{bmatrix} \phi_{s\alpha\beta} \\ \phi_{r\alpha\beta} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sigma T_s} I_2 & \frac{B_r}{\sigma T_s} R(-\theta) \\ \frac{1-\sigma}{\sigma T_r B_r} R(\theta) & -\frac{1}{\sigma T_r} I_2 \end{bmatrix} \begin{bmatrix} \phi_{s\alpha\beta} \\ \phi_{r\alpha\beta} \end{bmatrix} + \begin{bmatrix} v_{s\alpha\beta} \\ 0 \end{bmatrix} \quad (5.3)$$

avec :

$$B_r = \frac{M}{L_r}, [R(\theta)] = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \quad \text{et} \quad \sigma = \frac{1-M^2}{L_s * L_r}.$$

et l'expression du couple électromagnétique devient :

$$C_{em} = P \frac{B_r}{\sigma L_s} \phi_{s\alpha\beta}^T R(-\theta) J_2 \phi_{r\alpha\beta} \quad (5.4)$$

avec :

$$\begin{aligned} R(\theta) \cdot R(-\theta) &= I_2 \\ R(\theta) \cdot J_2 \cdot R(-\theta) &= J_2 \\ R(\theta) \cdot R(-\theta) &= -\omega \cdot J_2 \end{aligned}$$

### 5.2.3 Séparation des modes électromagnétiques

a)-L'application de l'algorithme des cercles de Gerschgorin à la matrice d'état :

$$A = \begin{bmatrix} -\frac{1}{T_{sp}} I_2 & \frac{B_r}{T_{sp}} R(-\theta) \\ \frac{1-\sigma}{T_{rp} B_r} R(\theta) & -\frac{1}{T_{rp}} I_2 \end{bmatrix}. \quad (5.5)$$

avec  $T_{sp} = \sigma T_s$ ,  $T_{rp} = \sigma T_r$

donne :

1. Les cercles sont doublés et centrés en  $C_{1,2} = -\frac{1}{T_{sp}}$  et  $C_{3,4} = -\frac{1}{T_{rp}}$ , et leurs rayons sont respectivement  $R_{1,2} = \frac{B_r}{T_{sp}} (|\cos(\theta)| + |\sin(\theta)|)$ , et  $R_{3,4} = \frac{1-\sigma}{B_r T_{rp}} (|\cos(\theta)| + |\sin(\theta)|)$ .
2. Pour toutes les valeurs de  $R_{1,2}$  et  $R_{3,4}$ , les cercles se coupent.

b)-Pour éliminer les coefficients  $B_r$  et  $R(\theta)$ , on utilise la transformation :

$$\phi_1 = P_1 \begin{bmatrix} \phi_{s\alpha\beta} \\ \phi_{r\alpha\beta} \end{bmatrix}, P_1 = \begin{bmatrix} I_2 & 0 \\ 0 & B_r R(-\theta) \end{bmatrix} \quad (5.6)$$

avec :

$$A_1 = P_1 A P_1^{-1}$$

Soit :

$$A_1 = \begin{bmatrix} -\frac{1}{T_{sp}} I_2 & \frac{1}{T_{sp}} I_2 \\ \frac{1-\sigma}{T_{rp}} I_2 & -\frac{1}{T_{rp}} I_2 - \omega J_2 \end{bmatrix}. \quad (5.7)$$

1. Les cercles sont toujours doublés et centrés en  $C_{1,2} = -\frac{1}{T_{sp}}$  et  $C_{3,4} = -\frac{1}{T_{rp}}$ , et leurs rayons sont respectivement  $R_{1,2} = \frac{1}{T_{sp}}$  et  $R_{3,4} = \frac{1-\sigma}{T_{rp}} + \omega$ .

d)-On peut déplacer les centres des cercles doublés par la transformation :

$$\phi_2 = P_2 \phi_1, P_2 = \begin{bmatrix} I_2 & 0 \\ -I_2 & I_2 \end{bmatrix}, \quad (5.8)$$

qui donne :

$$A_2 = \begin{bmatrix} 0 & \frac{1}{T_{sp}} I_2 \\ -\frac{\alpha\sigma}{T_{sp}} I_2 + \omega J_2 & -\frac{\alpha+1}{T_{sp}} I_2 - \omega J_2 \end{bmatrix}.$$

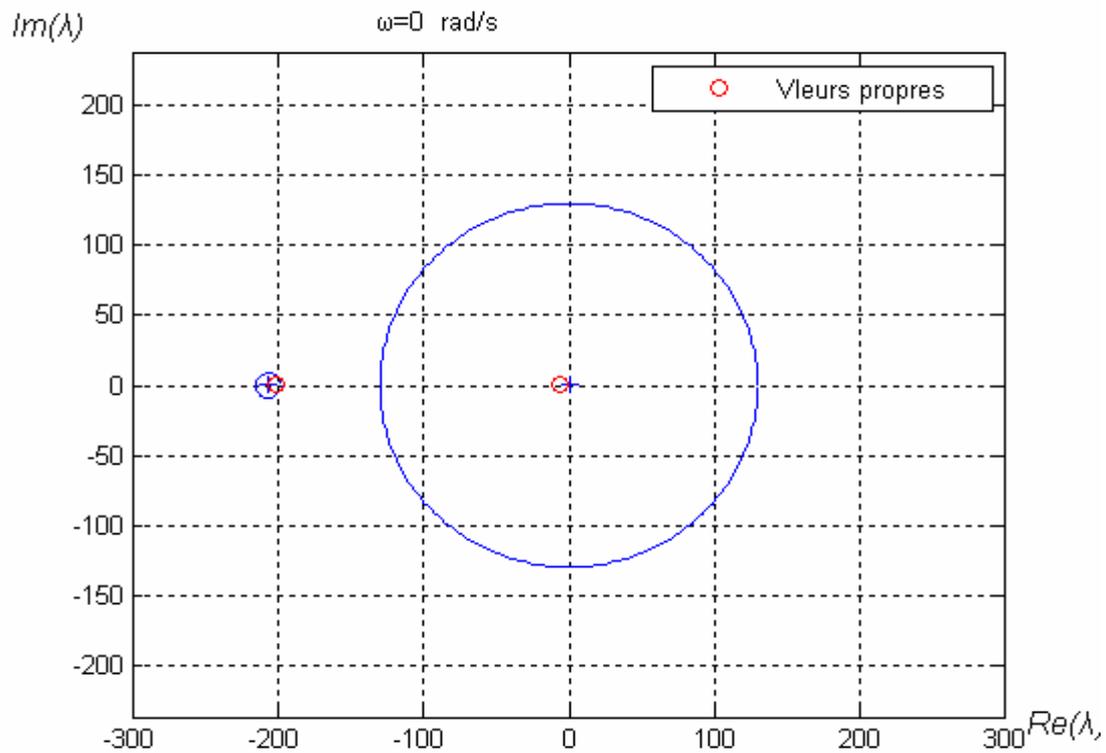
où

$$A_2 = \begin{bmatrix} 0 & 0 & \frac{1}{T_{sp}} & 0 \\ 0 & 0 & 0 & \frac{1}{T_{sp}} \\ -\frac{\alpha\sigma}{T_{sp}} & -\omega & -\frac{\alpha+1}{T_{sp}} & \omega \\ \omega & -\frac{\alpha\sigma}{T_{sp}} & -\omega & -\frac{\alpha+1}{T_{sp}} \end{bmatrix}. \quad (5.9)$$

avec  $\alpha = \frac{T_s}{T_r}$ .

On représente sur la figure (5.2), les cercles de Gerschgorin qui sont encore doublés et centrés en  $C_{1,2} = 0$  et  $C_{3,4} = -\frac{1+\alpha}{T_{sp}}$ . Leurs rayons sont respectivement  $R_{1,2} = \frac{1}{T_{sp}}$

et  $R_{3,4} = \frac{\alpha\sigma}{T_{sp}} + 2\omega$ .



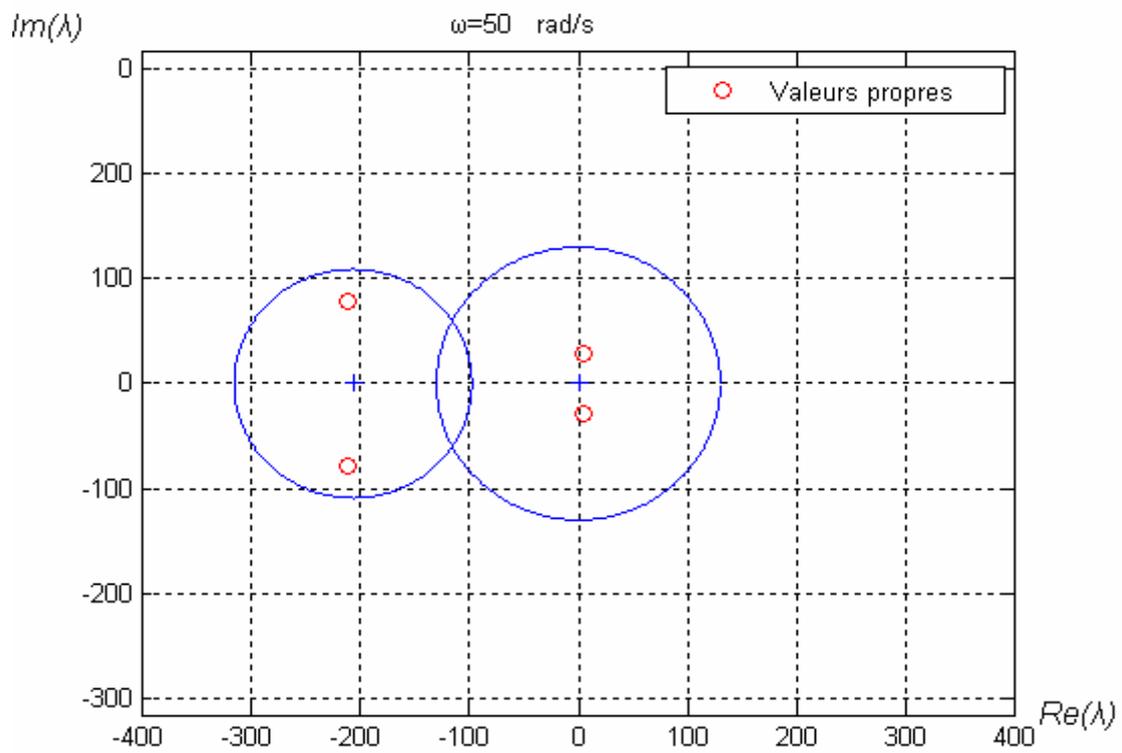
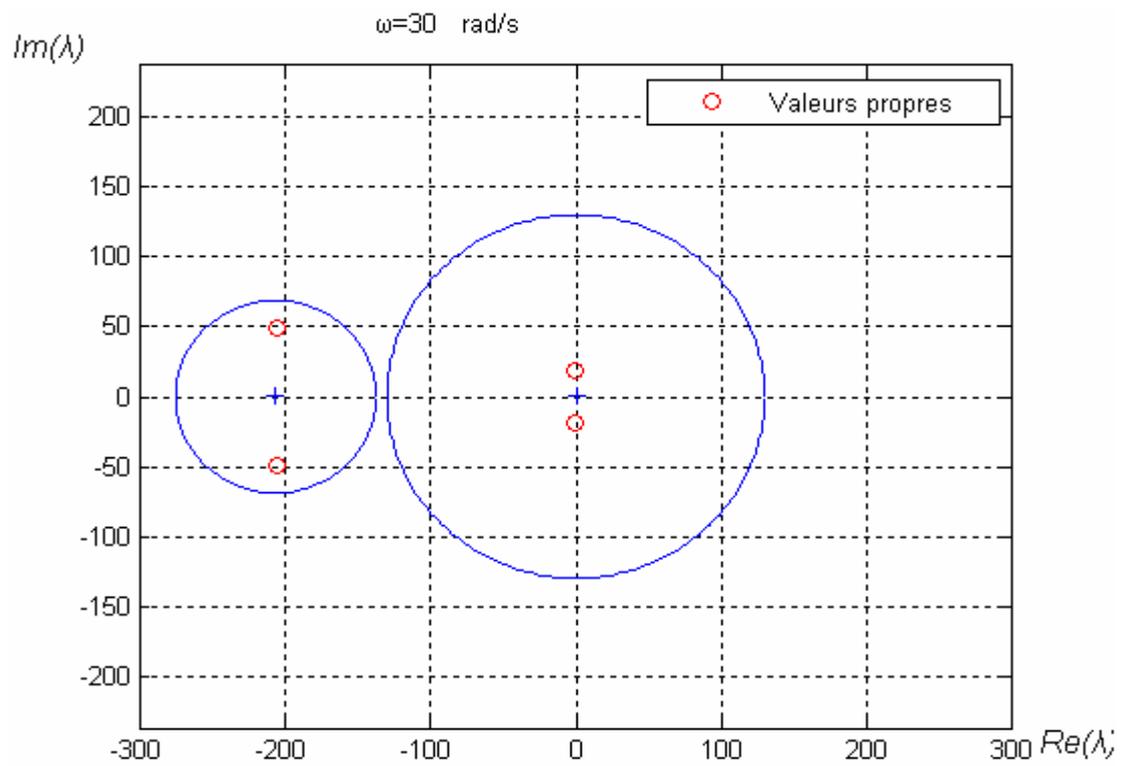


Figure 5.1 : Tracé des cercles de Gerschgorin pour la machine asynchrone pour différentes vitesses

### 5.2.4 Mise sous la forme standard singulièrement perturbée

Pour les petites vitesses, la transformation finale utilisée est :

$$\begin{bmatrix} x \\ z \end{bmatrix} = P_1 P_2 \begin{bmatrix} \phi_{s\alpha\beta} \\ \phi_{s\alpha\beta} \end{bmatrix}, \quad P_1 P_2 = \begin{bmatrix} I_2 & 0 \\ -I_2 & B_r R(-\theta) \end{bmatrix}, \quad (5.10)$$

et la nouvelle représentation d'état s'écrit :

$$\frac{d}{dt} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{T_{sp}} I_2 \\ -\frac{1}{T_{rp}} I_2 & -\frac{\alpha+1}{T_{sp}} I_2 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} I_2 & 0 \\ -I_2 & B_r R(-\theta) \end{bmatrix} \begin{bmatrix} v_{s\alpha\beta} \\ 0 \end{bmatrix}, \quad (5.11)$$

Le couple électromagnétique s'exprime par l'expression :

$$C_{em} = \frac{P}{L_{sp}} x^T J_2 z \quad (5.12)$$

Les modèles réduits sont calculés en posant :

$$\begin{aligned} x &= x_s(t) \\ z &= z_s(t) + z_f(\tau) \\ v_{s\alpha\beta} &= v_{s\alpha\beta(s)}(t) + v_{s\alpha\beta(f)}(\tau) \end{aligned} \quad (5.13)$$

#### 5.2.4.1. Modèles réduits de la machine asynchrone en repère $\alpha, \beta$

##### ▪ Modèle réduit lent

Ce modèle s'écrit :

$$\begin{cases} \frac{d}{dt} x_s = -\frac{1}{T_r(1+\alpha)} x_s + \frac{\alpha}{1+\alpha} v_{s\alpha\beta(s)} \\ Z_f = -\frac{T_{sp}}{1+\alpha} \left( \frac{1}{T_r} x_s + v_{s\alpha\beta(s)} \right) \\ C_{em} = \frac{P}{L_{sp}} x_s^T J_2 z_s \end{cases} \quad (5.14)$$

##### ▪ Modèle réduit rapide

Ce modèle s'exprime par :

$$\begin{cases} \frac{d}{dt} z_f(\tau) = -\frac{1+\alpha}{T_s} z_f - \sigma v_{s\alpha\beta}(f) \\ C_{em}(f) = \frac{P}{L_{sp}} x_s^T J_2 z_f \end{cases} \quad (5.15)$$

### 5.2.5 Modèles réduits en flux et en courants

On peut expliciter les expressions précédentes avec les flux et les courants.

1. D'après la relation (5.10), on écrit :

$$\begin{aligned}\phi_{s\alpha\beta} &= x \\ \phi_{r\alpha\beta} &= \frac{1}{B_r} R(\theta)(x + z)\end{aligned}\quad (5.16)$$

Les flux  $\phi_{s\alpha\beta}$  sont donc des variables lentes. Les flux  $\phi_{r\alpha\beta}$  peuvent être décomposés en des flux lents  $\phi_{r\alpha\beta l}$  et des flux rapides  $\phi_{r\alpha\beta r}$

Soit :

$$\phi_{s\alpha\beta} = \phi_{s\alpha\beta(s)} \quad (5.17)$$

$$\phi_{r\alpha\beta} = \phi_{r\alpha\beta(s)} + \phi_{r\alpha\beta(f)} \quad (5.18)$$

avec (5.13) et (5.16), l'équation (5.18) devient :

$$\phi_{r\alpha\beta} = \frac{1}{B_r} R(\theta) \left( \frac{1 + \alpha - \sigma\alpha}{1 + \alpha} x_s - \frac{T_{sp}}{1 + \alpha} v_{s\alpha\beta(s)} + z_f \right) \quad (5.19)$$

En identifiant les parties lentes et rapides, l'équation (5.18) permet d'écrire :

$$\phi_{r\alpha\beta(s)} = \frac{1}{B_r} R(\theta) \left( \frac{1 + \alpha - \sigma\alpha}{1 + \alpha} x_s - \frac{T_{sp}}{1 + \alpha} v_{s\alpha\beta(s)} \right) \quad (5.20)$$

$$\phi_{r\alpha\beta(f)} = \frac{1}{B_r} R(\theta) z_f \quad (5.21)$$

2) En utilisant la relation entre les flux et les courants de la machine asynchrone, il vient :

$$\begin{aligned}L_{sp} I_{s\alpha\beta} &= \phi_{s\alpha\beta} - B_r R(-\theta) \phi_{r\alpha\beta} \\ L_{rp} I_{r\alpha\beta} &= -\frac{1 - \sigma}{B_r} R(\theta) \phi_{s\alpha\beta} + \phi_{r\alpha\beta}\end{aligned}\quad (5.22)$$

En décomposant les courants en composantes lentes et rapides :

$$\begin{aligned}I_{s\alpha\beta} &= I_{s\alpha\beta(s)} + I_{s\alpha\beta(f)} \\ I_{r\alpha\beta} &= I_{r\alpha\beta(s)} + I_{r\alpha\beta(f)}\end{aligned}\quad (5.23)$$

On aboutit aux expressions suivantes :

$$\begin{aligned}L_{sp} I_{s\alpha\beta} &= \phi_{s\alpha\beta} - B_r R(-\theta) \phi_{r\alpha\beta(f)} \\ L_{rp} I_{r\alpha\beta} &= -B_r R(-\theta) \phi_{r\alpha\beta(f)}\end{aligned}\quad (5.24)$$

et :

$$\begin{aligned} L_{rp} I_{r\alpha\beta} &= -\frac{1-\sigma}{B_r} R(\theta) \phi_{s\alpha\beta} + \phi_{r\alpha\beta(s)} \\ L_{rp} I_{r\alpha\beta(f)} &= \phi_{r\alpha\beta(f)} \end{aligned} \quad (5.25)$$

Les sous systèmes (5.14) et (5.15) deviennent :

$$\frac{d}{dt} \begin{bmatrix} \phi_{s\alpha} \\ \phi_{s\beta} \end{bmatrix} = \begin{bmatrix} -\frac{1}{(1+\alpha)T_r} & 0 \\ 0 & -\frac{1}{(1+\alpha)T_r} \end{bmatrix} \cdot \begin{bmatrix} \phi_{s\alpha} \\ \phi_{s\beta} \end{bmatrix} + \frac{\alpha}{1+\alpha} \cdot \begin{bmatrix} v_{s\alpha(s)} \\ v_{s\beta(s)} \end{bmatrix} \quad (5.26)$$

$$C_{em(s)} = \frac{pT_s}{(1+\alpha)L_s} (\phi_{s\alpha} v_{s\beta(s)} - \phi_{s\beta} v_{s\alpha(s)}) \quad (5.27)$$

$$I_{s\alpha\beta(s)} = \frac{1}{L_{sp}} \left( \frac{\alpha\sigma}{1+\alpha} \phi_{s\alpha\beta(s)} + \frac{T_{sp}}{1+\alpha} v_{s\alpha\beta(s)} \right) \quad (5.28)$$

et :

$$\frac{d}{dt} \begin{bmatrix} I_{s\alpha(f)} \\ I_{s\beta(f)} \end{bmatrix} = \begin{bmatrix} -\frac{1+\alpha}{T_s} & 0 \\ 0 & -\frac{1+\alpha}{T_s} \end{bmatrix} \cdot \begin{bmatrix} I_{s\alpha(f)} \\ I_{s\beta(f)} \end{bmatrix} - \frac{1}{L_s} \cdot \begin{bmatrix} v_{s\alpha(f)} \\ v_{s\beta(f)} \end{bmatrix} \quad (5.29)$$

$$C_{em(f)} = p(\phi_{s\alpha} I_{s\beta(f)} - \phi_{s\beta} I_{s\alpha(f)}) \quad (5.30)$$

avec :

$$\begin{cases} I_{r\alpha\beta(s)} = \frac{1}{(1+\alpha)B_r L_r} R(\theta) (\phi_{s\alpha\beta(s)} - T_s v_{s\alpha\beta(s)}) \\ I_{r\alpha\beta(f)} = -\frac{L_s}{M} R(\theta) I_{s\alpha\beta(f)} \end{cases} \quad (5.31)$$

$$\begin{cases} \phi_{r\alpha\beta(s)} = \frac{1}{B_r} R(\theta) \left( \left( 1 - \frac{\alpha\sigma}{1+\alpha} \right) \phi_{s\alpha\beta(s)} - \frac{T_{sp}}{1+\alpha} v_{s\alpha\beta(s)} \right) \\ \phi_{r\alpha\beta(f)} = -\frac{L_s}{B_r} R(\theta) I_{s\alpha\beta(f)} \end{cases} \quad (5.32)$$

### 5.2.6 Simulations et interprétations

Nous utilisons les modèles réduits (5.14) et (5.15) pour la simulation numérique. Les tensions d'alimentation sont considérées sinusoïdales, équilibrées et de valeurs efficaces de 220v.

