

UNIVERSITÉ SAAD DAHLAB DE BLIDA 1

Faculté de Technologie

Département d'Electronique

THESE DE DOCTORAT

Spécialité : Signaux & Systèmes

**MODELISATION D'UN SYSTEME SINGULIEREMENT
PERTURBE PAR LA METAHEURISTIQUE NEUROGLIALE ET
CONTRÔLE FLOU OPTIMUM DU PROCESSUS**

Par

Samia SALAH

devant le jury composé de :

| | | |
|---------------|-------------------------------------|---------------|
| K. Kara | Professeur, U. de Blida 1 | Président |
| M.Hadj Sadok | Maitre de conférence, U. de Blida 1 | Rapporteur |
| A.Guessoum | Professeur, U. de Blida 1 | Co-rapporteur |
| S. Abudurra | Professeur, U. de Médéa | Examineur |
| F.Boudjema | Professeur, ENP Alger | Examineur |
| M.Ould zmirli | Professeur, U. de Médéa | Examineur |

Blida, Juin 2019

يقترح هذا العمل ، نظام ربط متجاوز حدود المساعدة على الكشف يسمى "الشبكة العصبونية الدبقية الاصطناعية" "ANGN"، يتكون من شبكة عصبونية بالتوازي مع شبكة دبقية ، يقوم بمحاكاة معالجة المعلومات في الدماغ البشري بطريقة تكاد تكون مثالية. للشبكتين الفرعيتين (العصبية والدبقية) ديناميكيات مختلفة (سريعة وبطيئة)، تعتمد بنية النظام الكلية على مفهوم "التقسيم إلى وحدات" القوي و التنفيذ، باستخدام أنظمة متعددة السلالم الزمنية. هذه الأنظمة التي يشار إليها بوجه عام على أنها "أنظمة ذات اضطرابات متميزة"، هي نماذج رياضية غير خطية، مقترنة بقوة لها ديناميكيات مختلفة. يصعب بشكل عام الحصول على نماذج رياضية دقيقة لهذه الأنظمة بسبب المتصلة و اللاخطية.

يتم الحصول على التحقق من النهج المنفذ في حالة تطبيقين: "تعريف حدودي لمحرك (MCC) وكذلك لماكينة (MAS) وضعا على شكل نموذجي متميز الاضطرابات . بالنسبة لكل تطبيق ، يتم تقسيم النموذج الكلي إلى نماذج فرعية: بطيئة وسريعة ، باختيار طريقة الاضطرابات المتميزة . مبدأ هذه التقنية الأساسي هو الحفاظ على النموذج الكلي، الحد من ترتيبه وفصل الكميات الفيزيائية المعنية ، للحصول على نماذج فرعية ذات ترتيب منخفض. تستخدم التقنية الهندسية لدوائر جرشكورين (Gerchgorin) للتغلب على مشكلة تحديد المتغيرات البطيئة والسريعة أثناء الفصل. تم تطوير خوارزميتين مختلفتين لاختبار كفاءة وأداء الشبكة العصبونية الدبقية الاصطناعية "ANGN" التي تم تعريفها.

لقد أتاحت البنية الجديدة إمكانية الحصول على شبكات ذات حجم منخفض بشكل كبير مع تعقيد أقل وأداء أفضل من حيث الخطأ الترتيبي المتوسط وسرعة التقارب ، مقارنة بنماذج الشبكة العصبونية التقليدية و كذلك شبكة مزيج (ME) ، بالإضافة إلى نماذج أخرى من الشبكات العصبونية متعددة السلالم الزمنية.

الكلمات المفاتيح:

شبكة دبقية، الشبكة العصبونية الدبقية الاصطناعية، تقنية الاضطرابات المتميزة، التقنية الهندسية لدوائر Gerchgorin، النمطية، أنظمة متعددة السلالم الزمنية.

RESUME

Ce travail, propose un système connexionniste métaheuristique appelé « réseau neuroglial artificiel » "ANGN", composé d'un réseau neuronal en parallèle avec un réseau glial, capable d'émuler le traitement de l'information dans le cerveau humain d'une manière presque parfaite. Etant donné que les deux sous réseaux (neuronal et glial) possèdent des dynamiques différentes (rapide et lente), l'architecture globale du système est basée sur le puissant concept de « modularité » et une mise en œuvre, utilisant les systèmes à multi - échelles de temps. Ces systèmes, qualifiés généralement de «systèmes singulièrement perturbés » sont des modèles mathématiques non-linéaires, fortement couplés présentant des dynamiques différentes. Les modèles mathématiques exactes de ces systèmes sont généralement difficiles à obtenir en raison des erreurs connexes et des non-linéarités.

Une validation de l'approche mise en œuvre est obtenue dans le cas de deux applications différentes : « Identification d'un moteur à courant continu (MCC) et d'une machine asynchrone (MAS), mis sous la forme standard singulièrement perturbée ». Pour chaque application, le modèle global est décomposé en sous-modèles : lents et rapides suivant la méthode de décomposition des perturbations singulières. Cette méthode a pour principe de base, la conservation du modèle global utilisé et la réduction de son ordre, en découplant les grandeurs physiques mises en jeu pour aboutir à des sous-modèles d'ordre réduit. La méthode géométrique des cercles de Gerschgorin est utilisée pour surmonter le problème de la définition des variables lentes et rapides, lors du découplage. Enfin, deux algorithmes différents, sont développés pour tester l'efficacité et le rendement du réseau neuroglial ainsi défini.

Cette nouvelle architecture a permis d'obtenir des réseaux de taille considérablement réduite avec moins de complexité et de meilleures performances en terme d'erreur moyenne quadratique et vitesse de convergence, comparativement aux modèles de réseau de neurones classiques, modulaires à deux réseaux de neurones et Mélange d'Experts (ME) et aux modèles de réseaux de neurones à multi-échelles de temps.

Mots clés:

Réseau glial, Réseau Neuroglial artificiel, Méthode des perturbations singulières, Méthode des cercles de Gerschgorin, Modularité, Systèmes à multi-échelles de temps.

ABSTRACT

In this work, a new connectionist system (metaheuristic) called "artificial neuroglial network" "ANGN" is proposed. This network consists of a neural network in parallel with a glial network. It emulates the processing of information in the human brain in an ameliorated way. The two sub networks (neuronal and glial) possess different dynamics (fast and slow). We opted for a global network architecture, based on the powerful concept of "modularity" and an implementation composed of multi-time scale sub-systems. These systems are usually labeled as "singularly perturbed". These systems are nonlinear, strongly coupled mathematical models with multi-dynamics. Precise and accurate mathematical models for these systems are generally difficult to obtain due to uncertainties and nonlinearities.

A validation of the approach is carried out for two different applications: «Identification of a DC motor and an asynchronous machine (ASM) put in the singularly perturbed standard form ». For each application, the original model is decomposed into sub-models: slow and fast and the decomposition method chosen is that of singular perturbations. This method has as a basic principle: from the global model a reduction of its order is accomplished, decoupling the physical quantities involved, to lead to reduced order sub-models. The Gerschgorin geometric method of circles is used to solve the problem of identifying slow and fast variables during this decoupling. Two different algorithms are developed to test the efficiency and the performance of our neuroglial network thus defined.

This new architecture has made it possible to obtain networks of considerably reduced size with less complexity and better performances in terms of squared mean error and convergence speed, compared to conventional neural network models, which use modularity with two neural networks as a Mixture of experts (ME), as well as to other multi-time scale neural networks models.

Keywords:

Glial network, Artificial Neuroglial network, Singular perturbation method, Gerschgorin circle theorem, Modularity, Multi-time scale systems.

REMERCIEMENTS

Le présent travail a été réalisé dans le cadre d'une thèse de Doctorat au sein du Laboratoire de Labset, Département d'Electronique, Faculté de Technologie de l'Université Saad Dahlab de Blida 1.

Je tiens tout particulièrement à remercier :

Monsieur Hadj sadok M'hamed, Maitre de conférence à l'université Saad Dahlab de Blida 1, et directeur de cette thèse. Je le suis reconnaissante pour sa disponibilité, ses suggestions pertinentes et ses précieux commentaires au cours de mes travaux.

Monsieur Guessoum Abderrezek , professeur à l'université Saad Dahlab de Blida 1, et co-directeur de cette thèse, pour ses remarques pertinentes et constructives , qui m'ont été très utiles. Permettez-moi Monsieur de vous exprimer mes reconnaissances et mes remerciements les plus sincères.

Que Monsieur KARA Kamel, professeur à l'université Saad Dahlab de Blida 1, trouve ici l'expression de mes plus vifs remerciements pour avoir accepté de présider le jury d'évaluation.

. Mes grands remerciements vont aussi à :

Monsieur ABOUDERRA Salam, professeur à l'université de Médéa qui a accepté de participer au jury de soutenance. Qu'il soit remercié chaleureusement.

Monsieur OULD ZMIRLI Mohamed, professeur à l'université de Médéa, d'avoir accepté de juger et d'enrichir ce modeste travail.

Monsieur BOUDJEMAI Fares, professeur à L'ENP, pour m'avoir fait l'honneur de participer au jury de soutenance de cette thèse.

Je tiens à exprimer ma profonde gratitude à tous les membres du jury, qui m'ont fait l'honneur d'examiner ce travail.

TABLE DES MATIERES

| | |
|-----------------------------------------------------------------------------------------------------------------|-----------|
| INTRODUCTION GENERALE..... | 13 |
| 1. LES CELLULES GLIALES BIOLOGIQUES | 20 |
| 1.1. Introduction | 20 |
| 1.2. Cellules gliales biologiques | 21 |
| 1.2.1. Classification | 21 |
| 1.2.2. Le processus de fonctionnement des astrocytes | 23 |
| 1.2.3. La communication entre les neurones et les astrocytes | 25 |
| 1.3. Conclusion | 27 |
| 2. METHODES DES PERTURBATIONS SINGULIERES ET LES CERCLES DE GERSCHGORIN | 28 |
| 2.1. Introduction | 28 |
| 2.2. Méthode des perturbations singulières appliquée aux modèles continus linéaires..... | 29 |
| 2.2.1. Modèles réduits lent et rapide..... | 30 |
| 2.2.2. Application..... | 32 |
| 2.3. Identification géométrique des dynamiques | 34 |
| 2.3.1. La méthode des Cercles de Gerschgorin | 34 |
| 2.3.2. Application | 35 |
| 2.3.3. Transformations modales | 36 |
| 2.3.4. Algorithme de séparation des dynamiques | 37 |
| 2.3.5. Regroupement et séparation des dynamiques d'un système linéaire | 41 |
| 2.3.6. Application | 43 |
| 2.4. Conclusion | 46 |
| 3. MODULARITE ET ARCHITECTURE NEUROGLIALE..... | 47 |
| 3.1. Introduction | 47 |
| 3.2. Comment introduire les connaissances des cellules gliales dans les réseaux de neurones artificiels..... | 48 |
| 3.3. Modularité | 50 |
| 3.4. Modularité dans le système nerveux..... | 51 |
| 3.5. Modularité dans le réseau de neurones artificiel | 52 |

| | |
|----------------------------------------------------------------------------------------------------|------------|
| 3.5.1. Décomposition..... | 55 |
| 3.5.2. Organisation de l'architecture modulaire..... | 58 |
| 3.5.3. Nature de l'apprentissage..... | 59 |
| 3.5.4. Communication entre les modules..... | 60 |
| 3.6. Architecture du réseau neuroglial artificiel "ANGN" | 61 |
| 3.6.1. Premier algorithme | 63 |
| 3.6.2. Deuxième algorithme..... | 65 |
| 3.7. Conclusion | 74 |
| 4. APPLICATION AUX SYSTEME SINGULIEREMENT PERTURBES..... | 76 |
| 4.1. Introduction | 76 |
| 4.2. Identification du moteur à courant continu en utilisant le second algorithme proposé | 77 |
| 4.2.1. Séparation des modes électromagnétiques..... | 78 |
| 4.2.2. Simulation..... | 81 |
| 4.3. Modélisation de la machine asynchrone par la métaheuristique "ANGN" ... | 87 |
| 4.3.2. Mise sous forme standard singulièrement perturbée | 92 |
| 4.3.3. Résultats des tests. | 97 |
| 4.3.4. Premier algorithme d'apprentissage..... | 98 |
| 4.3.5. Deuxième algorithme d'apprentissage (réseau glial softmax). | 101 |
| 4.3.6. Comparaison entre les différentes architectures | 105 |
| 4.4. Identification de la machine asynchrone | 110 |
| 4.5. Conclusion | 115 |
| CONCLUSION GENERALE | 116 |
| A : Méthode des perturbations singulières appliquée aux systèmes non- linéaires | 119 |
| B : Reseaux de neurones artificiels..... | 124 |
| C : Transformation de park | 140 |
| D : Les commandes MATLAB des Ondelettes..... | 143 |
| E : Données relatives aux machines asynchrones utilisées en simulation..... | 144 |
| Références..... | 145 |