Le tracé des couples des modèles réduits est représenté par la figure (5.2) et la figure (5.3) pour différentes valeurs initiales. On remarque que les variables rapides s'annulent pour  $t=0.2s$ , puis les couples globaux sont donnés par leurs composantes lentes. Une deuxième remarque concerne l'influence des valeurs initiales. Ainsi, le tracé du couple montre que les résultats de deuxième choix se rapprochent des résultats du modèle global mieux que ceux pour le premier choix.

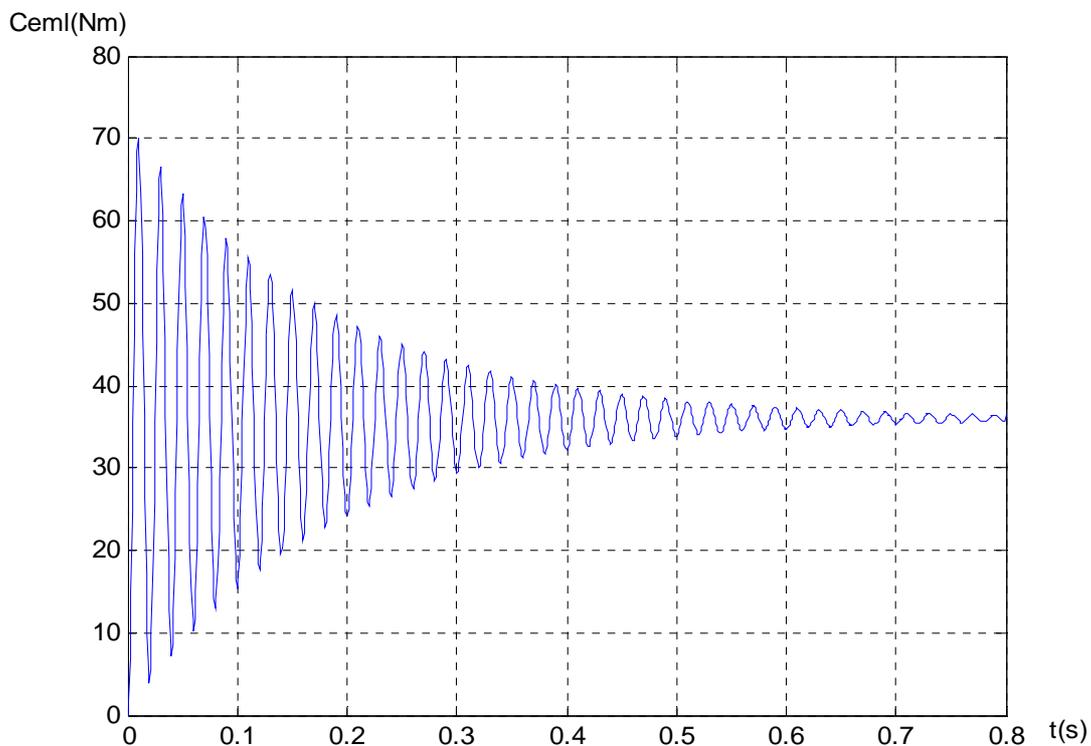


Figure 5.2.a : Couple électromagnétique lent

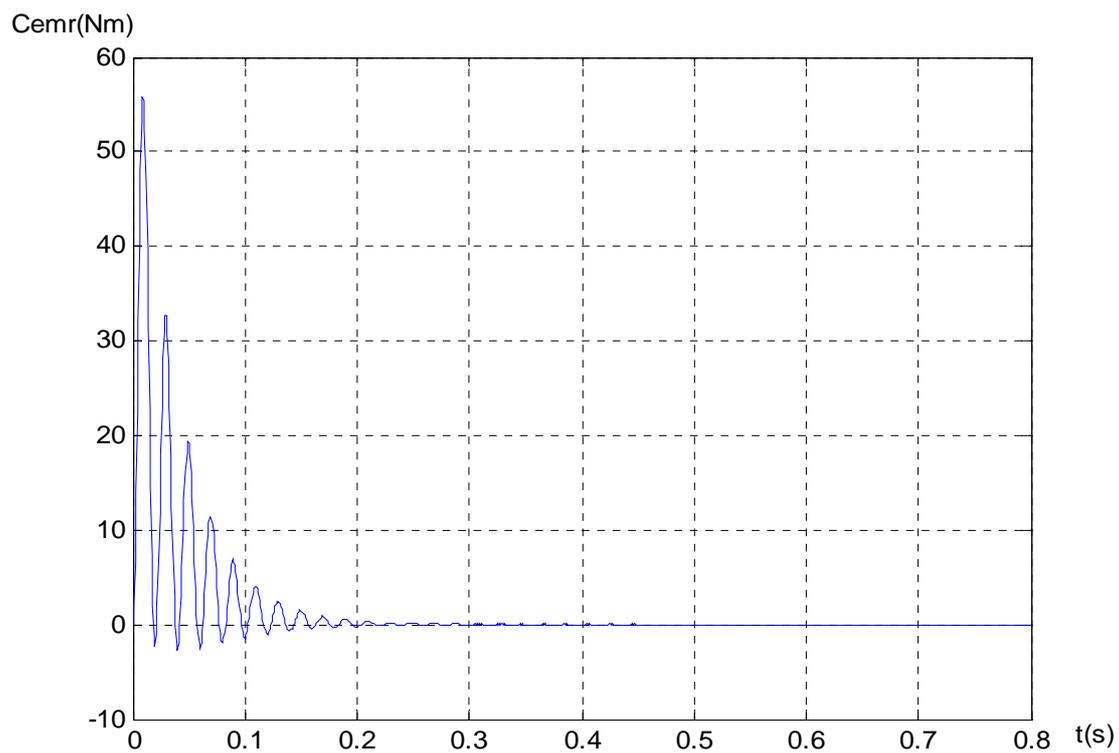


Figure 5.2.b : Couple électromagnétique rapide

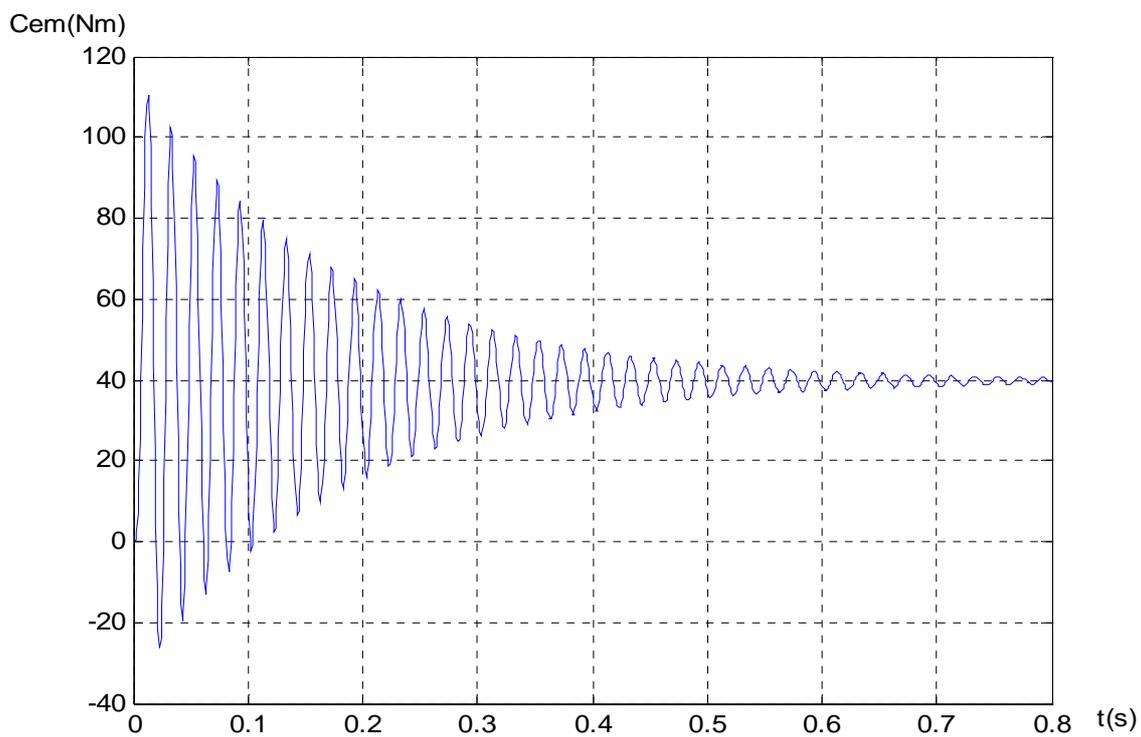


Figure 5.2.c : Couple électromagnétique

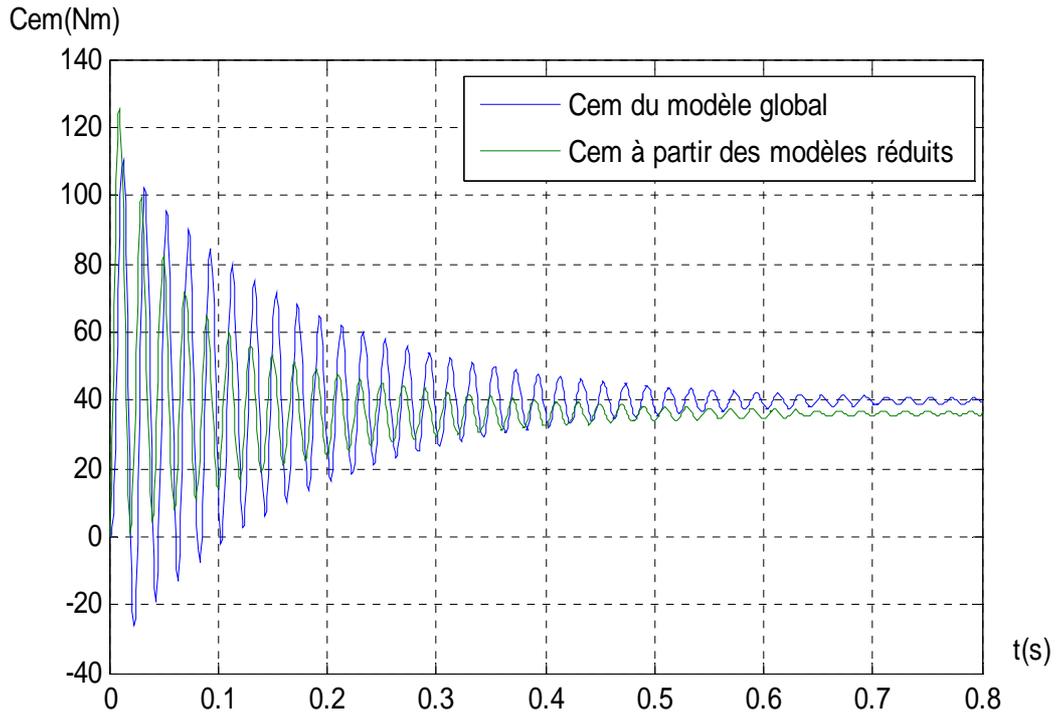


Figure 5.2.d : Comparaison du couple électromagnétique du modèle global et le couple électromagnétique simulé à partir des modèles réduits lent et rapide

Figure 5.2 : Evolution des couples électromagnétiques des modèles réduits et le couple du modèle global pour  $\Phi_{10} = [0 \ 0 \ 0 \ 0]$

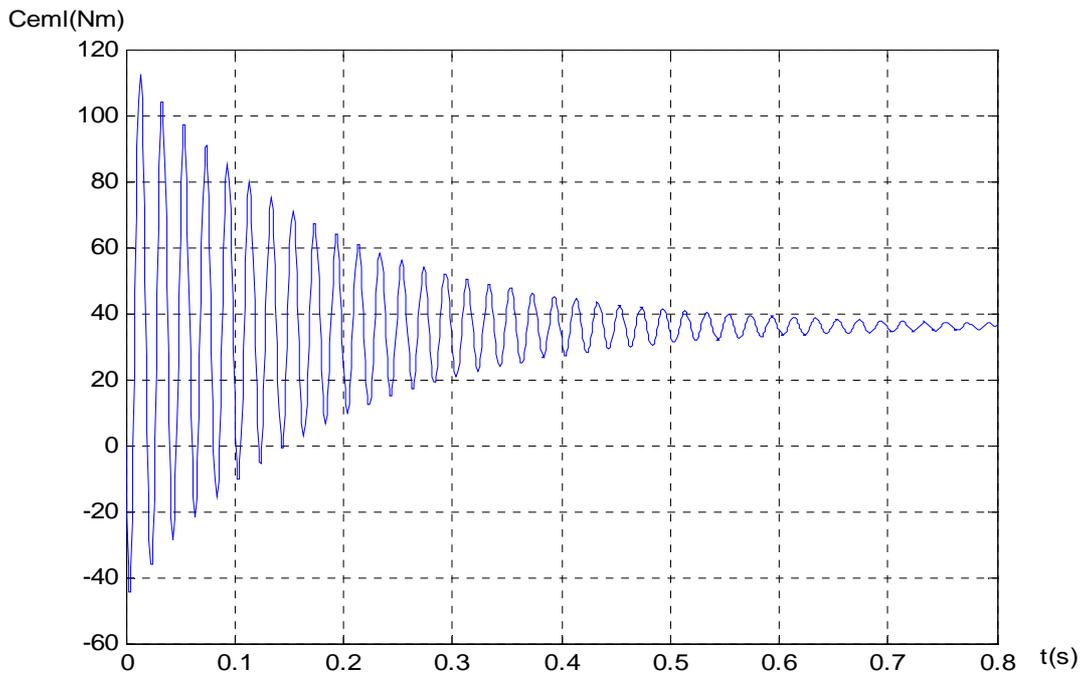


Figure 5.3.a : Couple électromagnétique lent

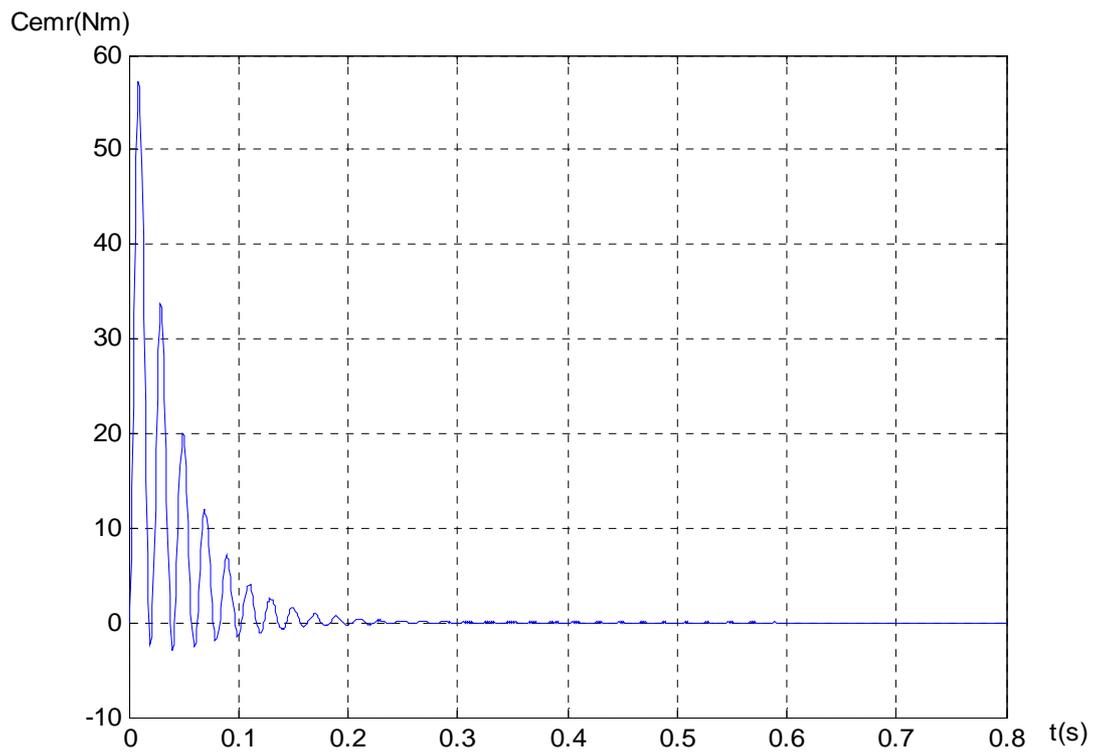


Figure 5.3.b Couple électromagnétique rapide

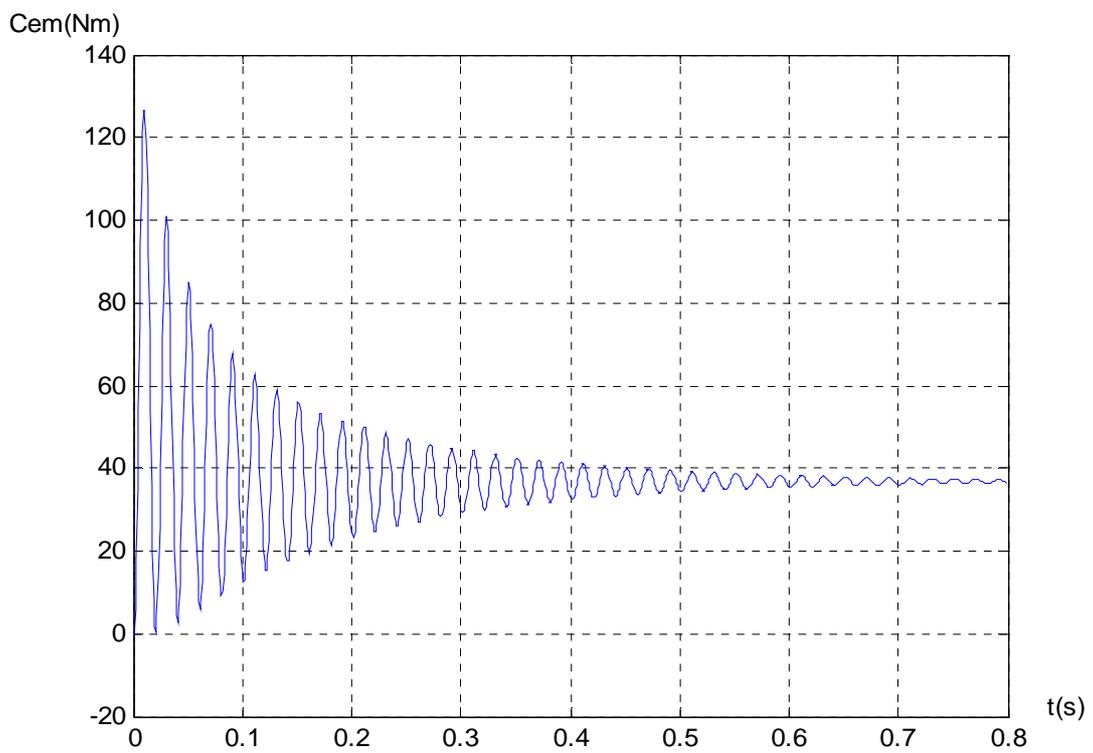


Figure 5.3.c : Couple électromagnétique

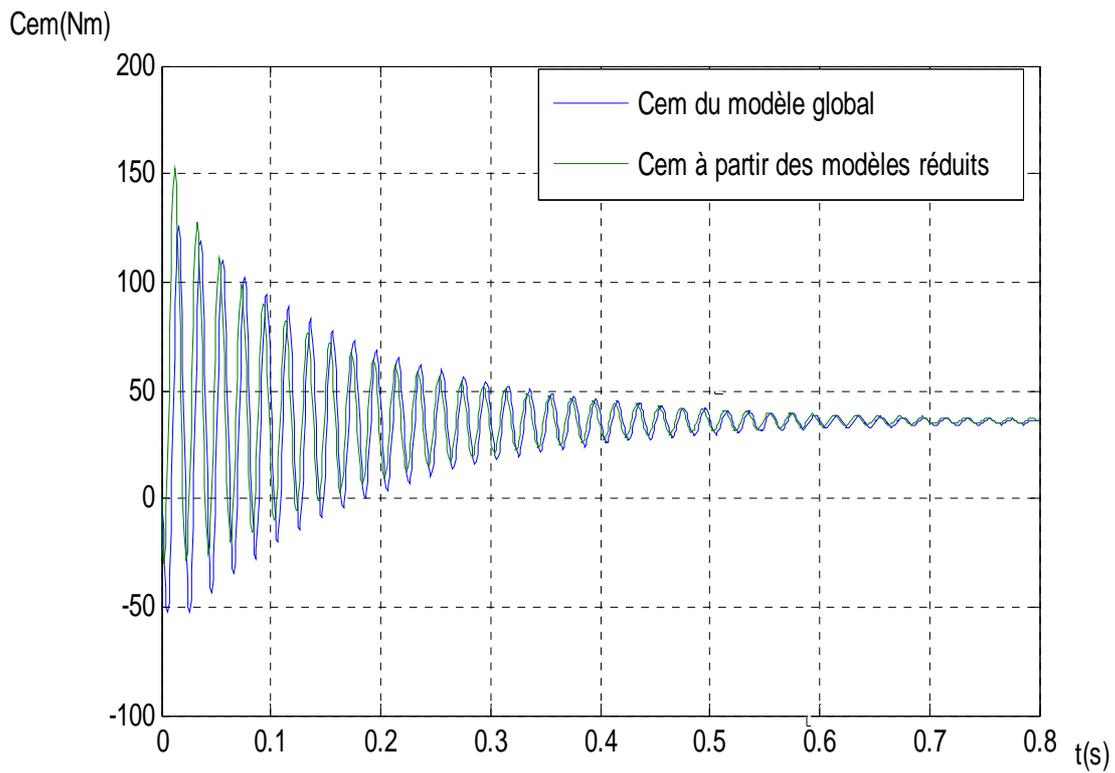


Figure 5.3.d : Comparaison le couple électromagnétique du modèle global et le couple électromagnétique simulé à partir des modèles réduits lent et rapide

Figure 5.3 : Evolution des couples électromagnétiques des modèles réduits et le couple du modèle global pour  $\Phi_{10} = [0 \ 0.9 \ 0 \ 0]$

## 5.3 Simulation

### 5.3.1 Réseau unique

En simulation, nous avons considéré une machine asynchrone triphasée à cage d'écureuil (voir appendice B) alimentée par un réseau alternatif 50(Hz) ayant une tension efficace de 220(V).

Dans cette section, nous allons apprendre le modèle de la machine asynchrone, que nous avons décrit plus haut. Le modèle est composé d'un réseau de neurones unique de type MLP. Dans le cadre de l'apprentissage, cette solution paraît être la plus simple.

- Choix des entrées / sorties

Dans le cas de l'utilisation d'un réseau de neurones unique, définir les entrées ne pose pas de problème. En effet, le vecteur d'entrée est composé de toutes les grandeurs qui participent au calcul de la sortie. Impérativement une mise en forme des entrées peut et doit être opérée. Chacune des entrées doit être normalisée. Cette normalisation peut être faite de diverses manières.

Dans cette application, le modèle à apprendre est fonction de deux entrées: les tensions sinusoïdales  $v_{s\alpha}$ ,  $v_{s\beta}$ . La sortie est le couple  $c_{em}$ .

- Architecture du réseau

Plusieurs essais ont été effectués en variant l'architecture du réseau (nombre de couches cachées, nombre de neurones dans la couche cachée, nombre d'exemples ...) et mêmes les algorithmes d'apprentissage. Pour chaque essai, nous testons les performances (phase de reconnaissance et test de généralisation).

En adoptant Les techniques citées dans le chapitre réseau de neurone avec augmentation progressive du nombre d'itérations et pour plusieurs architectures, nous avons abouti à une architecture permettant la convergence du réseau, avec :

Nombre de couches : 3

Taille des couches : 10 10 1

Taux d'apprentissage : 0.3

Nombre itérations : 5000

Poids initiaux aléatoires entre [-1.0, +1.0]

Normalisation des entrées [-1.0, +1.0]

Le nombre d'exemples est : 400.

Sachant que ces exemples ont été pris d'une simulation de la machine asynchrone.

Donc L'architecture du réseau est composée de plusieurs couches : une couche d'entrée contenant dix neurones; une couche cachée contenant elle-même dix neurones et une seule couche de sortie qui contient un seul neurone.

- Validation des résultats :

Pour évaluer les résultats obtenus par le modèle à réseau unique, la mesure de l'erreur moyenne quadratique EQM (Mean Square Error, MSE) a été utilisée :

$$EQM = \frac{1}{2} \sum_{i=1}^m (y_d - y_e)^2$$

- $m$  représente le nombre de sortie du réseau

- $y_d$  La sortie désirée

- $y_e$  La sortie estimée

- Les résultats d'apprentissage

En présentant au réseau 400 exemples, nous obtenons la convergence, après 5000 itérations pour le couple  $c_{em}$ , donnant ainsi, le résultat représenté à la figure (5.5).

Lors de l'apprentissage, on obtient la courbe d'erreur d'apprentissage et la courbe d'erreur moyenne quadratique suivantes :

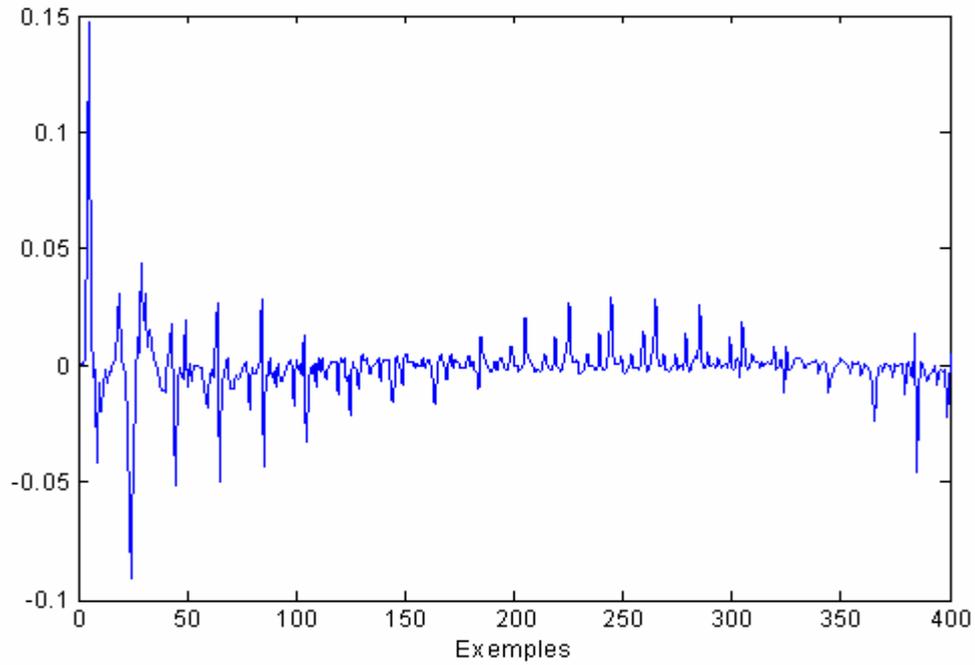


Figure 5.4 : Erreur d'apprentissage après 5000 itérations et de 400 exemples

Erreur maximale : 0.15

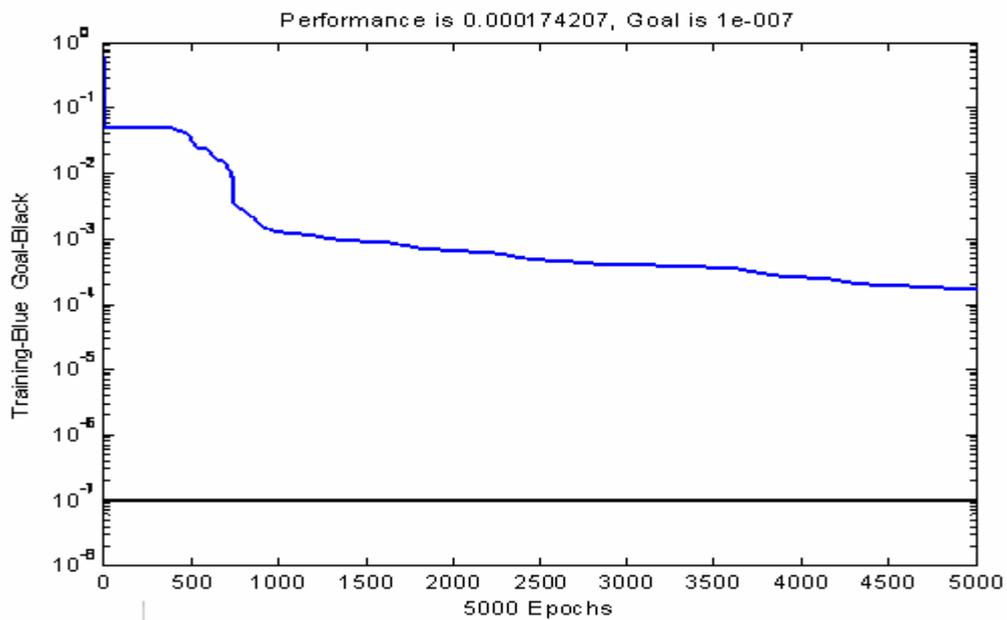


Figure 5.5 : Evolution d'EQM du réseau (phase d'apprentissage).

EQM du réseau MLP : 0.000174207

On obtient donc après 5000 itérations une erreur maximale de 0.15 et une erreur quadratique minimale de 0.000174207.

Nous avons effectué plusieurs expériences en augmentant le nombre d'itérations afin d'aboutir à des paramètres d'un réseau de neurones, qui peut donner des performances meilleures (tableau 5.1).

Tableau des différents tests :

N_iteration	EQM
100	0.0505282
500	0.0460538
1000	0.00142096
3000	0.000729612
4000	0.000174357
5000	0.000174207

Tableau 5.1 : Les performances du réseau unique MLP

Le tableau 5.1 montre que l'erreur quadratique diminue fortement à partir de la 3000<sup>ème</sup> itération, puis se stabilise un petit peu à partir de la 4000<sup>ème</sup> itération. Aussi on remarque, d'après le résultat de la figure 5.4 que la convergence est lente, du fait de la taille du réseau MLP.

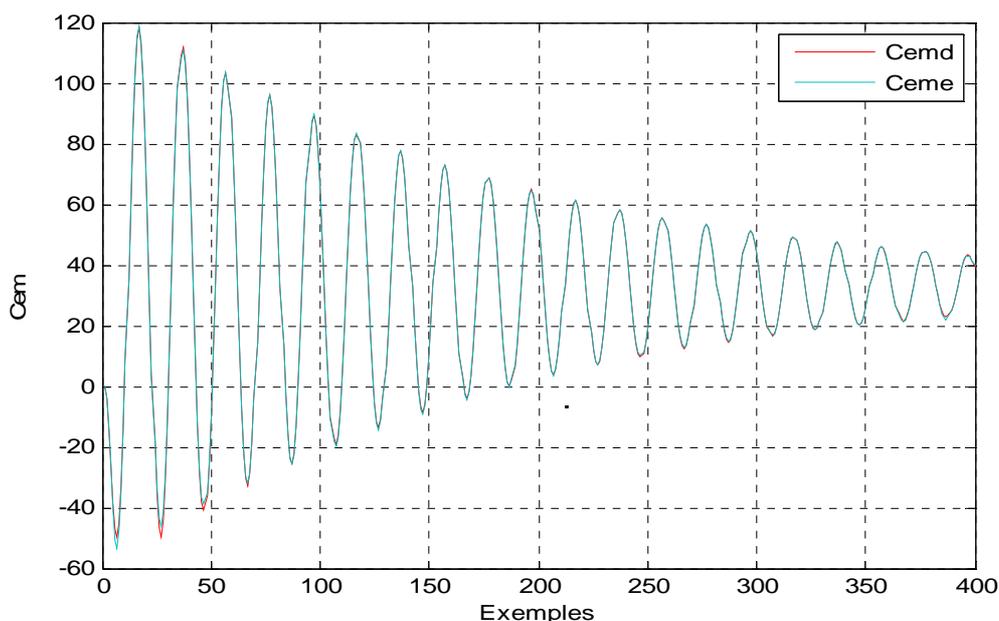


Figure 5.6 : Réponse estimée et désirée du modèle à réseau unique MLP

Pour palier aux problèmes de complexité et de convergence, aussi pour obtenir des meilleures performances avec une taille réduite, la solution présentée est l'emploi d'une architecture modulaire à deux réseaux : un réseau pour estimer le sous modèle réduit lent et un autre pour estimer le sous modèle réduit rapide, la tâche complexe va être décomposée en deux sous tâches plus petites et plus faciles à mettre en œuvre.

### 5.3.2 Apprentissage modulaire

Nous avons dit dans le chapitre précédent que la décomposition en modules est une solution alternative élégante qui permet de réduire la complexité d'une tâche à dimensionnalité élevée de l'espace d'entrée et en présence de non-linéarité.

Pour ne pas utiliser un réseau complexe, que nous ne savons pas forcément entraîner, nous allons plutôt utiliser deux réseaux de taille plus modeste, dont l'apprentissage est relativement bien maîtrisé. Un problème complexe va être décomposé en plusieurs sous-problèmes dans le but de réduire la dimensionnalité de chaque réseau de neurones. Les conséquences directes d'une dimensionnalité moindre sont une réduction du temps de traitement et de l'espace mémoire utilisé par les processeurs.

La décomposition que nous retenons pour simuler la machine est composée de deux modules :

- Un premier module est dédié à l'estimation du couple lent  $C_{eml}$  de la machine.
- Un second module est dédié à l'estimation du couple rapide  $C_{emr}$ .

Le schéma de décomposition utilisé est présenté à la figure 5.7.

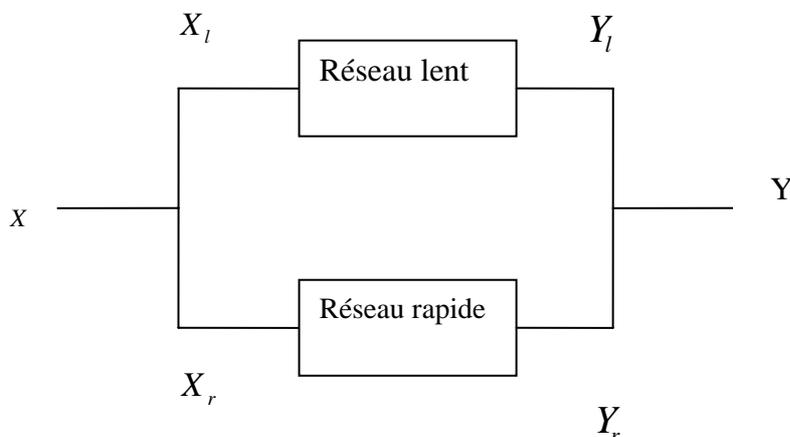


Figure 5.7 : Architecture modulaire à deux réseaux

Plusieurs étapes sont utilisées pour simuler ces deux modèles :

- 1- Présentation de la machine sous sa forme singulièrement perturbée en présentant les modèles réduits " lent et rapide "(paragraphe 5.2).
- 2- La décomposition des variables d'entrées : Les variables d'entrées  $v_\alpha, v_\beta$  aux variables lentes  $v_{\alpha l}, v_{\beta l}$  et aux variables rapides  $v_{\alpha r}, v_{\beta r}$ . Cette décomposition est réalisée en utilisant les ondelettes (appendice B).
- 3- L'apprentissage des deux réseaux " lent et rapide " .
- 4- La phase de reconnaissance des deux modèles " lent et rapide".
- 5- La combinaison des deux sorties  $c_{eml}, c_{emr}$  du réseau lent et rapide pour avoir la sortie globale  $c_{em}$  .

Nous allons maintenant présenter l'apprentissage de chacun des modules.

#### 5.3.2.1 Apprentissage du réseau lent

Le réseau lent apprend la relation entre l'espace d'entrée  $v_{\alpha l}, v_{\beta l}$  et la sortie qui est le couple lent  $c_{eml}$  de la machine (fig 5.7).

Comme pour le réseau unique vu précédemment, plusieurs essais ont été nécessaires pour pouvoir obtenir de bons résultats.

L'architecture qui a donné de meilleur résultat sur l'erreur est la suivante :

Nombre de couches : 2

Taille des couches : 4 1

Taux d'apprentissage : 0.3

Nombres itérations : 100

Poids initiaux aléatoires entre : [-1.0, +1.0]

Normalisation des entrées : [-1.0, +1.0]

Le nombre d'exemples est : 400.

D'autres tests ont été effectués en changeant les valeurs initiales des poids.