LISTE DES FIGURES ET DES TABLEAUX

| | |
|-------------------------------------------------------------------------------------------------------------|-----|
| Figure 1.1 : Cellules gliales | 21 |
| Figure 1.2 : Astrocyte [3]..... | 23 |
| Figure 1.3 : La communication entre les neurones et les astrocytes [3]..... | 25 |
| | |
| Figure 2.1 : Forme globale issue de l'équation (2.4) | 33 |
| Figure 2.2 : Schéma découplé issue des équations (2.6) (2.9) | 33 |
| Figure 2.3 : Cercles de Gerchgorin pour le modèle linéaire (2.12)..... | 36 |
| Figure 2.4 : Cercles de Gerschgorin pour le modèle (2.22) | 40 |
| Figure 2.5 : Cercles de Gerschgorin pour le modèle (2.24) | 41 |
| Figure 2.6 : Cercles de Gerschgorin du modèle 2.30..... | 4.2 |
| | |
| Figure 3.1 : Architectures de la modularité (a) modularité à modèle multiple (b) modularité vraie..... | 4.4 |
| Figure 3.4 : Exemple de décomposition verticale..... | 57 |
| Figure 3.5 : Architecture parallèle | 58 |
| Figure 3.6: Architecture séquentielle..... | 59 |
| Figure 3.7: Architecture du réseau neuroglial du 1 ^{er} algorithme..... | 62 |
| Figure 3.8 : Architecture du réseau neuroglial du 2 ^{ème} algorithme. | 63 |
| Figure 3.9 : Architecture du réseau expert..... | 66 |
| Figure 3.10: Architecture du réseau glial..... | 68 |
| | |
| Figure 4.1 : Schéma d'identification "ANGN" | 77 |
| Figure 4.2 : Cercles de Gerschgorin pour le modèle 4.6 du MCC..... | 79 |
| Figure 4.3 : Cercles de Gerschgorin pour le modèle 4.7 du MCC..... | 80 |
| Figure 4.4 : Cercles de Gerschgorin pour le modèle 4.8 du MCC..... | 81 |
| Figure 4.5 : Résultat d'identification de r : 2 ^{ème} algorithme proposé de "ANGN"..... | 83 |
| Figure 4.6 : Erreur d'identification de r : 2 ^{ème} algorithme proposé de "ANGN"..... | 83 |
| Figure 4.7 : Résultat d'identification de i_1 : 2 ^{ème} algorithme proposé de "ANGN"..... | 84 |
| Figure 4.8 : Erreur d'identification de i_1 : 2 ^{ème} algorithme proposé de "ANGN" | 84 |
| Figure 4.9 : (a) Résultat d'identification de r réf.[17] (b) Erreur d'identification de réf. [17]..... | 85 |
| Figure 4.10 :(a) Résultat d'identification i_r réf.[17] (b) Erreur d'identification réf.[17] | 85 |

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Figure 4.11 : Cercles de Gerschgorin pour différentes vitesses..... | 91 |
| Figure 4.12 : (a) Le couple rapide (b) Le couple lent..... | 96 |
| Figure 4.13 : Le couple global électromagnétique..... | 96 |
| Figure 4.14 : Superposition du couple du modèle réel simulé et du couple global estimé à partir des modèles réduits lent et rapide..... | 97 |
| Figure 4.15 : Evolution des EQM _s (a)Réseau glial (b) Réseau expert..... | 100 |
| Figure 4.16 : L'évolution des EQM _s du 2 ^{ème} algorithme (a) glial (b) expert..... | 102 |
| Figure 4.17 : Les couples estimés par le 2 ^{ème} algorithme | 103 |
| Figure 4.18 :Le couple global du réseau "ANGN" estimé par le 2 ^{ème} algorithme | 103 |
| Figure 4.19 : Evolution du EQM du réseau de neurones unique..... | 105 |
| Figure 4.20 : Architecture modulaire à deux réseaux..... | 106 |
| Figure 4.21: Evolution de l'EQM du modèle à deux réseaux de neurones..... | 107 |
| (a) Réseau lent (b) Réseau rapide | 107 |
| Figure 4.22: Evolution des EQM _s des experts du réseau ME | 108 |
| Figure 4.24 L'erreur de la variable d'état | |
| Figure 4.25: L'erreur de la variable d'état φ | 112 |
| (a) 2 ^{ème} algorithme proposé (b) réf. [17]..... | 112 |
| Figure 4.26 : L'erreur de la variable d'état φ | 112 |
| (a) 2 ^{ème} algorithme proposé (b) réf. [17]..... | 113 |
| Figure 4.27 : L'erreur de la variable d'état $i\alpha$, | 113 |
| (a) 2 ^{ème} algorithme proposé (b) réf. [17]..... | 112 |
| Figure 4.28 : L'erreur de la variable d'état $i\mathbf{l}$, | 113 |
| (a) 2 ^{ème} algorithme proposé (b) réf. [17]..... | 112 |
| | |
| Figure B.1: Neurone biologique [65]..... | 125 |
| Figure B.2 : Modèle mathématique | 127 |
| Figure B.3 : Fonction d'activation binaire à seuil. | 128 |
| Figure B.4 : Fonction d'activation à rampe avec saturation..... | 129 |
| Figure B.5 : Fonction d'activation sigmoïde. | 129 |
| Figure B.6 : Réseau dynamique et statique | 130 |
| Figure B.7 : Réseau de neurones multicouches..... | 131 |
| Figure B.8 : Réseau de neurones mono-couche..... | 132 |

| | | |
|---------------|-------------------------------------------------------------------------------------|-----|
| Table 3.1 : | Premier algorithme d'apprentissage du réseau neuroglial | 64 |
| Table 3.2 : | Deuxième algorithme d'apprentissage du réseau neuroglial..... | 73 |
| Tableau 4.1 : | Les valeurs RMSE pour les variables d'états r et i du MCC... | 86 |
| Tableau 4.2 : | Performances du réseau "ANGN" relatives au premier algorithme..... | 100 |
| Tableau 4.3 : | Poids et performances des experts pour chaque domaine de validité des données | 104 |
| Tableau 4.4: | Les valeurs RMSE des variables d'état.de la MAS | 115 |

LISTE DES ABREVIATIONS

| | |
|------------------|-----------------------------------------|
| CG | Cellule Gliale. |
| RN | Réseau de neurones |
| RNA | Réseau de neurones Artificiel. |
| RNA _s | Réseaux de neurones Artificiels |
| ANGN | Artificiel Neuroglial network |
| TPS | Technique des perturbations singulières |
| TCG | Technique des cercles de Gerchgorin |
| SSP | Système singulièrement perturbé |
| SSP _s | Systèmes singulièrement perturbés |
| ME | Mélange d'experts (Mixture of experts) |
| MLP | Multiple layer perceptron |
| BP | Back propagation |
| TSP | Travelling Salesman Problem |
| DNN | Dynamic neural network |
| RNN | Recurrent neural network |
| EKF | Extended kalman filter |
| RMS | Root mean square |
| EQM | Erreur quadratique minimum |
| OBE | Optimal bounded ellipsoid |

INTRODUCTION GENERALE

Dans ce travail, nous nous intéressons à une nouvelle réalité biologique découverte récemment, appelée « Neurogliale ». Un intérêt particulier est donné aux «cellules neuronales » et aux « cellules gliales » et parmi elles « les astrocytes », comme moyen de traitement de l'information. En effet plusieurs études ont montré que les astrocytes, qui constituent une deuxième variété de cellules cérébrales, forment un réseau, parallèle au réseau de neurones [1-5]. Ce réseau possède des capacités très intéressantes et notre cerveau n'aurait donc pas une, mais deux manières de faire circuler l'information [1-5]. De ce fait, sur la voie nerveuse (réseau de neurones), les informations circulent 100 000 fois plus vite que sur la voie astrocytaire (réseau glial) [1, 2]. Donc le réseau global qu'on appellera désormais « réseau neuroglial » possède deux dynamiques différentes.

En outre, les « cellules gliales » (CG) et parmi elles, les astrocytes, classiquement considérées comme des cellules de soutien pour les neurones, sans aucun rôle dans le traitement de l'information par le cerveau, apparaissent, comme des éléments pertinents dans la physiologie du cerveau à travers leur capacité à réguler l'activité neuronale, la transmission synaptique et la plasticité du réseau. Les astrocytes possèdent aussi, le moyen de faciliter, de ralentir et même bloquer le flux d'informations circulant [1, 2].

De nombreux systèmes industriels sont non linéaires avec des états dynamiques «lents» et «rapides» en raison de la présence de certains paramètres «parasites» tels que petites constantes de temps, résistances, inductances, capacités, masses et moments d'inertie [6]. Ces phénomènes se retrouvent dans les systèmes harmoniques d'entraînement [7], les systèmes de contrôle de la trajectoire des vols d'essais [8], les manipulateurs de liaison flexible [9], les systèmes d'alimentation [10], etc. Ces systèmes sont généralement qualifiés de «singulièrement perturbés» (SSP) ou « multi - échelles de temps ». Par exemple,

dans le système de moteur à courant continu, le courant est la dynamique rapide tandis que la vitesse angulaire est la dynamique lente.

En raison de l'existence d'états dynamiques rapides, les systèmes singulièrement perturbés ont généralement des ordres élevés, ce qui augmente considérablement les difficultés de modélisation, d'analyse et de conception de contrôleurs. Un moyen simple de réduire l'ordre du système consiste à négliger les états dynamiques rapides. Cependant, une conception basée sur un modèle simplifié peut entraîner un système moins performant.

Un moyen efficace de surmonter ce problème consiste à séparer les états du système d'origine en ceux qui changent rapidement et ceux qui changent lentement à l'échelle de temps choisie en utilisant une technique des perturbations singulières (TPS).

Quelques résultats de recherche récents utilisant la TPS pour analyser et contrôler les SSP_s sont publiés dans [11-15]. Cependant, des modèles mathématiques précis et fidèles pour ces systèmes sont généralement difficiles à obtenir en raison des incertitudes et des non-linéarités. Dans ce cas, une identification adéquate du système devient importante et nécessaire avant de pouvoir concevoir un schéma de commande basé sur une théorie de perturbation singulière.

Récemment, des recherches utilisant des réseaux de neurones à multi-échelles de temps ont été proposées dans la littérature pour résoudre le problème d'identification des SSP_s non linéaires. Parmi ces recherches, nous citons celle des réseaux de neurones dynamiques à multi-échelles de temps (DNN) proposés dans [16-19], et le réseau de neurones récurrent (RNN) proposé dans [20, 21]. Cependant, dans les articles cités, les méthodes d'apprentissage sont basées sur un algorithme de mise à jour de descente de gradient avec un «gain d'apprentissage» fixe, tel que la back propagation (BP).

Le principal inconvénient de ces méthodes d'entraînement est que la vitesse de convergence est généralement très lente. Pour accélérer le processus d'apprentissage d'un réseau de neurones, les chercheurs ont étudié des méthodes basées sur le filtre de Kalman étendu (EKF) proposées dans [22-24]. L'analyse théorique de l'algorithme d'entraînement basé sur EKF nécessite que

l'incertitude de modélisation du RNA soit un processus gaussien, ce qui peut ne pas être vrai dans des applications réelles. D'autres chercheurs ont également étudié les lois d'apprentissage optimales d'un réseau de neurone basées sur l'algorithme OBE (Optimal bounded ellipsoïde) [25-29]. Toutes ces méthodes complexes nécessitent beaucoup de paramètres et beaucoup de calcul.

Dans ce travail, un système connexionniste (métaheuristique) appelé « réseau neuroglial artificiel » "ANGN" est proposé pour résoudre les problèmes liés à l'apprentissage des systèmes singulièrement perturbés. Ce réseau composé d'un réseau neuronale en parallèle avec un réseau glial, émule le traitement de l'information dans le cerveau humain d'une manière presque parfaite. Les deux sous réseaux (neuronale et glial) possédant des dynamiques différentes (rapide et lente); nous avons opté pour une architecture du réseau global, basée sur le puissant concept de « modularité » et une mise en œuvre, utilisant les systèmes à multi - échelles de temps. Ces systèmes sont des modèles mathématiques non-linéaires, fortement couplés présentant des dynamiques différentes.

Pour une application donnée, selon la complexité et les caractéristiques physiques du problème, on décompose notre modèle global en sous- modèles : lents et rapides. La méthode de décomposition adoptée est la méthode des perturbations singulières [30-33]. Cette méthode a pour principe de base, la conservation du modèle global utilisé, la réduction de son ordre, en découplant les grandeurs physiques mises en jeux, pour aboutir à des sous modèles d'ordre réduit.

La première difficulté qui apparaît lors du découplage des variables, est l'identification des variables lentes et des variables rapides dans le modèle, d'où la technique géométrique des cercles de Gerschgorin (TCG) [33-36]. Cette technique permet de localiser les valeurs propres dans le plan complexe dans de 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.

Une validation de l'approche a été effectuée sur deux modèles : le modèle du moteur à courant continu et le modèle de la machine asynchrone, misent sous forme standard singulièrement perturbée ; la technique géométrique des cercles de Gerschgorin a été appliquée pour le découplage des variables lentes et rapides. Deux algorithmes différents, ont été adoptés pour tester l'efficacité et le rendement de notre réseau neuroglial ainsi défini.