Lors de l'apprentissage, on obtient les courbes d'erreurs suivantes :

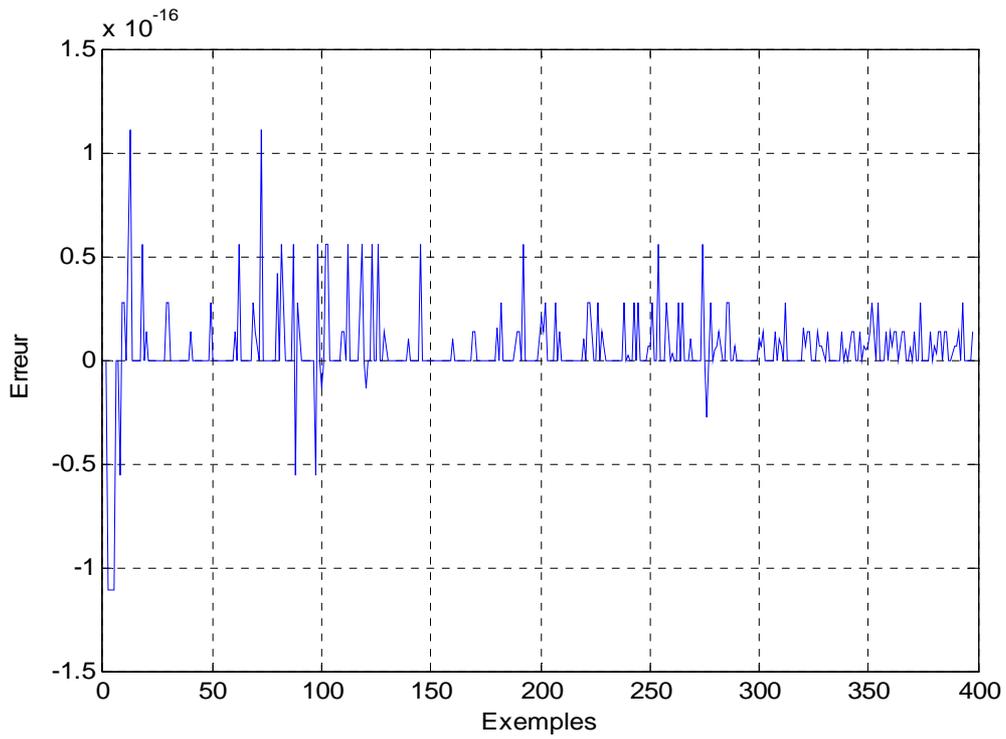


Figure 5.8 : Erreurs d'apprentissage après 100 itérations et 400 exemples

Erreur maximale : 1.11E-16

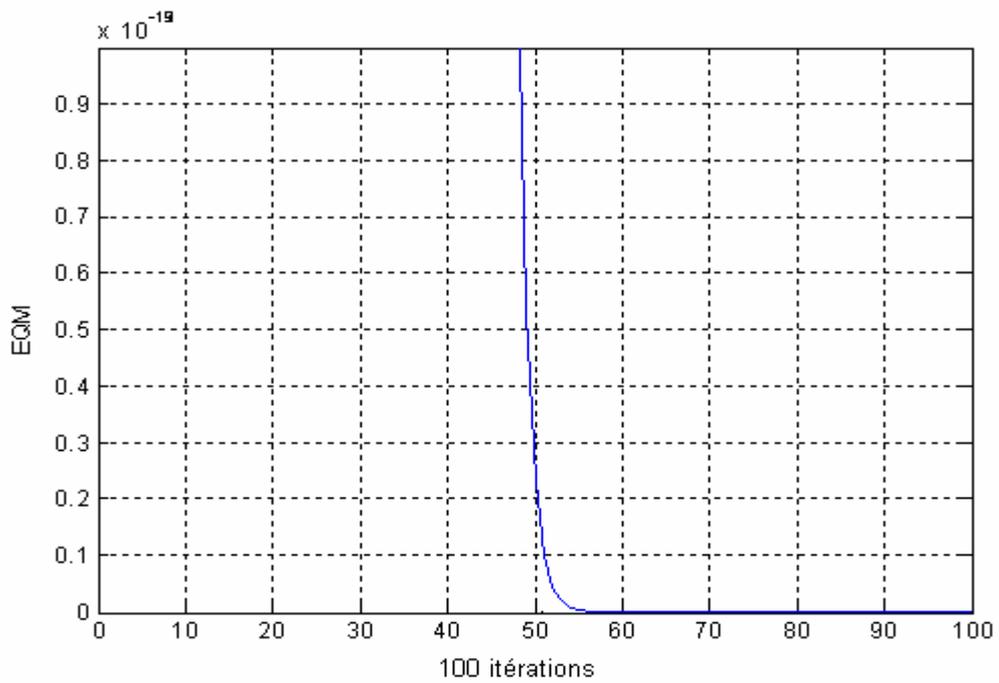


Figure 5.9 : Evolution d'EQM du réseau lent.

EQM : 7.23E-32

On remarque que la courbe d'erreur se stabilise à une valeur de :  $7.23E-32$ . Quelques 100 d'itérations sont suffisantes pour apprendre le modèle lent. Les sorties estimées sont comparées aux sorties réelles.

On remarque que l'erreur moyenne quadratique d'estimation est très proche de zéro (de l'ordre de  $7.32E-32$ ) et l'erreur maximale d'estimation est de l'ordre de ( $1.11E-16$ ). L'algorithme a convergé parfaitement et rapidement donnant ainsi, la réponse représentée par La figure 5.9.

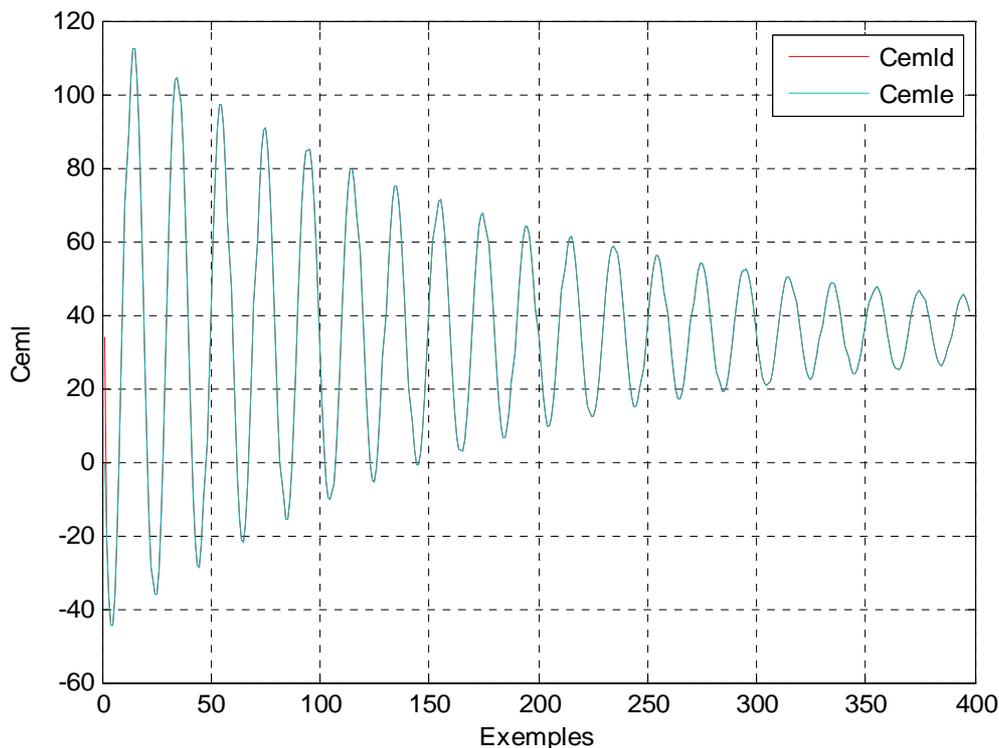


Figure 5.10 : La réponse estimée et désirée lente après 100 itérations et avec 400 exemples

Ces résultats ne sont pas exceptionnels. Ils s'expliquent par le fait que la fonction est quasi-linéaire. Les sorties de ce réseau lent, très satisfaisantes, vont être associées aux valeurs courantes des sorties du réseau rapide pour former le couple global qui représente la réponse de la machine.

### 5.3.2.2 L'apprentissage du réseau rapide

Ce réseau va apprendre le modèle réduit rapide de la machine. Ces entrées sont les deux entrées  $v_{\alpha r}$ ,  $v_{\beta r}$ , la sortie est le couple rapide  $c_{emr}$ .

Plusieurs tests sont effectués pour avoir une erreur minimale aussi proche que celle du réseau lent ou inférieur.

Différentes architectures ont été testées pour essayer d'apprendre le modèle réduit rapide de façon similaire à celle du modèle lent. Le meilleur résultat a été obtenu avec l'architecture suivante :

Nombre de couches : 2

Taille des couches : 4 1

Taux d'apprentissage : 0.3

Nombre d'itérations : 100

Poids initiaux aléatoires entre : [-1.0, +1.0]

Normalisation des entrées : [-1.0, +1.0]

Le nombre d'exemples est : 400.

Lors de l'apprentissage, on obtient les courbes d'erreurs suivantes :

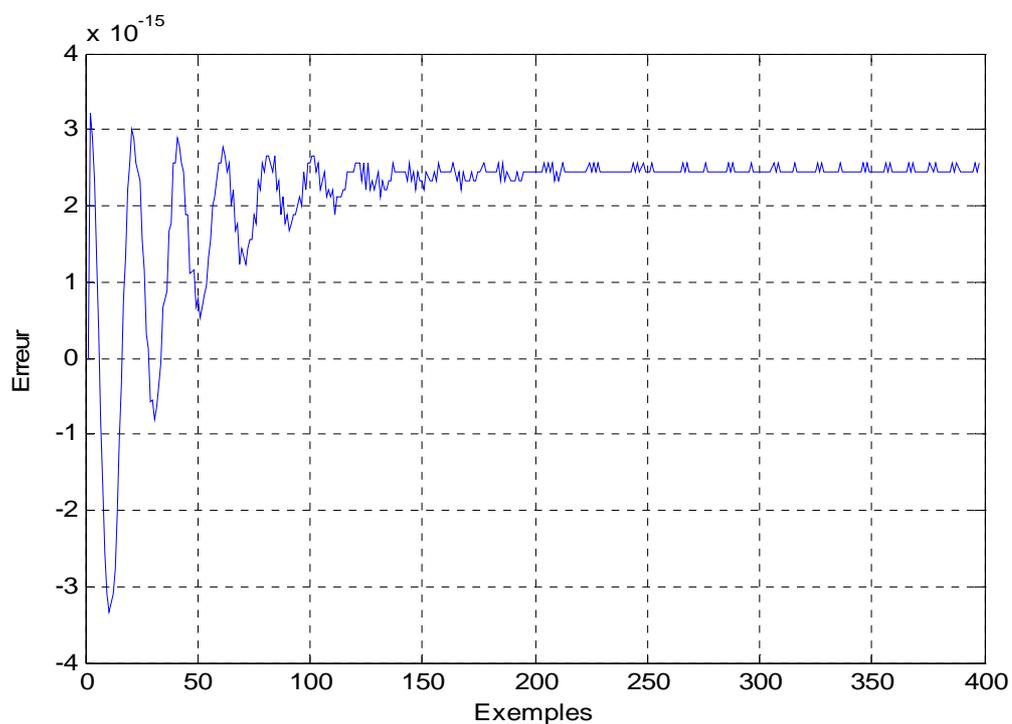


Figure 5.11 : Erreur d'apprentissage après 100 itérations et avec 400 exemples

Erreur maximale : 3.22E-15

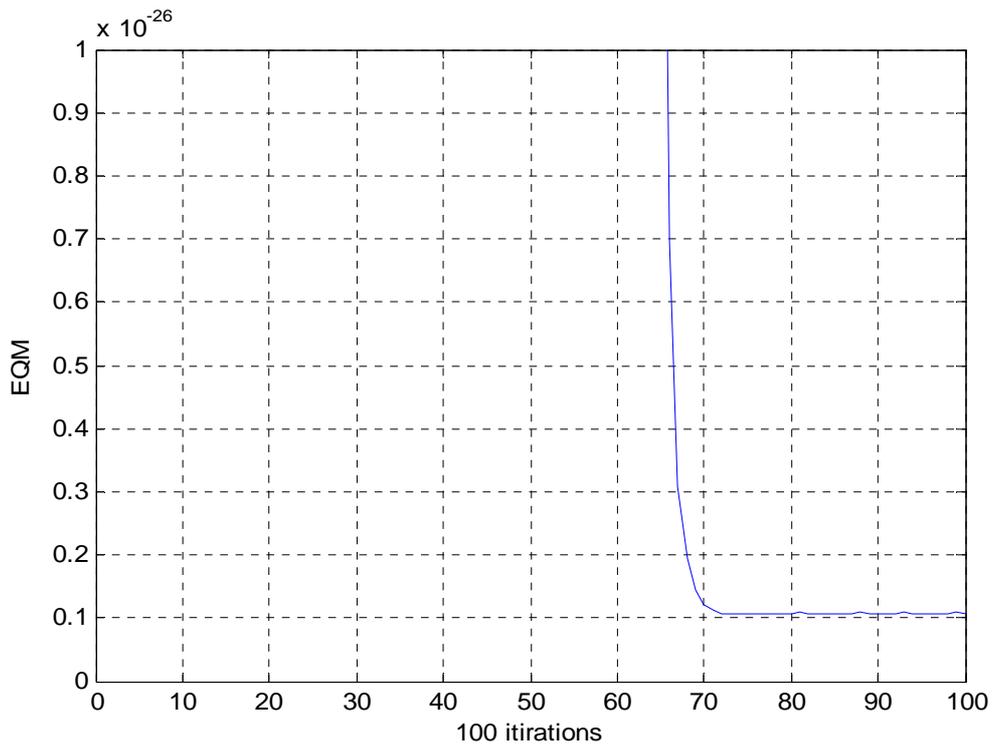


Figure 5.12 : Evolution d'EQM du réseau rapide.

EQM : 1.082E-27

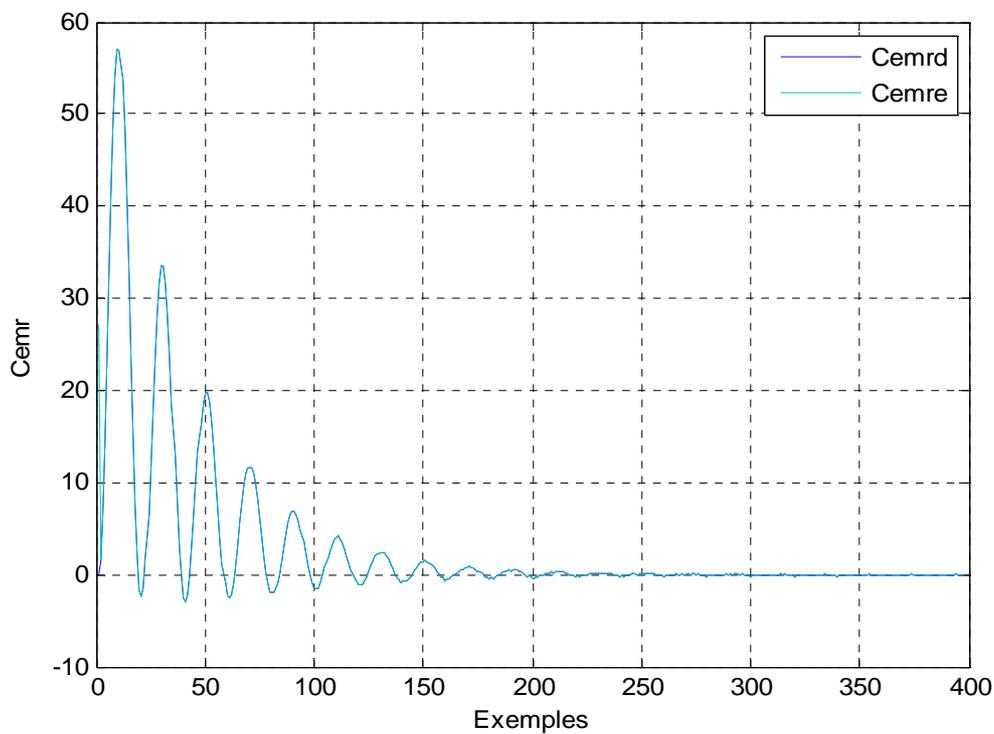


Figure 5.13 : La réponse estimée et désirée rapide après 100 itérations et de 400 exemples

On remarque que les erreurs maximales et quadratiques minimales sont très différentes et meilleures que le modèle à réseau de neurones unique, par contre elles sont moins performantes que celles du réseau lent.

### 5.3.2.3 Combinaison et comparaison

Le couple lent estimé va être associé au couple rapide estimé pour former le couple global estimé de la machine. Ainsi l'erreur d'estimation globale est la somme des erreurs d'estimation du modèle lent et celles du modèle rapide.

$$C_{eme} = C_{emle} + C_{emre}$$

$$e_g = e_l + e_r$$

Le couple global estimé est comparé au couple désiré de la machine (figure 5.14). L'erreur est de l'ordre de 1.08932E-27

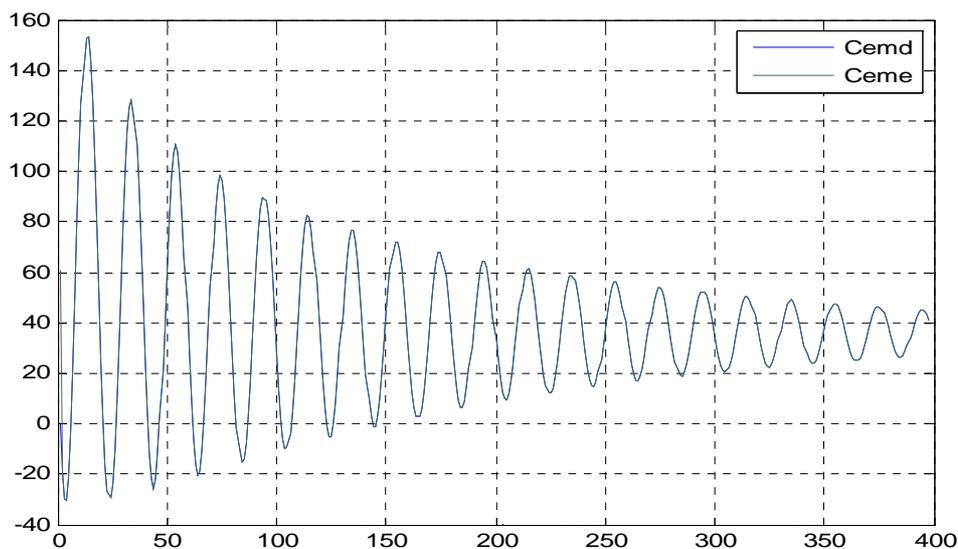


Figure 5.14 : La réponse globale

Ces résultats mettent en évidence les grands avantages offerts par le modèle modulaire à deux réseaux de neurones relativement au modèle à réseau unique. Ceci concernant d'un coté la vitesse de convergence et l'erreur minimale quadratique et d'autre coté, la complexité qui est considérablement réduite (taille de l'architecture réduite).

On remarque que la valeur de l'erreur globale est proche de celle du réseau rapide. Ceci s'explique par le fait que l'erreur obtenue par l'estimation du modèle lent, qui est de  $7.23E-32$ , est petite et négligeable devant celle du modèle rapide, qui est de l'ordre de  $1.082E-27$ .

Les performances du modèle global dépendent des performances du modèle réduit lent et celles du modèle réduit rapide.

De cela et pour améliorer les performances du modèle à deux réseaux de neurones, on est amené à utiliser une architecture modulaire plus sophistiquée pour estimer le couple de la machine asynchrone. Ceci nous a conduits à utiliser la nouvelle approche Neurogliale.

### 5.3.3 Réseau Neuroglial

Dans cette section, nous allons utiliser plusieurs réseaux pour estimer le modèle réduit rapide d'une machine asynchrone. Cette application va nous permettre de développer une architecture modulaire plus complexe mettant en œuvre un nombre de modules plus importants.

Nous allons considérer un réseau Neuroglial dont les experts sont des sous réseaux rapides et dont le réseau glial est un réseau lent. Après avoir appliqué différentes architectures neuronales classiques et modulaires pour simuler la machine asynchrone, nous présentons maintenant la mise en œuvre du réseau Neuroglial(NG) en utilisant deux algorithmes d'apprentissages différents pour voir les performances de chacun deux.