Cette nouvelle architecture a permis d'obtenir des réseaux de taille considérablement réduite avec une meilleure performance, possédant une forte capacité d'approximation non linéaire et permettant de modéliser plus précisément des systèmes singulièrement perturbés avec moins de complexité, une erreur moyenne quadratique minimale et une grande vitesse de convergence, comparativement aux réseaux de neurones classiques, modulaires (avec deux réseaux de neurones ou mélange d'experts « ME ») et autres recherches de réseau de neurones à multi-échelles de temps.

▪ **Contributions originales**

La première contribution originale de cette thèse s'appuie sur la conception et la validation d'une nouvelle approche méta-heuristique Neurogliale "ANGN" basée sur le puissant concept modularité et le développement de deux algorithmes.

La seconde contribution originale de cette thèse exploite sur la modélisation et l'identification des systèmes non-linéaires singulièrement perturbés, en améliorant la précision et la vitesse de convergence par l'application de la nouvelle approche "ANGN".

▪ **Publications**

- ▶ Samia Salah, M'hamed HadjSadok, Abderrezak Guessoum "A Novel Neuroglial Architecture for Modelling Singular Perturbation System ",

International Journal of Electrical and Computer Engineering Volme 8, No. 6, December 2018, pp. 4810~4822 ISSN: 2088-8708,
doi : [10.11591/ijece.v8i6.pp.4810-4822](https://doi.org/10.11591/ijece.v8i6.pp.4810-4822).

- ▶ Samia Salah, M'hamed HadjSadok, Abderrezak Guessoum "Application New Approach with two Networks Slow and Fast on the Asynchronous Machine",
World Academy of Science, Engineering and Technology, Volume7, N°8, pp.1087-1091, 2013. doi : [10.5281/zenodo.1088080](https://doi.org/10.5281/zenodo.1088080)
URL <https://www.waset.org/publication/16914>.

▪ **Conférences**

- ▶ Samia Salah, M'hamed HadjSadok, Abderrezak Guessoum "Development of the New Neuroglial Approach: Application on Singularly Perturbed Systems", International Conference on Computer and Information Technology (WCCIT'2013) Sousse, Tunisie, Juin 2013.
- ▶ Samia Salah, M'hamed HadjSadok, Abderrezak Guessoum "Application New Approach with two Networks Slow and Fast on the Asynchronous Machine", World Academy of Science, Engineering and Technology, International Conference on Artificial Intelligence and Neural Networks (ICAINN'2013) France, Paris, August 2013.

▪ **Organisation de la thèse**

La suite de la thèse est organisée de la façon suivante:

Le premier chapitre décrit le fonctionnement biologique des cellules gliales ainsi leur rôle majeur pour le contrôle et la modulation des activités neuronales.

La première partie du chapitre 2 est dédié à la présentation de la méthode des perturbations singulières. Nous avons détaillé cette méthode pour les systèmes linéaires et non linéaires dans (annexe A). 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 dans la deuxième partie, 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 chapitre 3 traite de la question générale de l'introduction des dernières connaissances acquises sur les cellules gliales biologiques dans les réseaux de neurones artificiels, qui aboutissent 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 sont développés.

Les travaux présentés dans le chapitre 4 font l'objet de deux grandes parties. La première partie concerne la mise sous la forme standard singulièrement perturbée des systèmes à multi-échelles de temps 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.

Dans une première application, nous avons utilisé l'approche "ANGN" pour identifier les grandeurs (ω, τ) d'un moteur à courant continu. Les résultats de simulations ainsi obtenus sont comparés à d'autres résultats de recherche récents présentés dans [17]. La seconde application traitée dans ce chapitre concerne la modélisation et l'identification de la machine asynchrone. Les résultats de la modélisation obtenus sont comparés aux résultats obtenus respectivement avec une architecture classique puis avec une architecture modulaire (à deux réseaux de neurones et avec l'architecture de mélange d'experts).

Les résultats obtenus de l'identification des grandeurs $(\omega, \varphi_r, \varphi_r, i_{\alpha}, i_{\beta})$ de la machine asynchrone sont comparés à d'autres résultats de recherche récents présentés dans [17].

Pour clôturer ce travail, une conclusion générale a été donnée pour avancer quelques remarques et présenter d'éventuelles perspectives quant à la continuation de ce travail.

L'étude de la configuration du réseau de neurone artificiel, et de ses principales architectures ainsi les paradigmes d'apprentissage sont décrits en annexe B. Les annexes contiennent aussi la transformation de PARK (annexe C), les paramètres des machines asynchrones utilisées (annexe D) et les commandes MATLAB des ondelettes (annexe E).

Chapitre 1

LES CELLULES GLIALES BIOLOGIQUES

1.1. Introduction

Pour pouvoir créer un réseau de neurones artificiel (RNA) qui émule le cerveau on se doit comprendre la structure biologique et le fonctionnement 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, la quelle regroupe 10 milliards de cellules composant un immense réseau où se produisent des milliers de milliards d'interactions [1-5]. 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 [1-5].

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 [1-5]. 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, 3-5] .

Les cellules gliales furent découvertes en 1904 par Santiago Ramón y Cajal. Le fait fut découvert au début du XX^e siècle. 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 [1-5] .

Le but de ce chapitre est de présenter quelques notions générales sur les cellules gliales afin de préciser leurs rôles dans 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) [4, 5].

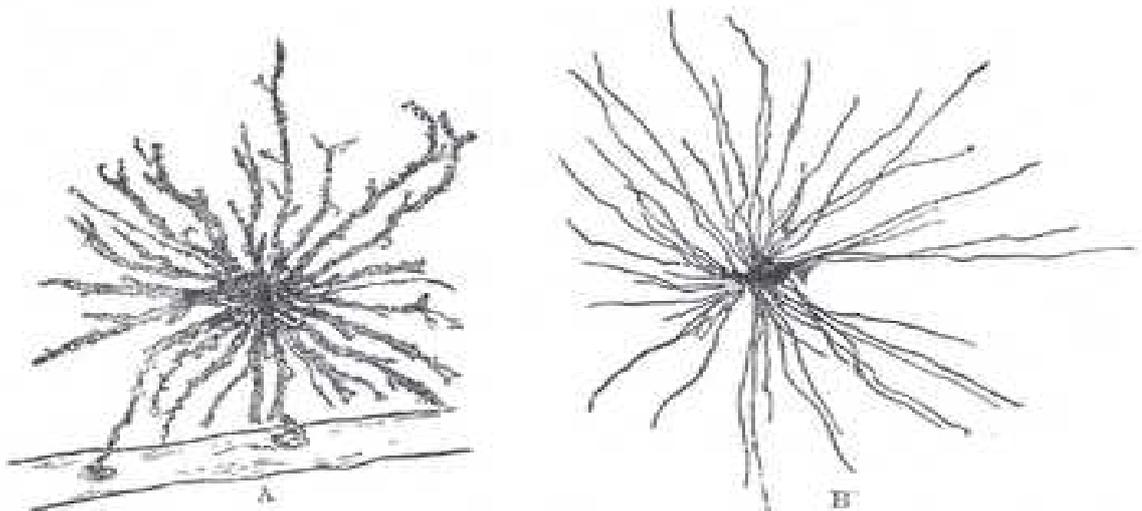


Figure 1.1 : Cellules gliales

1.2.1. Classification

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

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 [3-5], Les cellules microgliales constituent la première ligne de la défense contre tout envahissement de corps 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 elles ont des fonctions analogues à certaines cellules gliales du système nerveux central [3-5].

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 [3-5].

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 [3-5]. Mais de toutes les cellules gliales, l'astrocyte est sans doute celui qui a les fonctions les plus complexes [3-5].

1.2.1.4. Les astrocytes

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 [3-5] :

- 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.

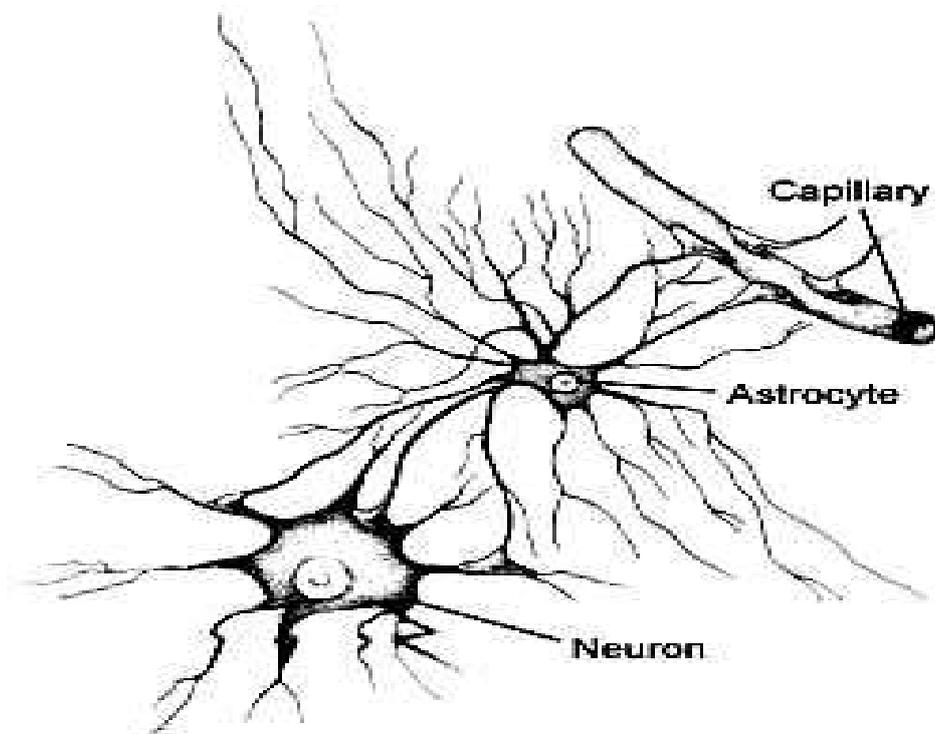


Figure 1.2 : Astrocyte [3]

1.2.2. Le processus de fonctionnement des astrocytes

Les neurologues s'interrogent depuis longtemps sur le rôle des cellules gliales situées dans l'espace inter neuronal et constituent une part importante de la masse du cerveau. Ils ont 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 [1].

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

A- 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 [1].

B- 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 entoure [1].

C- Ils vont modifier l'influx nerveux :

Les astrocytes peuvent être activés par un neurone, soit par eux même. En modifiant les messages entre neurones de trois manières différents [1] :

- 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 [1].

D - 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 [1].

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 [3, 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 [3, 4].

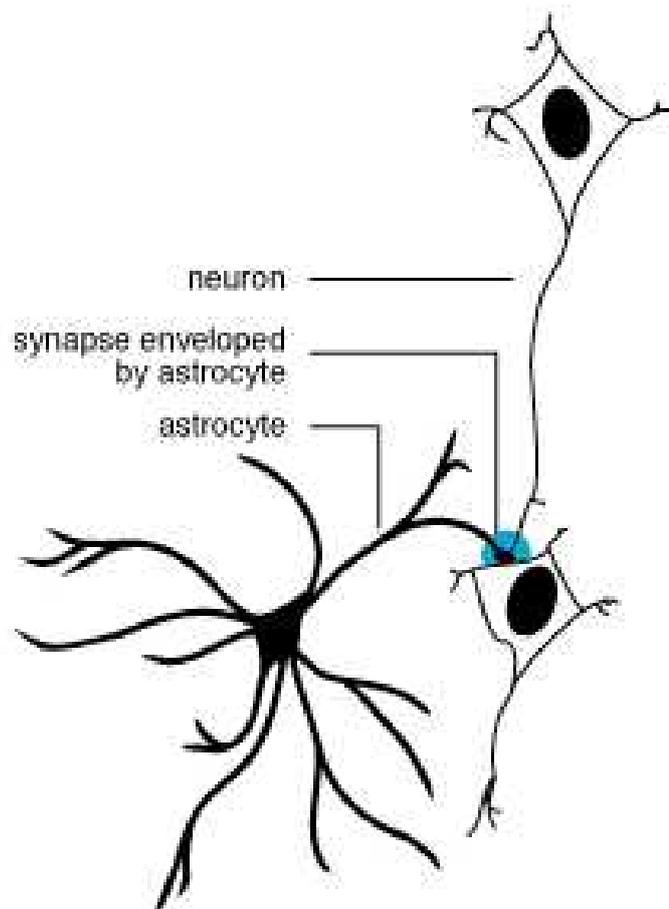


Figure 1.3 : La communication entre les neurones et les astrocytes [3]

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 [3-5] .

Proportion cellules gliales : neurone [3-5] .

- 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 [3-5] .

Les dernières avancées scientifiques montrent 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 permet 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 [3-5] .

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 [3-5] .

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 [1]. 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.