Dans un premier temps, nous décrirons le modèle de la machine asynchrone, le réseau Neuroglial, son architecture et plus particulièrement la séparation effectuée par l'algorithme du Neuroglial.

Dans un deuxième temps, nous effectuerons des comparaisons avec le même modèle et les mêmes données sur plusieurs cas d'études entre le modèle Neuroglial\_MAS et les autres modèles étudiés précédemment.

### 5.3.3.1 Architecture du modèle NG MAS

La précision du modèle neuronal dépend de son architecture, du nombre de données utilisées pour la détermination de ses poids et de la qualité des données utilisées pour son apprentissage. Il faut donc simplement déterminer une architecture suffisamment complexe pour bien représenter le problème. En règle générale, on choisit l'architecture la plus simple. La base d'apprentissage est très importante, elle permet de bien représenter la relation entre les tensions alternatives d'entrées et le couple.

Les paramètres d'apprentissage sont décrits ci dessous. De manière pratique, différentes architectures ont été testées. Dans un premier temps, nous avons fait varier le nombre d'experts (sous-réseaux rapides), afin de déterminer le nombre de régimes que pourrait détecter le modèle du réseau Neuroglial. Quelle que soit l'architecture des experts et du réseau glial testée, seulement quatre régimes ont été détectés. A partir de cette constatation, différentes architectures de réseaux experts et de réseaux gliaux ont été testées, de façon à déterminer l'architecture optimale.

Nous avons fait varier le nombre de neurones dans les couches d'entrées pour les quatre experts et le réseau glial.

L'architecture optimale est définie comme l'architecture la plus simple qui minimise en mieux l'erreur sur la base de test.

Nombre d'itérations : 100

Poids initiaux aléatoires entre : [-1.0, +1.0]

Normalisation des entrées : [-1.0, +1.0]

Pas d'apprentissage : 0.3

Le modèle Neuroglial\_MAS se compose de 4 réseaux experts et d'un réseau glial.

La couche d'entrée des experts et du réseau glial se compose de 4 neurones. Les quatre experts ont une architecture similaire, composée de deux couches, une couche d'entrée comportant quatre neurones et d'une couche de sortie comportant un neurone unique (Fonction d'activation linéaire), estimant la réponse rapide de la

machine. Le réseau glial est composé d'une couche d'entrée comportant quatre neurones et d'une couche de sortie comportant 5 neurones : les quatre neurones pour la sélection des experts, un pour chaque expert et un neurone pour l'estimation de la réponse lente.

Un certain nombre de tests comparatifs a été réalisé, plus précisément avec des réseaux Neurogliaux avec le même nombre d'experts, mais avec des réseaux gliaux différents. Deux algorithmes d'apprentissage différents ont été appliqués. Les performances de ces deux algorithmes sont basées sur l'erreur quadratique moyenne.

#### 5.3.3.2 1<sup>er</sup> algorithme d'apprentissage

Le réseau glial estime la réponse lente et sélectionne un expert automatiquement parmi tous les experts. Il lui affecte une pondération unité et met une pondération nulle à tous les autres réseaux experts pour chaque vecteur d'entrée donné.

La sortie globale du réseau rapide est alors égale à la somme des sorties de chaque expert pondérées par leurs coefficients de mélange et identique à l'une des sorties des experts.

$$y_r = g_1 y_1 + g_2 y_2 + g_3 y_3 + g_4 y_4$$

Le réseau expert sélectionné représente la réponse rapide correspondante. Le réseau glial est similaire au réseau lent du modèle à deux réseaux de neurones. Les performances sont identiques.

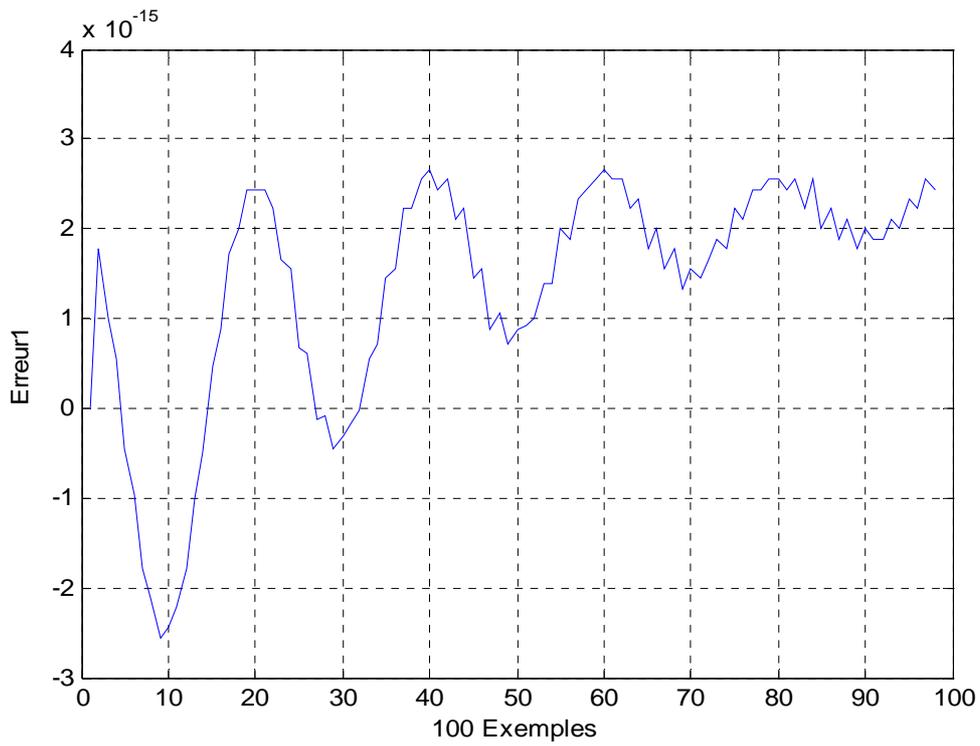


Figure 5.15 : Erreur d'apprentissage du réseau expert1 après 100 itérations et de 100 exemples

Erreur maximale : 2.44E-15

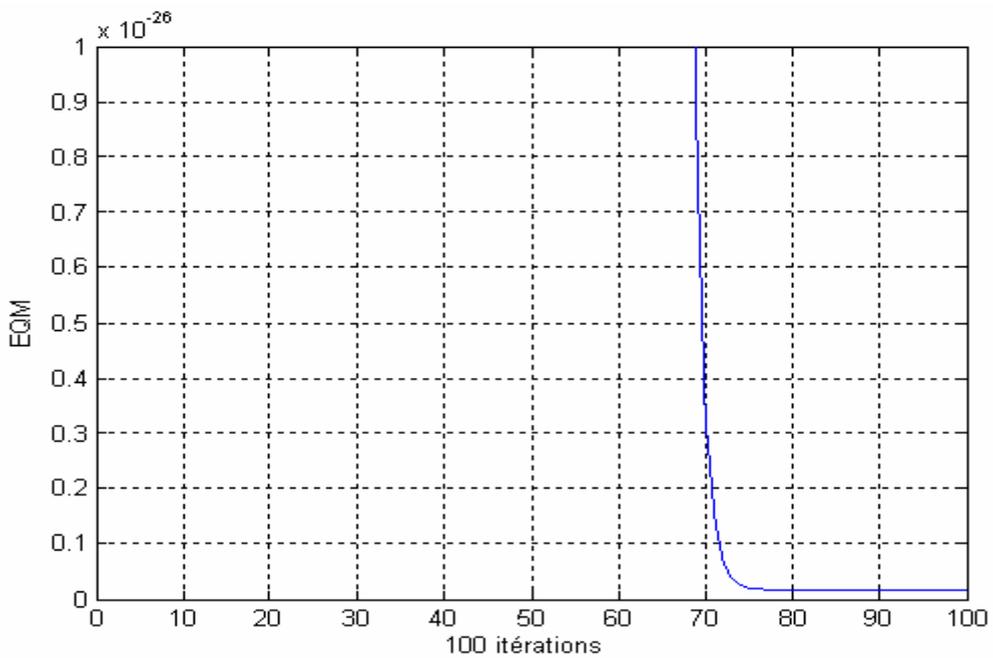


Figure 5.16 : Evolution d'EQM du réseau expert1 après 100 itérations et de 100 exemples

EQM : 1.7014E-28

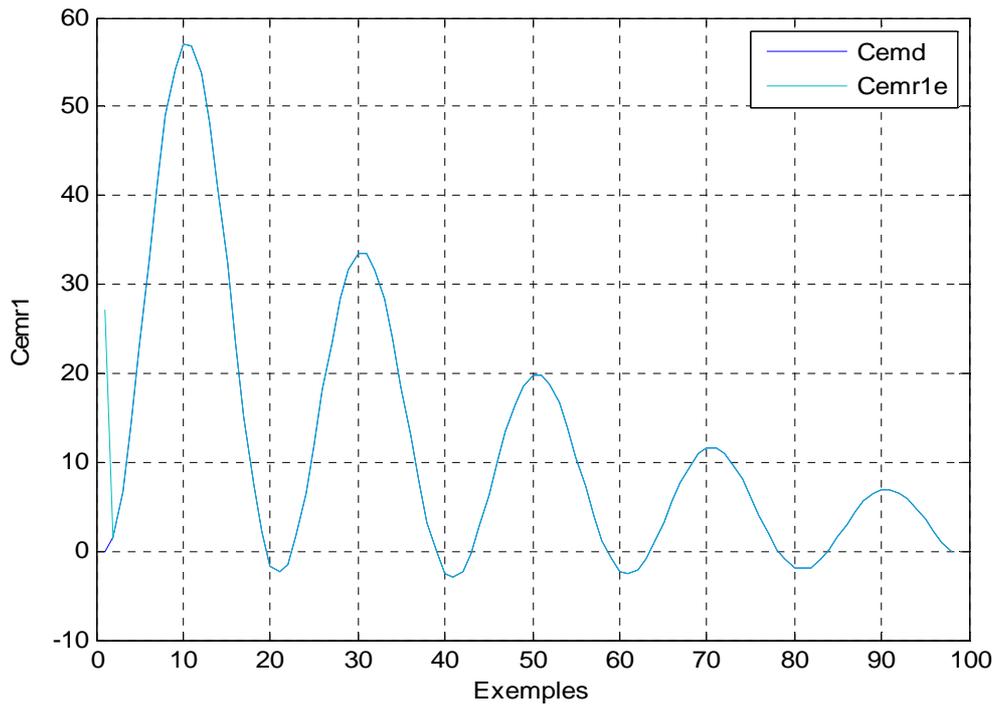


Figure 5.17 : La réponse estimée et désirée rapide du réseau expert1 après 100 itérations et de 100 exemples

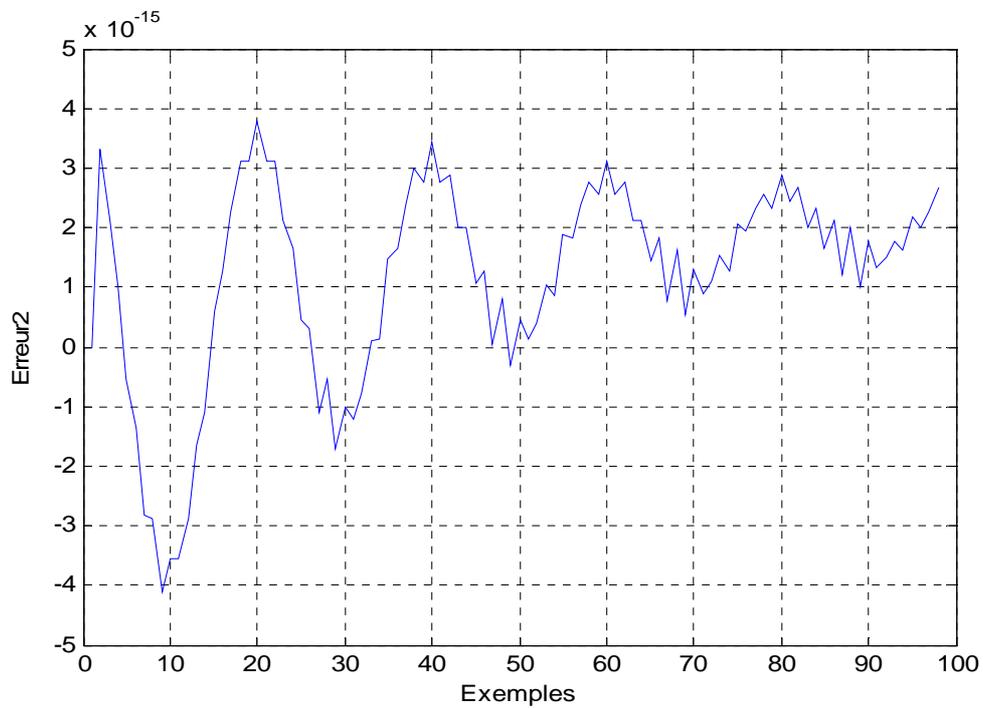


Figure 5.18 : Erreur d'apprentissage du réseau expert2 après 100 itérations et de 100 exemples

Erreur maximale : 3.775E-15

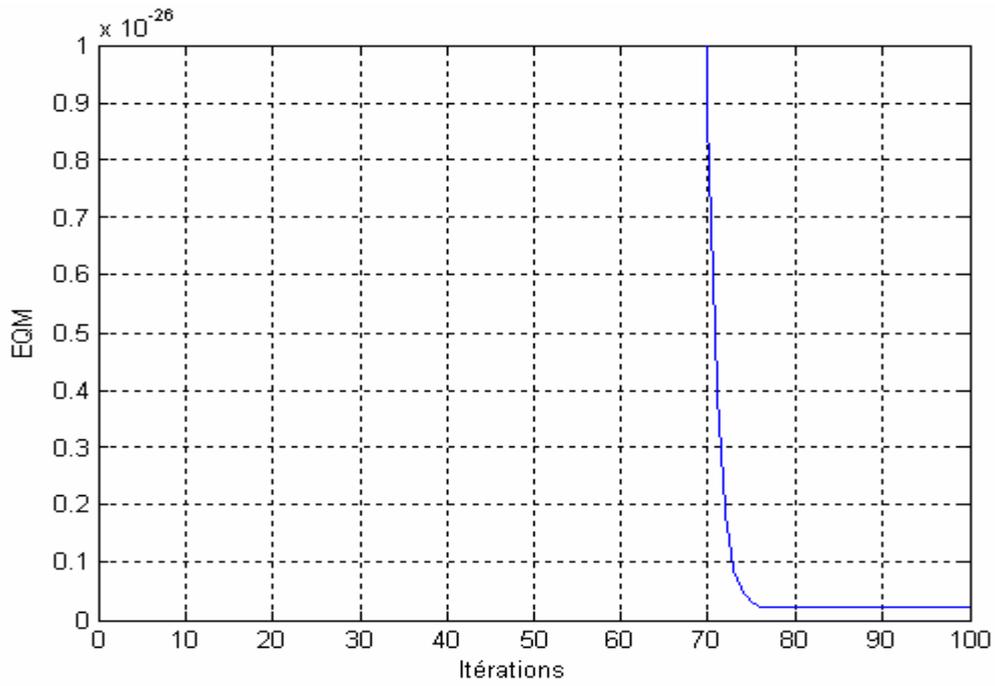


Figure 5.19 : Evolution d'EQM du réseau expert 2 après 100 itérations et 100 exemples.

EQM : 2.055E-28

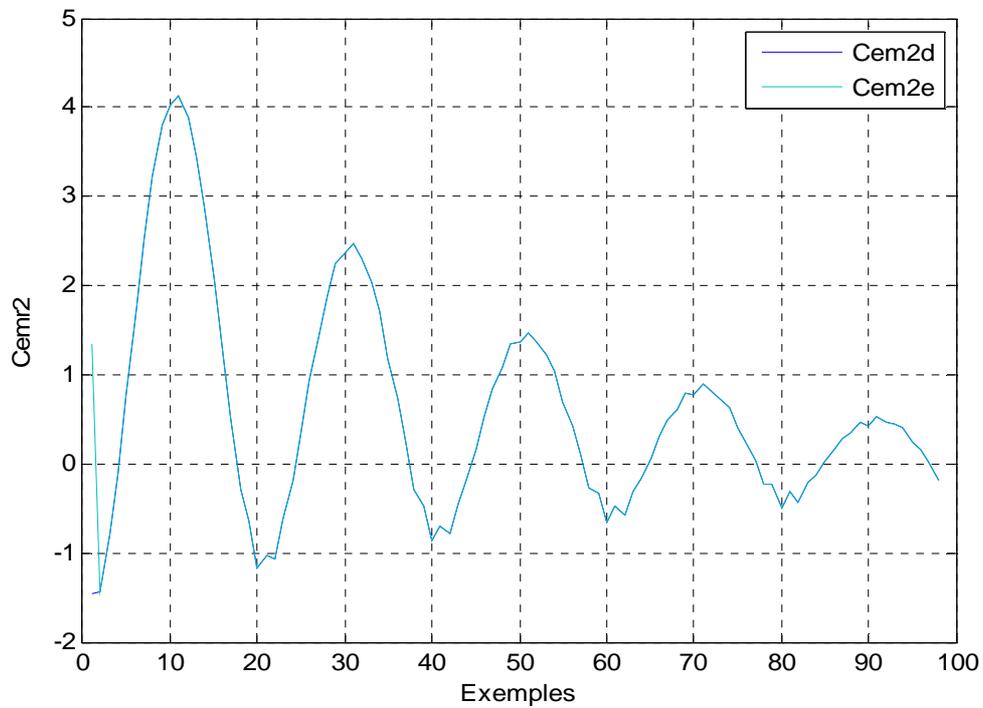


Figure 5.20 : La réponse estimée et désirée rapide du réseau expert2 après 100 itérations et de 100 exemples

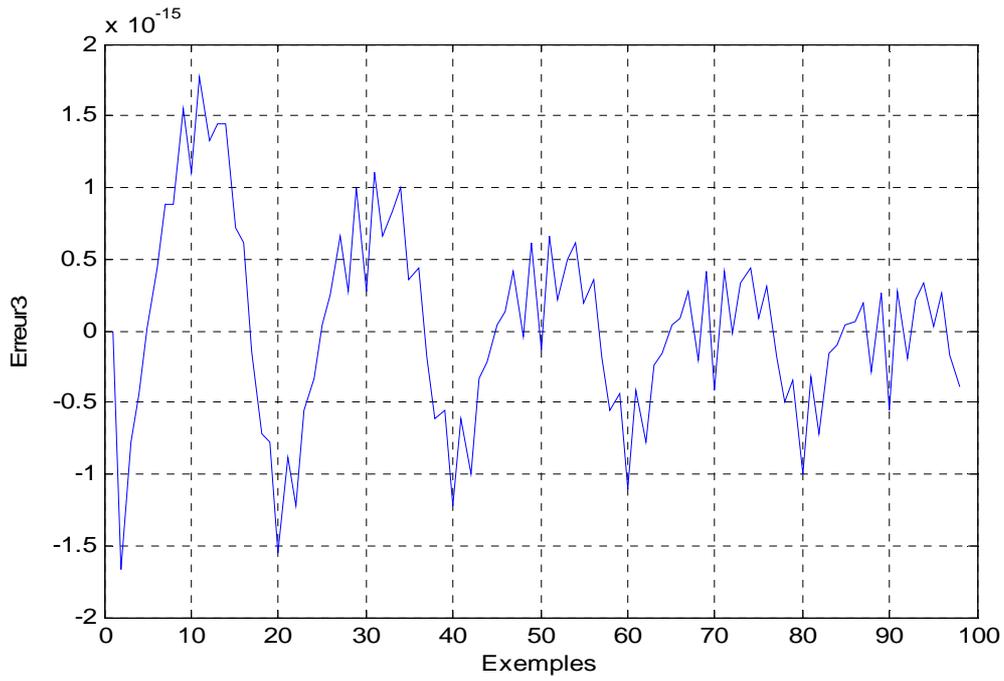


Figure 5.21 : Erreur d'apprentissage du réseau expert3 après 100 itérations et de 100 exemples

Erreur maximale : 1.776E-15

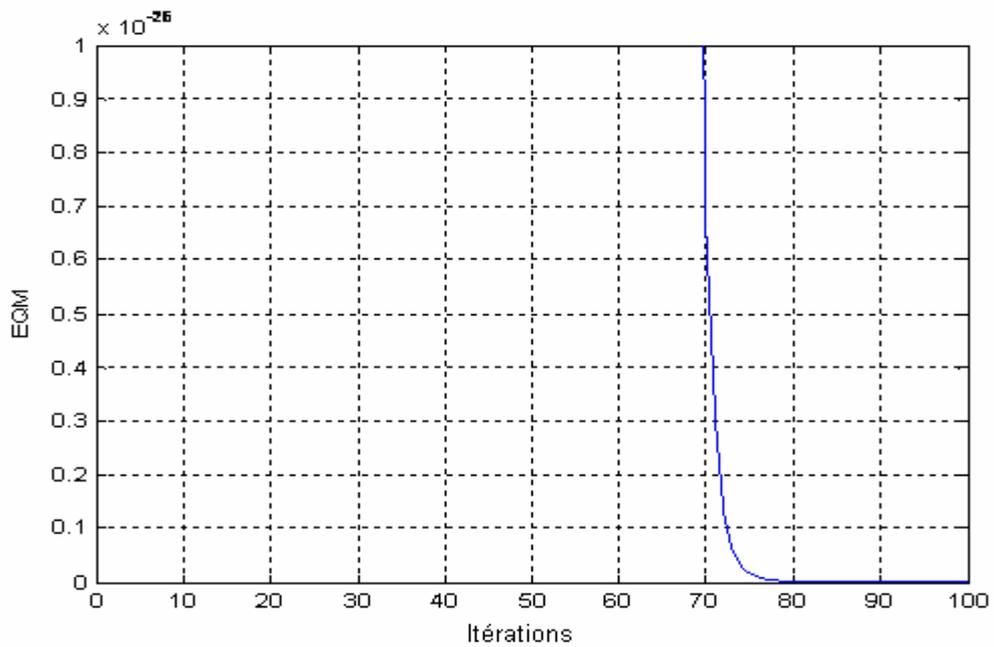


Figure 5.22 : Evolution d'EQM du réseau expert3 après 100 itérations et de 100 exemples.