Chapitre 2

METHODES DES PERTURBATIONS SINGULIERES ET LES CERCLES DE GERSCHGORIN

2.1. Introduction

Lorsque les dynamiques des grandeurs physiques d'un modèle sont très différentes, on utilise la méthode des perturbations singulières. Le principe de cette méthode consiste à découpler ces grandeurs sur la base de leurs constantes de temps respectives. Les éléments présentant une faible constante de temps possèdent des dynamiques rapides alors que ceux dont la constante de temps est élevée ont des dynamiques lentes. On dira que l'élément dynamique rapide se comporte comme un élément parasite qui augmente la dynamique du modèle. On notera que la séparation des dynamiques dépend des valeurs numériques des paramètres du modèle. Cette méthode des perturbations singulières est valable pour des modèles linéaires, non-linéaires, continus ou discrets [37].

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 ν des vitesses des dynamiques lentes et rapides tend vers zéro [37]. Quand ν 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 utilisation dans le domaine de la modélisation et contrôle des machines est très récente. La mise en œuvre de cette théorie dans ce domaine a connu un développement très rapide.

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.

2.2. Méthode des perturbations singulières appliquée aux modèles continus linéaires

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

$$\begin{cases} \dot{x} = f(x, z, u, t, v) & x \in \mathfrak{R}^n \\ v \dot{z} = g(x, z, u, t, v) & z \in \mathfrak{R}^m \end{cases} \quad (2.1)$$

Considérons le modèle d'état d'un système linéaire de dimension n :

$$\begin{cases} \dot{X} = A X + B U \\ Y = C X \end{cases} \quad (2.2)$$

Evoluant suivant deux échelles de temps, et pouvant être découpé en deux sous-systèmes lent et rapide. Le vecteur d'état X contient toutes les variables d'état correspondant aux éléments dynamiques. Si x est l'ensemble des variables d'état des éléments lents, et z celui des éléments rapides[35, 38] . Le modèle s'écrit :

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} U, \quad (2.3)$$

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

avec :

$$-x(t_0) = x_0, \quad e \quad z(t_0) = z_0$$

$-A_{21}, A_{22}, B_2$ sont très grands par rapport à ceux de A_{11}, A_{12}, B_1 .

On introduit le paramètre ε pour normaliser notre modèle, on écrira: $A_{21}^* = \varepsilon A_{21}$,

$$A_{22}^* = \varepsilon A_{22}, \quad B_2^* = \varepsilon B_2. \quad \varepsilon \text{ peut être donné par : } v = \|A_{22}^{-1}\|(\|A_0\| + \|A_{12}\| \cdot \|L_0\|)$$

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

en supposant que la matrice A_{22} est inversible, l'équation d'état sous la forme standard singulièrement perturbée avec ε comme paramètre de perturbation, s'écrit alors :

$$\begin{bmatrix} \dot{x} \\ v\dot{z} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21}^* & A_{22}^* \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2^* \end{bmatrix} U, \quad (2.4)$$

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

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

avec :

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

2.2.1. Modèles réduits lent et rapide

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

Le modèle devient :

$$\begin{cases} \dot{x}_s = A_{11}x_s + A_{12}z_s + B_1u_s \\ 0 = A_{21}x_s + A_{22}z_s + B_2u_s \\ y_s = C_1x_s + C_2z_s \end{cases} \quad (2.5)$$

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 :

$$\begin{cases} \dot{x}_s = A_s x_s + B_s u_s \\ y_s = C_s x_s + D_s u_s \\ z_s = -A_{22}^{-1}(A_{21} x_s + B_2 u_s) \end{cases} \quad (2.6)$$

avec

$$\begin{aligned} A_s &= A_{11} - A_{12} A_{22}^{-1} A_{21} \\ B_s &= B_1 - A_{12} A_{22}^{-1} B_2 \\ C_s &= C_1 - C_2 A_{22}^{-1} A_{21} \\ D_s &= -C_2 A_{22}^{-1} B_2 \end{aligned}$$

Si A_{22} est inversible, le modèle (2.6) fournit une bonne approximation du comportement du modèle global, qui sera valable pour $t \geq t_0$, si on fixe : $x_s(t_0) = x_0$ la valeur initiale des composantes lentes z_s vaut :

$$z_s(t_0) = -A_{22}^{-1} A_{21} x_0 \quad (2.7)$$

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

Introduisons le terme correctif z_f , défini par : $z_f = z - z_s$; représentant les variations rapides de z . Donc, l'équation de couche limite, exprimée en temps dilaté † telque :

$$\left(v \frac{dz_f}{dt} = \frac{dz_f}{d\dagger} \right) : \quad (2.8)$$

$$\begin{cases} \frac{dz_f}{d\dagger} = A_{22} z_f(\dagger) + B_2 u_f(\dagger) \\ y_f = C_2 z_f(\dagger) \\ z_f(t_0) = z_0 + A_{22}^{-1} A_{21} x_0 \end{cases} \quad (2.9)$$

est obtenue par la différence entre le modèle initial (2.4) et le modèle simplifié (2.6), en considérant immobiles les variables lentes pendant les transitoires rapides ($\frac{dx_s}{d\dagger} = 0$).

Remarque : la variable $z_f(\dagger)$ 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 Fig 2.2 à partir de la forme initiale présentée par la Fig 2.1 [37].

2.2.2. Application

Considérons le système d'état [37], correspondant à la forme singulièrement perturbée :

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

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

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

Conformément aux équations (2.4) à (2.9), 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**