EQM : 2.1211E-28

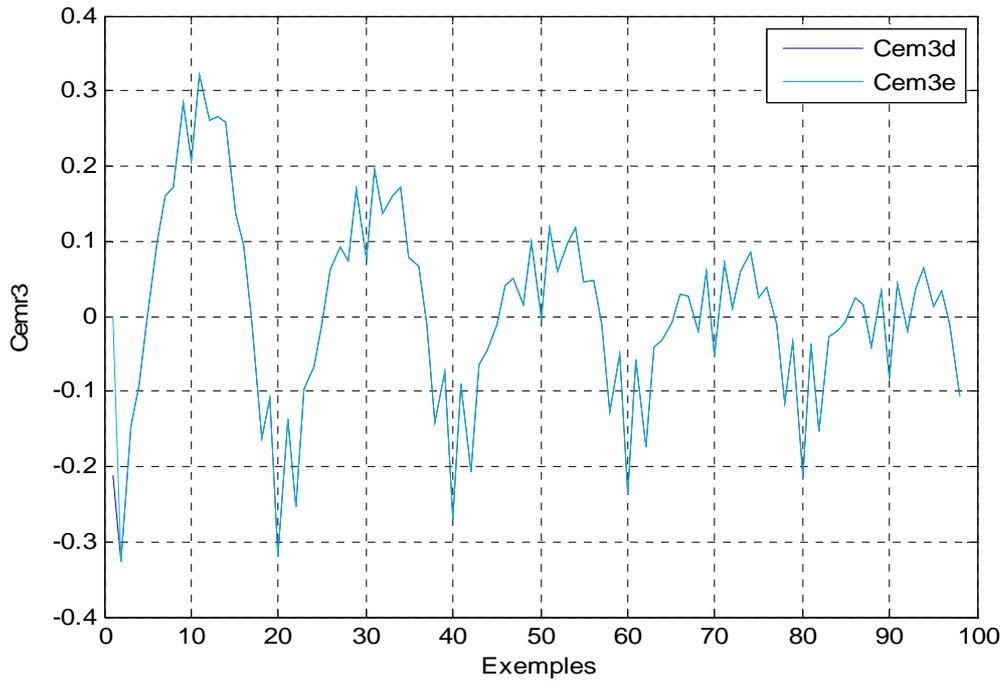


Figure 5.23 : La réponse estimée et désirée rapide du réseau expert3 après 100 itérations et de 100 exemples

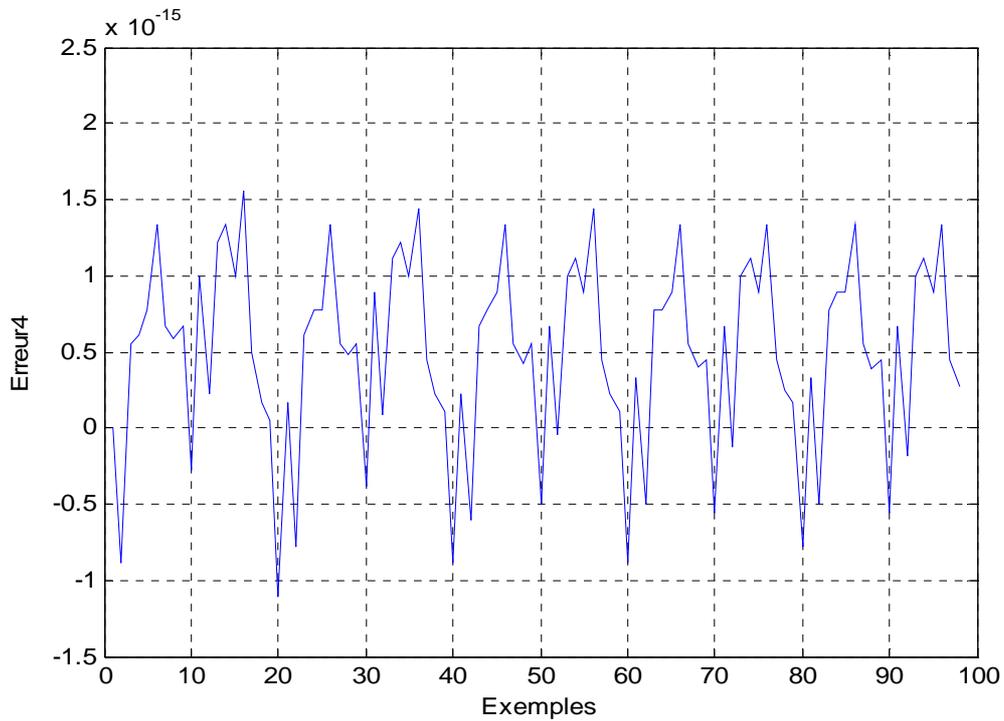


Figure 5.24 : Erreur d'apprentissage du réseau expert4 après 100 itérations et de 100 exemples

Erreur maximale : 1.554E-15

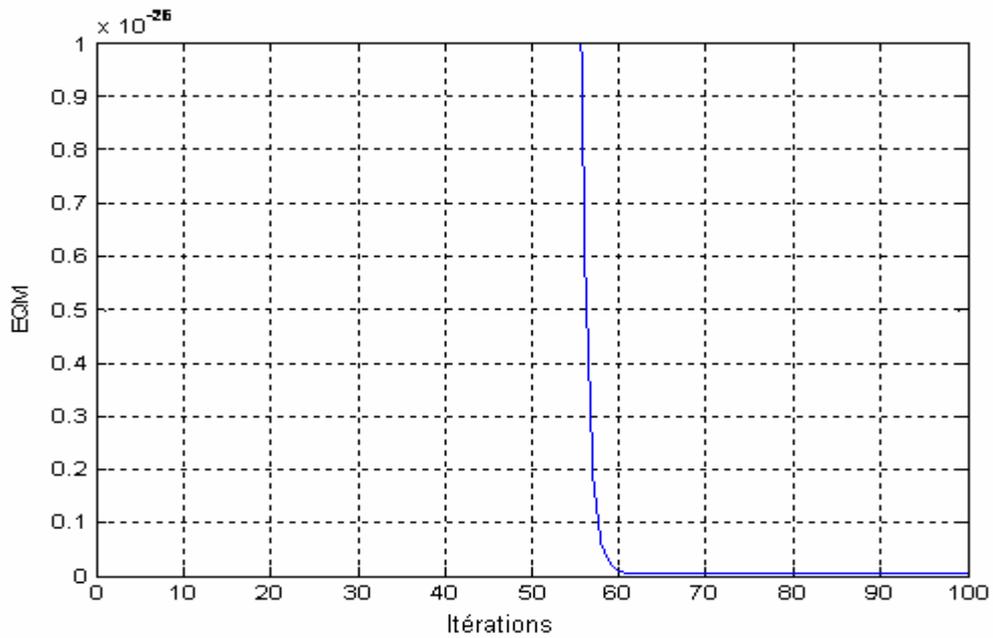


Figure 5.25 : Evolution d'EQM du réseau expert 4 après 100 itérations et de 100 exemples.

EQM : 3.0205E-29

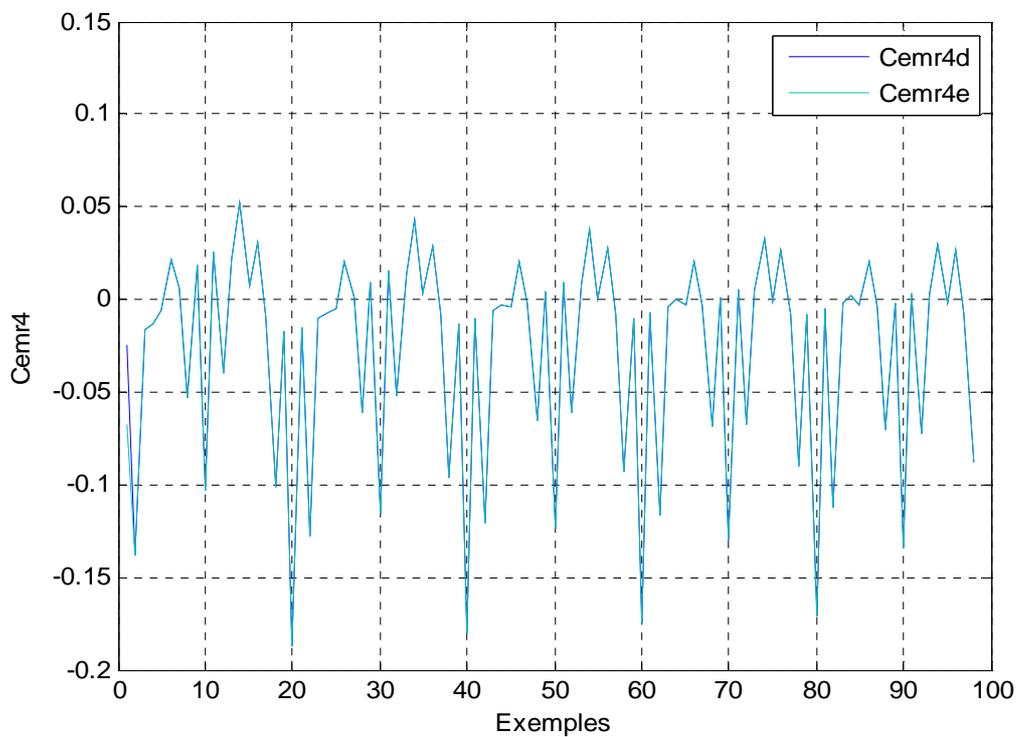


Figure 5.26 : La réponse estimée et désirée rapide du réseau expert4 après 100 itérations et de 100 exemples

Les figures (5.15, 5.18, 5.21, 5.24) représentent les écarts entre les réponses rapides estimées et les réponses rapides désirées des quatre réseaux experts respectivement.

Les figures (5.16, 5.19, 5.22, 5.25) représentent les erreurs quadratiques moyennes des quatre réseaux experts respectivement. Ces résultats montrent clairement la grande vitesse de convergence de cet algorithme. L'allure illustrée dans la figure (5.25) montre que la valeur minimale, qui est de l'ordre de  $3.0205E-29$ , est atteinte juste après (60 itérations).

Pour pouvoir comparer les performances de cet algorithme avec celles du modèle à deux réseaux, nous avons sommé les erreurs minimales obtenues des quatre réseaux experts. Les résultats sont classés dans le tableau ci-dessous :

Les experts	EQM
Expert1	1.7014E-28
Expert2	2.0550E-28
Expert3	2.1211E-28
Expert4	3.0205E-29
Réseau rapide global	6.1980E-28
Réseau Neuroglial	6.1980E-28

Tableau 5.2 : Performances des quatre réseaux experts, réseau rapide global et réseau Neuroglial pour 100 itérations

Les résultats du tableau 5.2 montrent clairement que l'erreur quadratique minimale de la réponse rapide a diminué en utilisant quatre réseaux au lieu d'un seul réseau.

### 5.3.3.3 2<sup>ème</sup> algorithme

Cette simulation permet essentiellement de comprendre et de valider l'association intelligente d'un réseau expert à une portion de l'espace d'entrée faite par le réseau glial softmax.

L'apprentissage du réseau glial permet de réaliser la sélection d'un neurone de sortie. Pour chaque entrée présentée, le réseau glial estime la réponse lente de la machine et choisi un expert vainqueur, pour avoir la réponse rapide estimée correspondante.

Chaque expert s'est spécialisé dans une région de l'espace d'entrée. Les valeurs des coefficients  $g_i$  de mélange déterminent le domaine de validité de chaque expert. Le coefficient de mélange  $g_i$  peut être compris comme la probabilité, à posteriori, de l'expert  $i$  sachant  $x_j$ . Le domaine de validité de l'expert  $i$  correspond au domaine où le coefficient de mélange  $g_i$  est égal à 1. Le domaine de validité de l'expert 1 est [0, 100], l'expert 2 est [101, 200], l'expert3 est de [201, 300] et celui de l'expert4 est de [301, 400]. Dans chacun de ces domaines, la sortie globale du réseau  $(g_1y_1 + g_2y_2 + g_3y_3 + g_4y_4)$  est identique à l'une des sorties des quatre experts.

Il est difficile de déterminer au départ, le nombre de réseaux experts nécessaire pour résoudre un problème. Si un nombre insuffisant des réseaux experts peut limiter les performances du mélange, en revanche, ajouter plus d'experts que nécessaires ne les améliore pas.

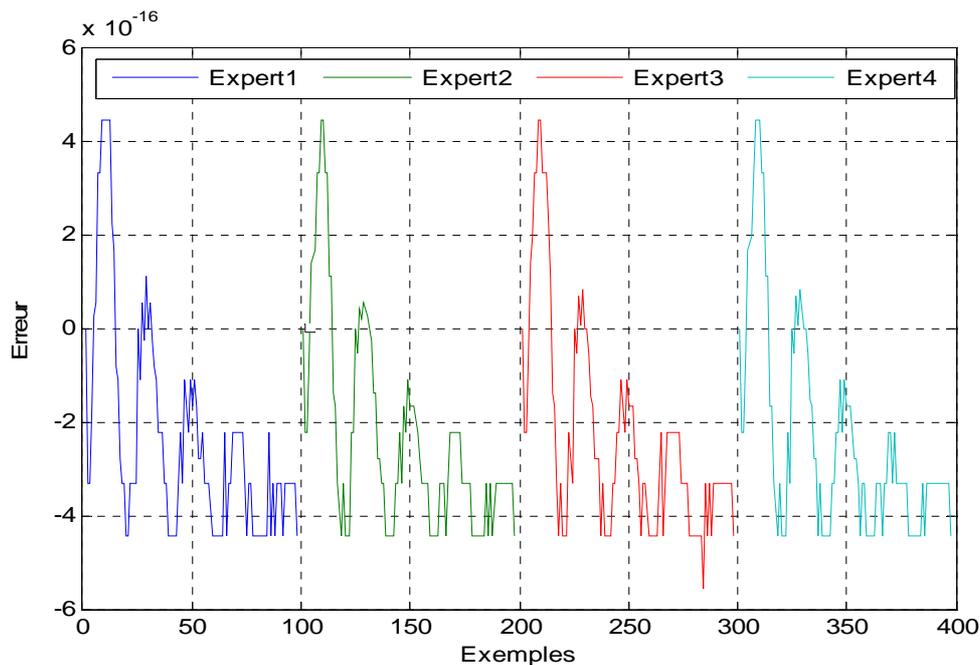


Figure 5.27 : Erreur d'apprentissage des quatre réseaux experts de l'estimation de la 1<sup>ère</sup> partie de la réponse rapide après 100 itérations et de 100 exemples

Erreur maximale expert1= 5.551E-16

Erreur maximale expert2= 5.551E-16

Erreur maximale expert3= 5.551E-16

Erreur maximale expert4= 5.551E-16

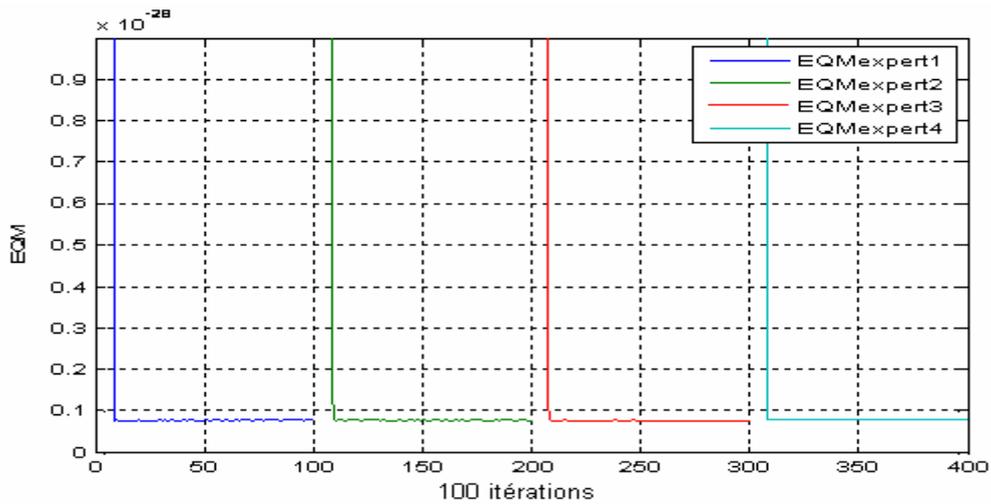


Figure 5.28 : Evolution des EQM des quatre réseaux experts de l'estimation de la 1<sup>ère</sup> partie de la réponse rapide après 100 itérations et de 100 exemples.

EQM expert1:7.7759E-30

EQM expert2: 7.8146E-30

EQM expert3: 7.7968E-30

EQM expert4: 7.8426E-30

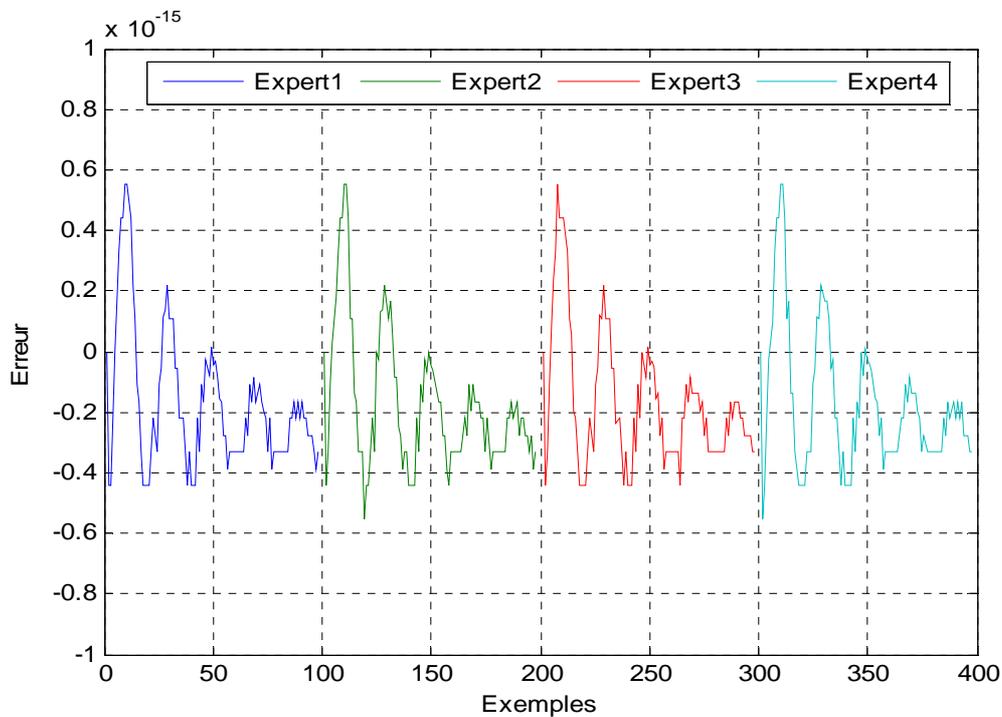


Figure 5.29 : Erreur d'apprentissage des quatre réseaux experts de l'estimation de la 2<sup>ème</sup> partie de la réponse rapide après 100 itérations et de 100 exemples

Erreur maximale expert1 : 5.51E-16

Erreur maximale expert2 : 5.51E-16

Erreur maximale expert3 : 5.51E-16

Erreur maximale expert4 : 5.51E-16

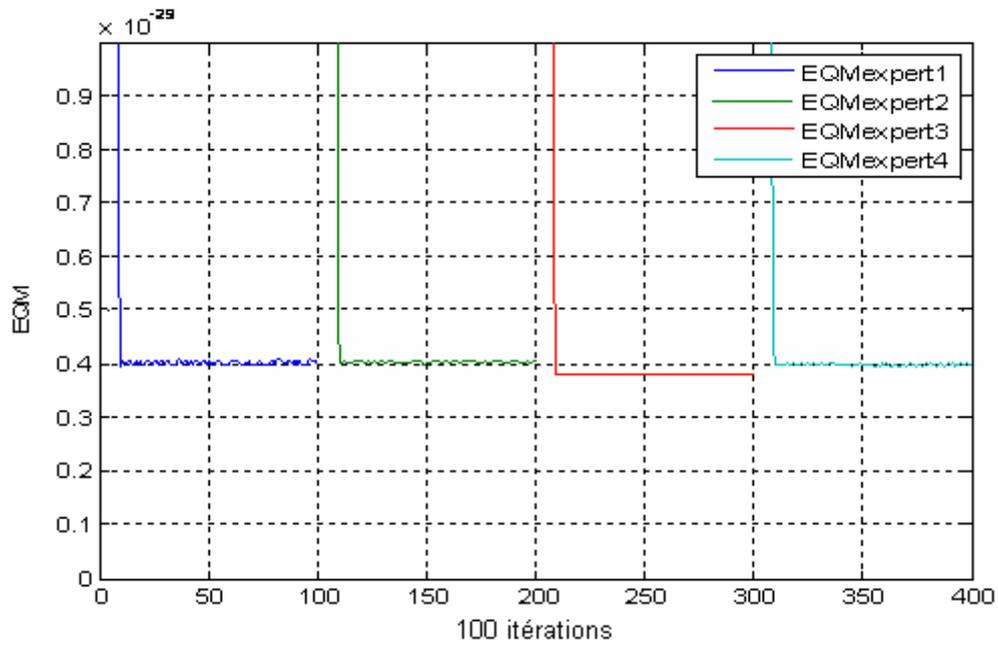


Figure 5.30 : Evolution des EQM des quatre réseaux experts de la 2<sup>ème</sup> partie de la réponse rapide après 100 itérations et de 100 exemples.

EQM expert1 : 3.9425E-30

EQM expert2 : 3.9761E-30

EQM expert3 : 3.7791E-30

EQM expert4 : 3.9544E-30

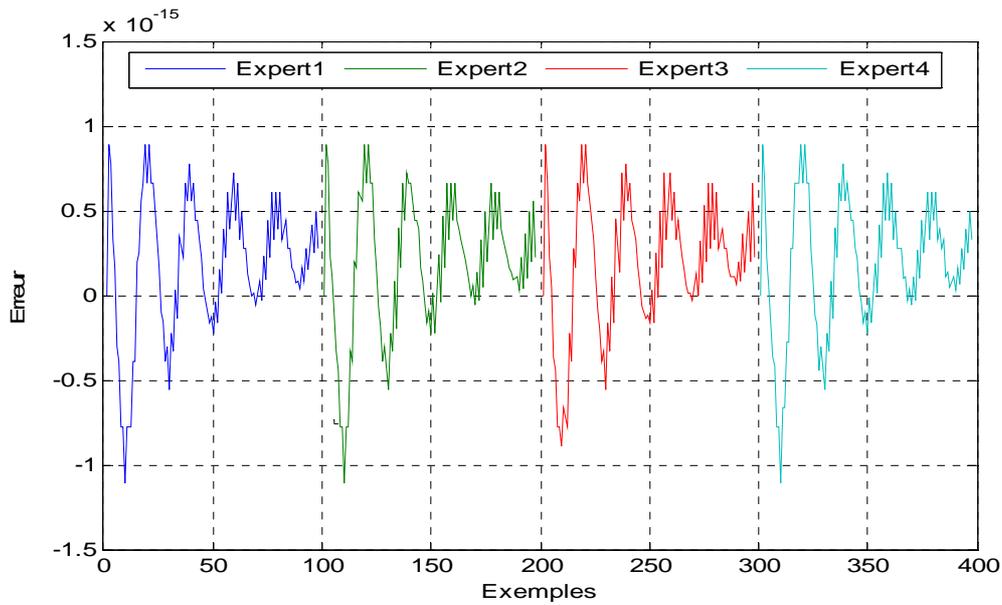


Figure 5.31 : Erreur d'apprentissage des quatre réseaux experts de l'estimation de la 3<sup>ème</sup> partie de la réponse rapide après 100 itérations et de 100 exemples

Erreur maximale expert1 : 2.7756E-16

Erreur maximale expert2 : 2.2204E-16

Erreur maximale expert3 : 2.2204E-16

Erreur maximale expert4 : 3.3307E-16

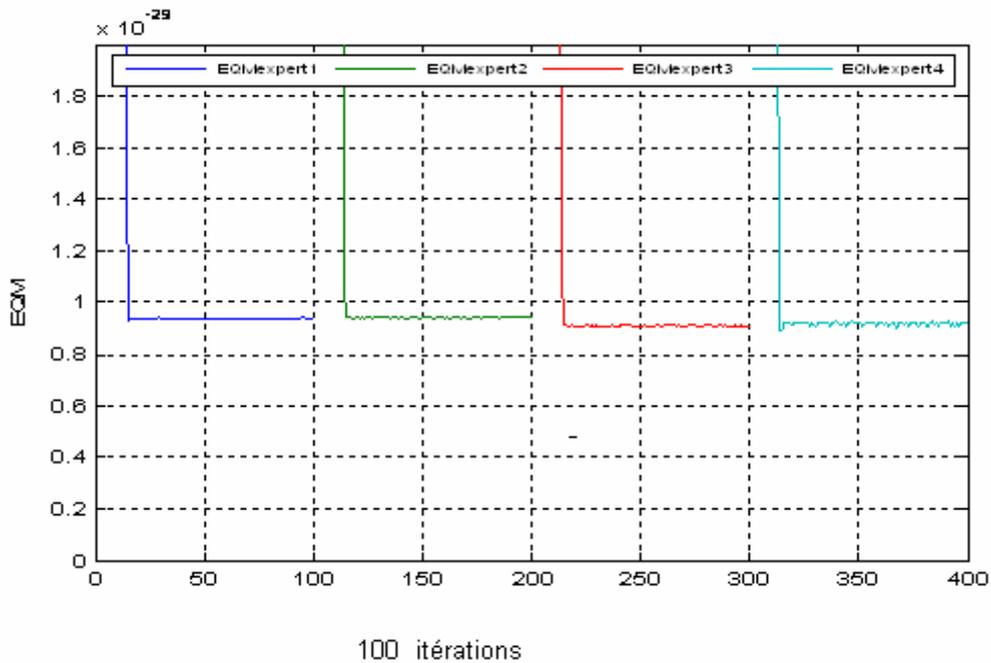


Figure 5.32 : Evolution des EQM des quatre réseaux experts de l'estimation de la 3<sup>ème</sup> partie de la réponse rapide après 100 itérations et de 100 exemples

EQM expert1 : 9.2781E-30

EQM expert2 : 9.3802E-30

EQM expert3 : 9.0447E-30

EQM expert4 : 8.9189E-30

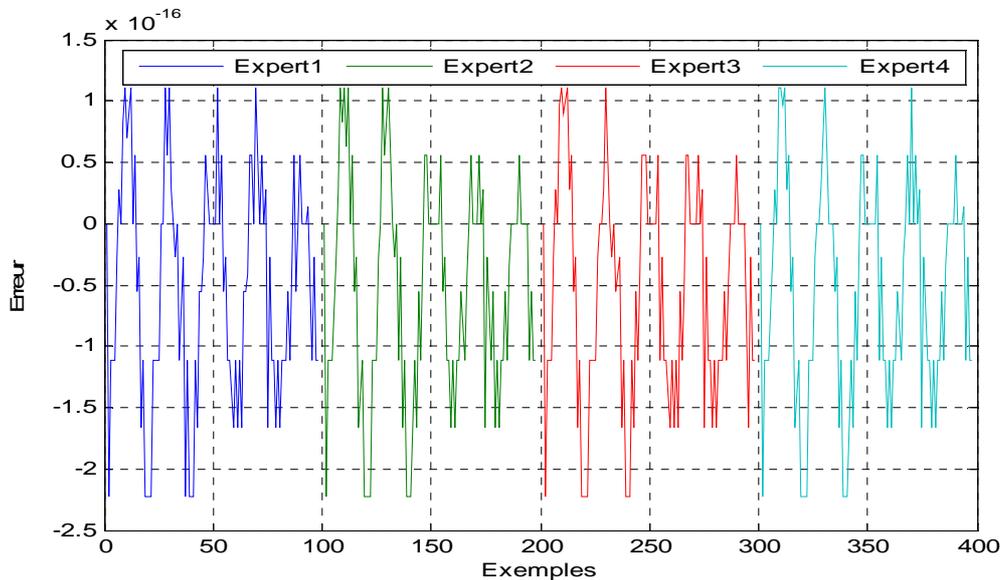


Figure 5.33 : Erreur d'apprentissage des quatre réseaux experts de l'estimation de la 4<sup>ème</sup> partie de la réponse rapide après 100 itérations et de 100 exemples

Erreur maximale expert1 : 1.11E-16

Erreur maximale expert2 : 1.11E-16

Erreur maximale expert3 : 1.11E-16

Erreur maximale expert4 : 1.11E-16

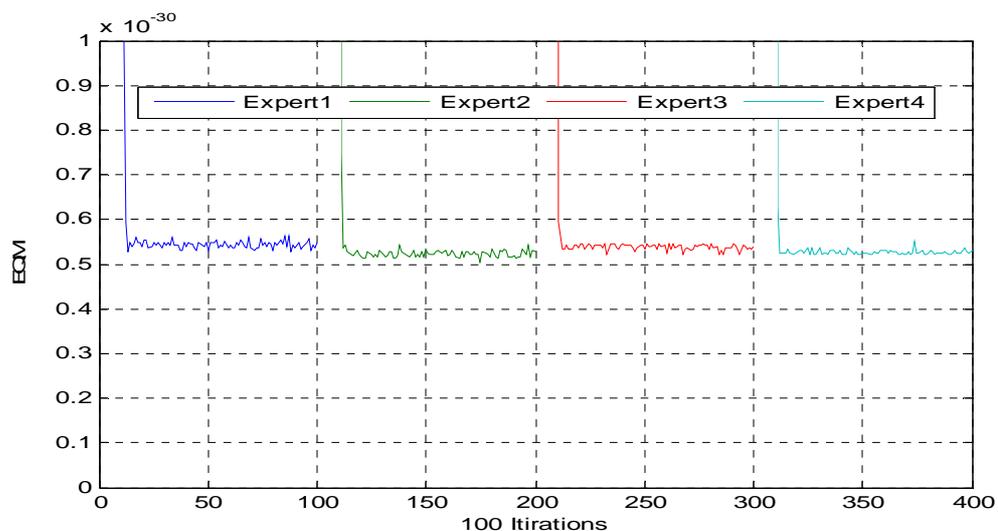


Figure 5.34 : Evolution des EQM des quatre réseaux experts de l'estimation de la 4<sup>ème</sup> partie de la réponse rapide après 100 itérations et de 100 exemples

EQM expert1 : 5.2722E-31

EQM expert2 : 5.0512E-31

EQM expert3 : 5.2224E-31

EQM expert4 : 5.2060E-31

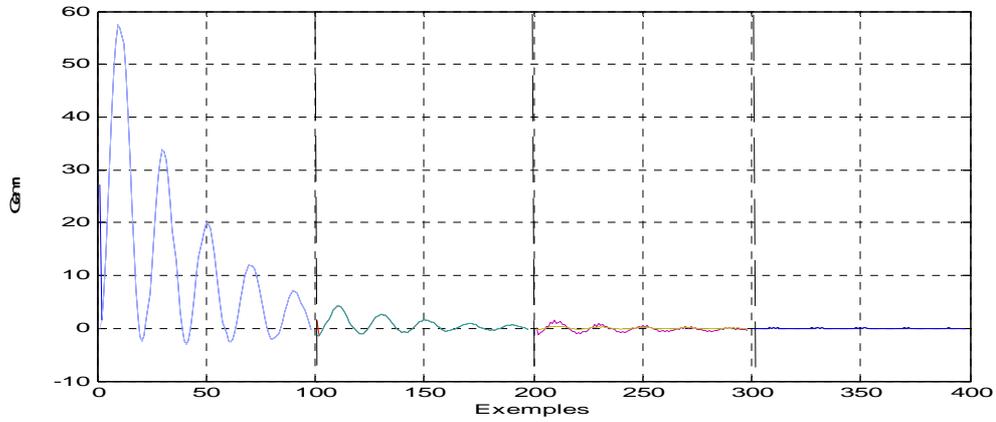


Figure 5.35 : la réponse rapide estimée par le 2<sup>ème</sup> algorithme

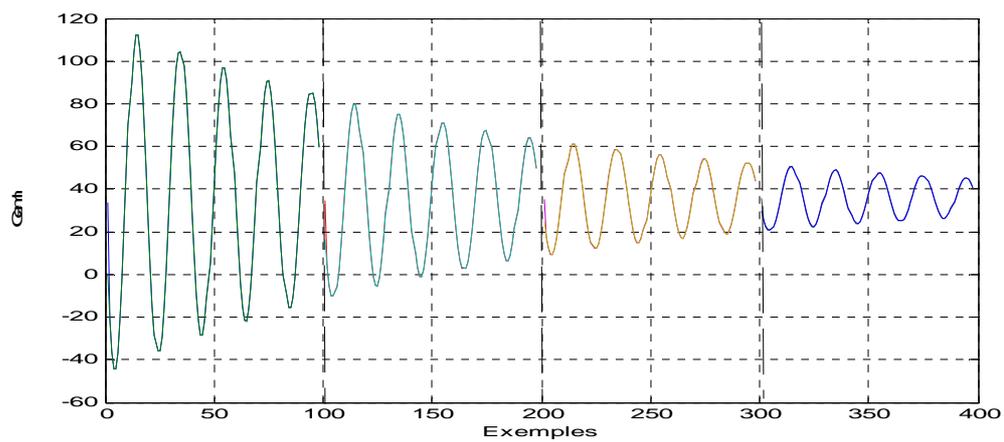


Figure 5.36 : la réponse lente du réseau glial estimée par le 2<sup>ème</sup> algorithme

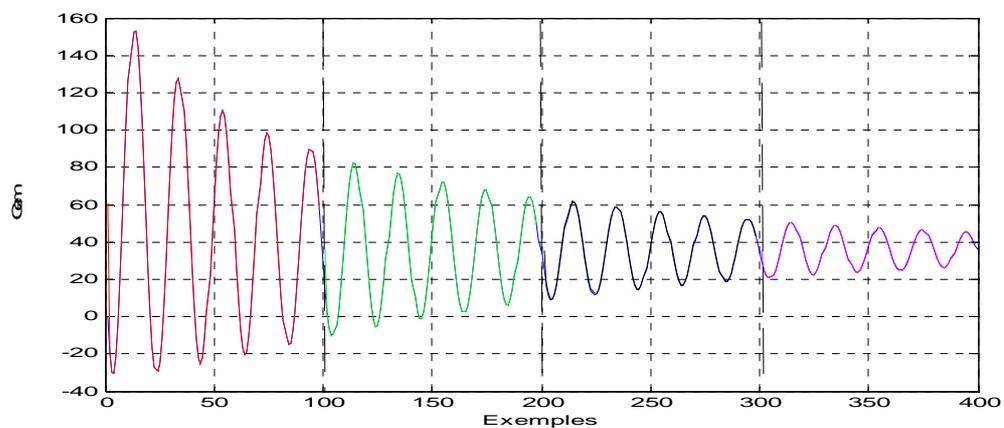


Figure 5.37 : la réponse globale du réseau Neuroglial estimée par le 2<sup>ème</sup> algorithme

Les figures (5.27, 5.29, 5.31, 5.33) représentent les écarts entre les réponses rapides estimées et les réponses rapides désirées des quatre réseaux experts pour la 1<sup>ère</sup> partie, 2<sup>ème</sup> partie, 3<sup>ème</sup> partie et 4<sup>ème</sup> partie respectivement du vecteur d'entrée. On remarque que les valeurs maximales ont diminuées par rapport aux valeurs maximales du 1<sup>ier</sup> algorithme.

Les figures (5.28, 5.30, 5.32, 5.34) montrent l'évolution de l'erreur moyenne quadratique des quatre réseaux experts ainsi que la grande vitesse de convergence de l'algorithme. La (figue 5.27) montre que la valeur minimale, de l'ordre de 7.7759E-30, est atteinte juste après 5 itérations. Ceci nous mène à dire que la bonne vitesse de convergence et la diminution de l'erreur quadratique moyenne minimale de l'algorithme sont dues à la taille réduite des réseaux et aux nombres d'exemples limités, présentés à l'entrée de chaque réseau expert, qui sont les propriétés principales de la modularité.

La répartition binaire des coefficients de mélange montre la très bonne séparation des experts effectuée par le réseau glial. Pour l'expert 1, par exemple, on constate que dès que le réseau glial indique une probabilité de 1 ( $g_1 = 1$ ), la sortie de l'expert 1 ( $y_{r1}$ ) correspond bien à la portion de la réponse rapide désirée, donc l'expert1 est le gagnant de la compétition. Inversement, lorsque que la sortie de l'expert 1 n'est pas correcte, cet expert n'est pas utilisé ( $g_1 = 0$ ). Les autres experts se comportent de manière similaire. L'expert 2 donne des valeurs de la réponse rapide correctes lorsque les données d'apprentissage appartiennent au deuxième intervalle; le réseau glial lui attribue alors une probabilité de 1. Par contre, l'expert2 donne de faibles résultats lorsque les données appartiennent aux autres intervalles mais cet expert n'est pas utilisé, car la probabilité  $g_2$  est nulle.

Les résultats de cette simulation sont classés dans le tableau ci-dessous

Domaine de validité	Expert vainqueur	$g_1$	$g_2$	$g_3$	$g_4$	EQM
La 1 <sup>ère</sup> partie	Expert1	1	0	0	0	7.7759E-30
La 2 <sup>ème</sup> partie	Expert3	0	0	1	0	3.7791E-30
La 3 <sup>ème</sup> partie	Expert2	0	1	0	0	5.0512E-31
La 4 <sup>ème</sup> partie	Expert4	0	0	0	1	8.9189E-30

Tableau 5.3: pondérations et performances des réseaux experts pour chaque domaine de validité

Les résultats de simulation classés dans le tableau 5.3 représentent les erreurs quadratiques moyennes minimales des quatre réseaux experts pour chaque partie de la réponse rapide désirée ainsi de leurs pondérations dans chaque cas. Ces résultats montrent la bonne convergence de l'algorithme.

L'erreur minimale du réseau rapide global est la somme des erreurs des quatre experts vainqueurs de chaque partie qui est de l'ordre de  $21.0251E^{-30}$ .

La combinaison du réseau glial (figure 5.35) et du réseau rapide (figure 5.36) donne le réseau Neuroglial (figure 5.37) dont l'erreur est de l'ordre de  $21.0974E^{-30}$ .

D'après ces résultats, on peut conclure que les performances sont nettement améliorées par le réseau Neuroglial en utilisant le deuxième algorithme.

### 5.3.4 Comparaison entre les différentes architectures

L'objectif de ce paragraphe est de mettre en évidence les avantages des modèles modulaires Neuroglial\_MAS par rapport à d'autres modèles neuronaux modulaire et unique employés pour l'estimation de la machine asynchrone.

Pour ce faire, nous avons développé :

- Le modèle à réseau de neurone unique
- Le modèle modulaire à deux réseaux de neurones
- Et Le modèle Neuroglial en parallèle avec la même base de données pour une comparaison la plus juste possible.

Les performances en termes d'erreurs quadratiques moyennes minimales sont décrites dans le Tableau 5.3.

Le choix du «meilleur» algorithme n'est pas évident et peut se faire uniquement sur des considérations de performances, d'autant plus que les erreurs sont globalement très faibles. Le problème le plus important des modèles uniques MLP c'est qu'ils sont ininterprétables.

Les modèles modulaires ont l'avantage de pouvoir remédier partiellement à ce problème. Ils ont l'avantage indéniable, de par leur nature, de pouvoir effectuer des partitions de L'espace d'entrée selon des critères physiques (figure5.4), comme dans le cas du modèle Neuroglial\_MAS.

Finalement le modèle Neuroglial\_MAS a l'avantage d'être relativement interprétable.

D'après ces comparaisons, on constate que l'approche Neurogliale, basée sur la fonction softmax, permet une meilleure estimation de la réponse de la machine asynchrone. En effet l'erreur quadratique atteint  $10E-31$  après quelques itérations. Ceci avec un minimum de calcul et une grande vitesse de convergence.

#### 5.4 Conclusion

Dans ce chapitre, nous avons commencé par développer le modèle de la machine asynchrone sous sa forme singulièrement perturbée.

Les modèles réduits sont obtenus après deux étapes :

- Application de l'approche géométrique basée sur les cercles de Gerschgorin pour mettre les modèles originaux sous forme standard singulièrement perturbée.
- Application de la technique des perturbations singulières.

Les modèles réduits "lent et rapide" de la machine asynchrone constituent des modèles adéquats à notre application.

Nous avons discuté les avantages d'une architecture modulaire Neurogliale pour l'apprentissage des fonctions possédants deux échelles de temps (2 vitesses différentes).

En effet l'apprentissage a été effectué pour les modèles suivants :

- Le modèle à réseau de neurone unique MLP.
- Le modèle à deux réseaux de neurones.
- Et enfin, le modèle Neuroglial.

On constate que le modèle modulaire à deux réseaux de neurones et le modèle modulaire Neuroglial ont une grande vitesse de convergence, une complexité réduite, et une erreur quadratique minimale par rapport au modèle à réseau unique.

Egalement le modèle Neuroglial\_MAS est plus performant que le modèle à deux réseaux de neurones en termes d'erreur quadratique minimale et vitesse de convergence.

Une autre comparaison a été effectuée avec la même architecture Neurogliale mais avec deux algorithmes d'apprentissages différents du réseau glial. L'algorithme basé sur la fonction softmax est beaucoup plus meilleur que l'algorithme basé sur la sélection automatique.

En fin, pour conclure le modèle modulaire Neuroglial offre de meilleures performances en termes de vitesse de convergence, de complexité de calcul et d'erreur minimale quadratique. Ceci met en évidence qu'une telle architecture est la mieux adaptée aux systèmes complexes présentant deux échelles de temps.

## CONCLUSION

L'étude présentée dans ce mémoire concerne l'application des réseaux de neurones à réseau glial aux modèles de la machine asynchrone. Cependant, nous avons décrit le fonctionnement biologique des cellules gliales et explicité les dernières connaissances acquises concernant le rôle majeur de ces cellules dans la communication cérébrale. D'après les recherches avancées nous avons retenue que dans notre cerveau l'information circule en deux voies : une rapide, réseau de neurones et l'autre lente, réseau glial et que le réseau de neurones doit être alimenté et contrôlé par le réseau glial.

Dans ce contexte, nous avons développé une nouvelle approche neuronale à réseau glial, appelée Neurogliale qui émule cette réalité biologique.

A cet effet, des notions des réseaux de neurones ont été abordées ainsi que les paradigmes d'apprentissages, architectures, les avantages, les inconvénients et les applications, des réseaux de neurones.

Notre conception est basée sur le concept puissant modularité. Les différentes formes de décomposition qui existent ont été exposées. Nous avons présenté, en détail, la nouvelle approche Neurogliale, aussi deux algorithmes d'apprentissages ont été développés.

Le modèle de la machine asynchrone, mise sous forme singulièrement perturbée, constitue un modèle adéquat pour la validation de l'approche Neurogliale. En effet, pour le développement d'un tel modèle, on s'est appuyé sur l'approche géométrique des cercles de Gerschgorin. Cette approche permet de localiser et de séparer les variables lentes et rapides. En suite la méthode des perturbations singulière a été appliquée pour le développement des modèles réduits lent et rapide.

Afin d'évaluer le modèle Neuroglial\_MAS, une étude comparative, entre ce modèle, le modèle à réseau unique et le modèle modulaire à deux réseaux, a été effectuée.

On constate que :

-Le modèle modulaire à deux réseaux a permis d'obtenir deux réseaux de taille réduite avec moins de complexité et de meilleures performances en terme d'erreur moyenne quadratique que le modèle à réseau unique.

-Le 1<sup>er</sup> algorithme du modèle Neuroglial\_MAS présente de meilleures performances et une grande rapidité de convergence, que le modèle modulaire à deux réseaux.

-Le 2<sup>ème</sup> algorithme du modèle Neuroglial\_MAS basé sur la fonction softmax présente de meilleures performances en termes d'erreur quadratique et vitesse de convergence que le 1<sup>er</sup> algorithme.

Dans le futur, plusieurs axes de recherches peuvent être poursuivis, notamment :

-La modélisation et le contrôle des systèmes à multi-échelles de temps en utilisant l'approche Neurogliale.

-La modélisation l'analyse et le contrôle dans le domaine de la robotique où se présentent des dynamiques différentes.

## APPENDICE A TRANSFORMATION DE PARK MACHINE ASYNCHRONE TRIPHASÉE

D'après la théorie générale des machines électriques, le comportement de n'importe quelle machine polyphasée, (m et n phase au stator et au rotor), à condition d'égalité des résistances au stator et au rotor, peut être représentée par son modèle diphasé. De ce fait, un enroulement (triphasé du stator ou du rotor) peut être substitué par un enroulement diphasé équivalent. Il est donc possible, de définir une matrice qui porte le nom de transformation de « CLARKE »  $[T_1]$ , permettant le passage des composantes  $X_{a,b,c}$  du système triphasé aux composantes  $X_{\alpha,\beta,0}$  du système diphasé ; tournant à la même vitesse, telle que :

$$\begin{bmatrix} X_\alpha \\ X_\beta \\ X_0 \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & \cos(2\pi/3) & \cos(4\pi/3) \\ 0 & \sin(2\pi/3) & \sin(4\pi/3) \\ 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} X_a \\ X_b \\ X_c \end{bmatrix} \quad (\text{A.1})$$

Sous la forme condensée, nous avons :

$$[X_{\alpha,\beta,0}] = [T_1] [X_{a,b,c}].$$

Afin d'aboutir à une expression du couple électromagnétique développé par la machine en fonction des grandeurs continues et des paramètres physiques fixes, il faut exprimer les équations de la machine dans un repère où, le rotor devient mutuellement fixe par rapport au stator.

A partir du repère de la figure (A-1), nous ajoutons un autre repère. Ce dernier est caractérisé par un système d'axe (d,q) tournant à la vitesse  $\omega_c = \frac{d\theta_c}{dt}$ ; par rapport au

stator. C'est la

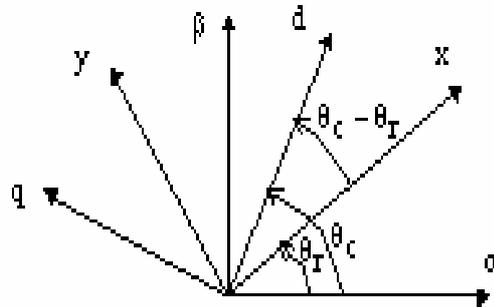


Figure A.2 : représentation de la transformation dans le repère tournant.

Donc, on peut définir une matrice de passage, du repère  $(\alpha, \beta, 0)$  au repère  $(d, q, 0)$ , figure (A-2).

Cette matrice est notée  $[T_2]$ , telle que :

$$\begin{bmatrix} X_d \\ X_q \\ X_0 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_\alpha \\ X_\beta \\ X_0 \end{bmatrix} \quad (\text{A.2})$$

Soit, sous la forme condensée :

$$[X_{d, q, 0}] = [T_2] [X_{\alpha, \beta, 0}].$$

Où  $\theta$  représente  $\begin{cases} \theta_c & \text{pour les grandeurs statoriques.} \\ \theta_c - \theta_r & \text{pour les grandeurs rotoriques.} \end{cases}$

Ainsi, nous pouvons définir une transformation unique permettant le passage direct du système triphasé au système diphasé tournant.

Cette transformation sera notée  $[P]$ , elle est donc définie par le produit matriciel suivant :

$$[P] = [T1]. [T2].$$

D'où :

$$P(\theta) = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & \cos(\theta - 2\pi/3) & \cos(\theta - 4\pi/3) \\ -\sin(\theta) & -\sin(\theta - 2\pi/3) & -\sin(\theta - 4\pi/3) \\ 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \quad (A.3)$$

Le passage des variables  $X_{a, b, c}$  aux variables  $X_{d,q}$ , s'écrit sous la forme matricielle suivante :

$$\begin{bmatrix} X_d \\ X_q \\ X_0 \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & \cos(\theta - 2\pi/3) & \cos(\theta - 4\pi/3) \\ -\sin(\theta) & -\sin(\theta - 2\pi/3) & -\sin(\theta - 4\pi/3) \\ 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} X_a \\ X_b \\ X_0 \end{bmatrix} \quad (A.4)$$

Cette matrice est souvent appelée transformation de Park normalisée.

Nous définissons de la même façon, la matrice inverse permettant le passage (diphase-triphasé) ; notée par  $[P^{-1}]$  et donnée par :

$$P(\theta)^{-1} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 1/\sqrt{2} \\ \cos(\theta - 2\pi/3) & -\sin(\theta - 2\pi/3) & 1/\sqrt{2} \\ \cos(\theta - 4\pi/3) & -\sin(\theta - 4\pi/3) & 1/\sqrt{2} \end{bmatrix} \quad (A.5)$$

Sous la forme condensée, nous avons :

$$[X_{a,b,c}] = [P]^{-1} [X_{d,q,0}]$$

Nous remarquons que :  $P^{-1} = P^t$ , cette transformation est orthogonale.

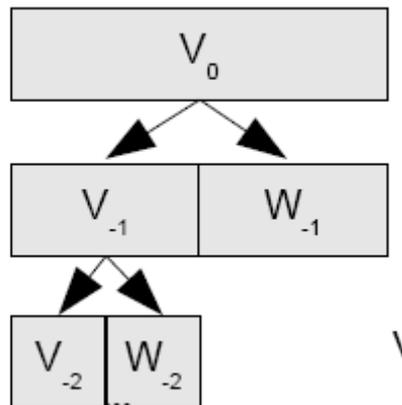
**APPENDICE B**  
**DONNEES RELATIVES À LA MACHINE ASYNCHRONE TRIPHASEE**  
**UTILISEES EN SIMULATION :**

Tension nominale $V_n$	220/380 V
Nombre de paires de pôles p	2
Résistance rotorique par phase $R_r$	3,421 $\Omega$
Résistance statorique par phase $R_s$	5,793 $\Omega$
Inductance cyclique statorique $L_s$	0,386 H
Inductance cyclique rotorique $L_r$	0,386 H
Inductance mutuelle cyclique stator-rotor $L_M$	0,363 H
Moment d'inertie de la partie tournante J	0,018 Kg.m <sup>2</sup>
La fréquence du réseau f	50 Hz

## APPENDICE C LES ONDELETTES

La théorie des Ondelettes est apparue au début des années 1990, elle touche de nombreux domaines des mathématiques, notamment le traitement du signal et des images. La boîte à outils Ondelettes de Matlab propose de nombreuses fonctions de manipulation des Ondelettes.

L'analyse multirésolution donne un ensemble de signaux d'approximation et de détails d'un signal de départ en suivant une approche fin à grossier (*fine-to-coarse*). On obtient une décomposition multi-échelle du signal de départ en séparant à chaque niveau de résolution les basses fréquences (approximation) et les hautes fréquences (détails) du signal [29].



$V_0$  : Espace de départ,  $V_{-1}$  : Approx. Niv. 1,  $W_{-1}$  : Détails niv. -1, ...

Les commandes ondelette en matlab utilisées pour la décomposition :

**S** : le signal à décomposer

```
ls=length(s);
```

```
[ca1,cd1]=dwt(s,'db1');
```

```
a1=upcoef('a',ca1,'db1',1,ls)
```

```
d1=upcoef('d',cd1,'db1',1,ls);
```

a1 : Composante lente

d1 : Composante rapide

## **APPENDICE D LISTE DES ABRÉVIATIONS**

SNC	: Système Nerveux Centrale
ANGN	: Artificiel Neuroglial Networks
RNA	: Réseaux de Neurones Artificiels
GS	: Glial System
ME	: Mixture of Experts
HMES	: Hierarchical Mixture of Experts
EQM	: Erreur Quadratique Moyenne
MSE	: Mean Square Error
NG	: Neuroglial
NG_MAS	: Neuroglial Machine asynchrone
MLP	: Multi Layer Perceptron
ANN	: Artificiel Neural Networks

## REFERENCES

1. G.Fortin «La Cellule Gliale» La Presse, Montréal, 25 Août .2002.
2. «La Cellule Gliale», article de wikipédia.
3. J.Zin «Les réseaux de neurones», article de Science Sep. 2004.
4. A. Dale Stanley «A Theory of Cortical Neuron-Astrocyte Interaction», department of Pediatrics', Harvard Medical School, The Fernald Center, 1998.
5. «Astrocytes», Science & Vie, n° 1045, p. 67, Nov 2005.
6. J.Guilan Zin«Astrocytes et maladies Neurodégénératives », conférence oct 2006.
7. Ana B. Porto, A. Pazos «Neuroglial Behaviour in Computer Science», Idea Group Publishing 2006.
8. V. Vladislav, E. Ben-Jacob, L. Levine «TheAstrocyte as a Gatekeeper of Synaptic Information Transfer», article Mai 2006.
9. R. Lepage «Reconnaissance d'algue toxiques par vision artificielle et réseau de neurones» mémoire de recherché présenté du l'université du Québec a Rimouski, 09 2004.
10. N.Izboudjen «Synthèse VHDL et Implémentation d'un réseau de neurones basé sur L'algorithme de Kohonen Sur un circuit FPGA», mémoire de fin d'étude, Département instrumentation et automatique, Juin 2002
11. P. Kokotovic, H.K. Khalil and J. O'reilly, « Singular Perturbation Methods in Control. Analysis and Design », Academic Press London, 1986.
12. G. Gandanegara «Analyse et Simplification de Modèle par Bond Graphs de Systèmes de Traction Electrique», LEEI, UMR INPT, 31071 Toulouse Cedex.
13. R. Bouyekhif, A. El Moudni, A. El Hami, N. Zerhouni , M. Ferney, «Reduced Order Modelling of Two-time-scale Discrete Non-linear Systems », Journal of the Franklin Institute, vol. 333, pp. 499-512, Elsevier Science Ltd, 1996.
14. O. Touhami, A. Mezouar, R. Ibtouen et S.Mekhtoub. «Dynamics Separation of Induction Machine Models Using Gerschgorin's Circles and Singular Perturbations», IEEE-ICEEE 2004, Research Laboratory of Electrotechnic-Polytechnic National School.

15. J. H. Chow and P. V. Kokotovic «A decomposition of near-optimum regulators for systems with slow and fast modes », IEEE Trans. Cont, Vol. 21, pp701-705, 1976.
16. A. Ansary, J. A. De Abreu-Garcia «A simple algorithm for time scale separation », IEEE-TEES, 1989, PP. 445-448.
17. H. Guesbaoui et C. lung «Multi time Scale Modelling in Electrical Machines », Centre de Recherche en Automatique de Nancy, CNRS URA 821
18. Happel1, M.J. Murre «The Design and Evolution of Modular Neural Network Architectures », Proceedings of the 1992 International Conference on Artificial Neural Networks, 1994, 7, 985-1004.
19. T. Caelli, L. Guan, W.Wen «Modularity in neural computing », Proceeding of the IEEE, VOL. 87, NO. 9, SEPTEMBER 1999, 1497-1518.
20. M. Khairiyah «Development, analysis and comparison connectionist models for real time optimisation», these présentée à l'université Waterloo, Canada, 2001
21. E. Ronco, H. Gollee, P. Gawthrop « Modular neural networks and self decomposition», Technical Report CSC-96012, 11 Fev 1997.
22. W. Charles, W. Anderson Zhaohui Hong, « Reinforcement Learning with Modular Neural Networks for Control» Department of Computer Science Colorado State University Fort Collins CO 80523.
23. T.Caelli, L. Guan, W.Wen «Modularity in neural computing », proceedings of IEEE Special issue on computational intelligence, 1497-1518.
24. Y.Neng Hwang «Neural Network Signal Processing», livre 2001.
25. I. Kirschning, H. Tomabechi, J.I. Aoe « A Parallel Recurrent Cascade-Correlation Neural Network with Natural Connectionist Glue», Proceedings of the IEEE ICNN, Perth, Australia 1995.
26. Druker, Al « Boosting and other ensemble methods », Neural computation 1289-13014, 1994
27. M.P. Perrone, L.N. Cooper «When networks disagree: Ensemble methods for Hybrid neural networks», Artificial neural networks for speech and vision October 27 1992.
28. S. Hashem, B. Schmeiser, Y.yih«Optimal linear combinations of neural networks:an overview», IEEE Technical report SMS
29. J. Landré « Boîte à outils Ondelettes sous Matlab», Journées du club E.E.A. : Ecole Analyse Multirésolution pour l'Image 11-13 janvier 2006.

30. C. Sueur, G. Dauphin Tanguy, «Bond Graph Approach to Multi-time Scale Systems Analysis », Journal of the Franklin Institute vol. 328, pp. 1005-1026, Press, 1991.