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

$$y_f = z_f$$

$$z_f(t_0) = 0.8778$$

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

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

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

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

$$\lambda(A_s) \quad \{(-0.0922 + j0.0310), (-0.0922 - j0.0310)\}$$

$$\lambda(A_f) \quad \{-0.9000\}.$$

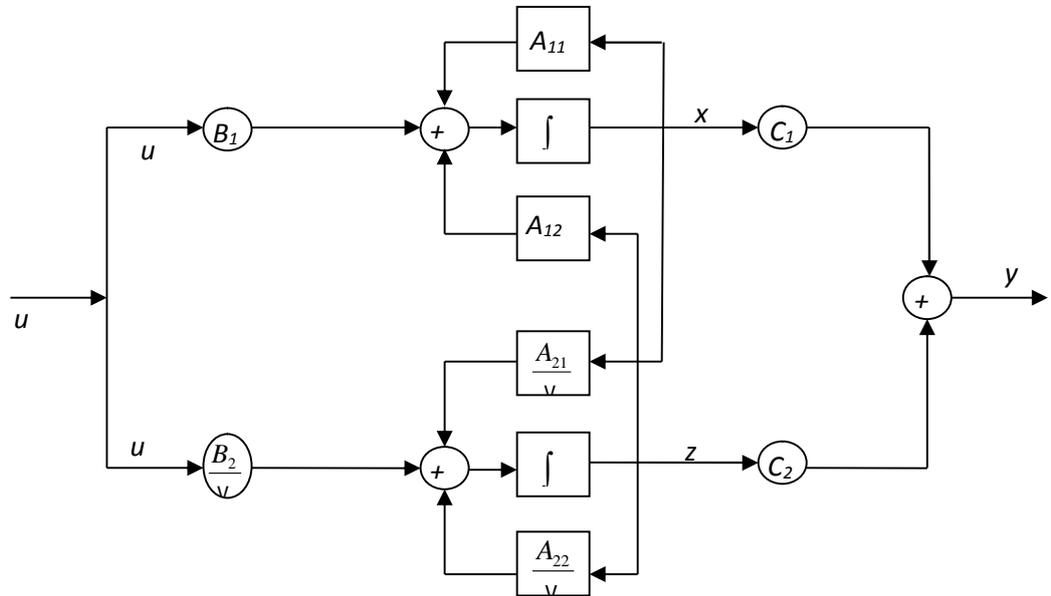


Figure 2.1 : Forme globale issue de l'équation (2.4)

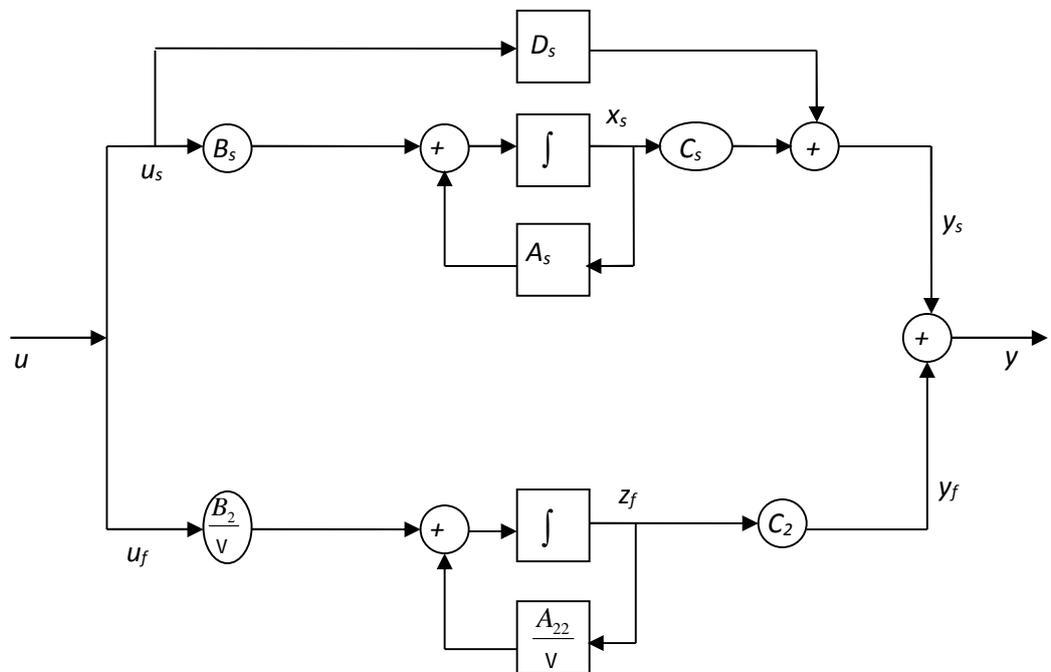


Figure 2.2 : Schéma découplé issue des équations (2.6) (2.9)

2.3 Identification géométrique des dynamiques

La mise sous forme standard suppose au préalable :

- La connaissance des valeurs propres pour la détermination de la taille des vecteurs propres lents et rapides.
- Un regroupement adéquat des modes lents et des modes rapides.

Notre attention c'est portée sur les méthodes géométriques dont celle des cercles de Gerschgorin [33-36]. La localisation des valeurs propres sur le plan complexe permettent de mettre un système sous la forme standard sans devoir calculer ces valeurs propres. Dans le cas des cercles de Gerschgorin, le regroupement des modes est immédiat dès que les valeurs propres sont circonscrites dans des faisceaux de cercles disjoints ; La méthode géométrique des cercles de Gerschgorin pour la sélection et la séparation des différentes échelles de temps est représentée comme suit:

2.3.1. La méthode des cercles de Gerschgorin

Notons $(a_{ij}, i, j = 1, \dots, n)$ les éléments de la matrice carrée d'état A et p_i, Q_i sont exprimées par:

$$p_i = \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|, \quad i = 1, 2, \dots, n \quad (2.10)$$

$$Q_i = \sum_{\substack{i=1 \\ j \neq i}}^n |a_{ij}|, \quad j = 1, 2, \dots, n \quad (2.11)$$

Deux théorèmes dus à Gerschgorin [34] donnent une localisation sur le plan complexe :

▪ Théorème 1

Toutes les valeurs propres d'une matrice de rang n quelconque :

sont contenues dans des faisceaux de n cercles centrés en a_1, a_2, \dots, a_n et de rayons R_{11}, R, \dots, R_i pour les cercles lignes ou $R_{c1}, R_{c2}, \dots, R_c$ pour les cercles

colonne obtenus en faisant la somme des modules des termes hors-diagonale apparaissant dans une même ligne ou dans une même colonne :

$$R_{li} = p_i \quad e \quad R_c = Q_i$$

▪ **Théorème 2**

Lorsqu'un groupe de k cercles-ligne (ou de k cercles-colonne) est complètement disjoint des autres cercles, il contient k valeurs propres [34].

Lorsqu'un groupe de k cercles est complètement disjoint des autres cercles, on peut affirmer que le système possède alors au moins deux échelles de temps. Si ce groupe de cercles est à droite ou à gauche des autres cercles, on peut déterminer les k modes lents correspondant à ces k cercles ou respectivement les k modes rapides. Chaque cercle représentant un état du système, il est alors possible de donner une partition adéquate du modèle.

2.3.2 application

Considérons le modèle suivant [32, 37]:

$$\dot{\tilde{x}} = \begin{bmatrix} -2 & 0 & 0 & 0 & -0.5 \\ 3.493 & -5 & 0 & 0 & 0 \\ 0 & 0.908 & -0.167 & 0 & 0 \\ 0 & 0 & 0.5 & -2 & 0 \\ 0 & 0.273 & 0.047 & 0.28 & -0.1125 \end{bmatrix} \cdot \tilde{x}, \quad (2.12)$$

Description :

- Le tracé des cercles de Gerschgorin est représenté à la figure (2.3).
- 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 $v_s = 0.747$.

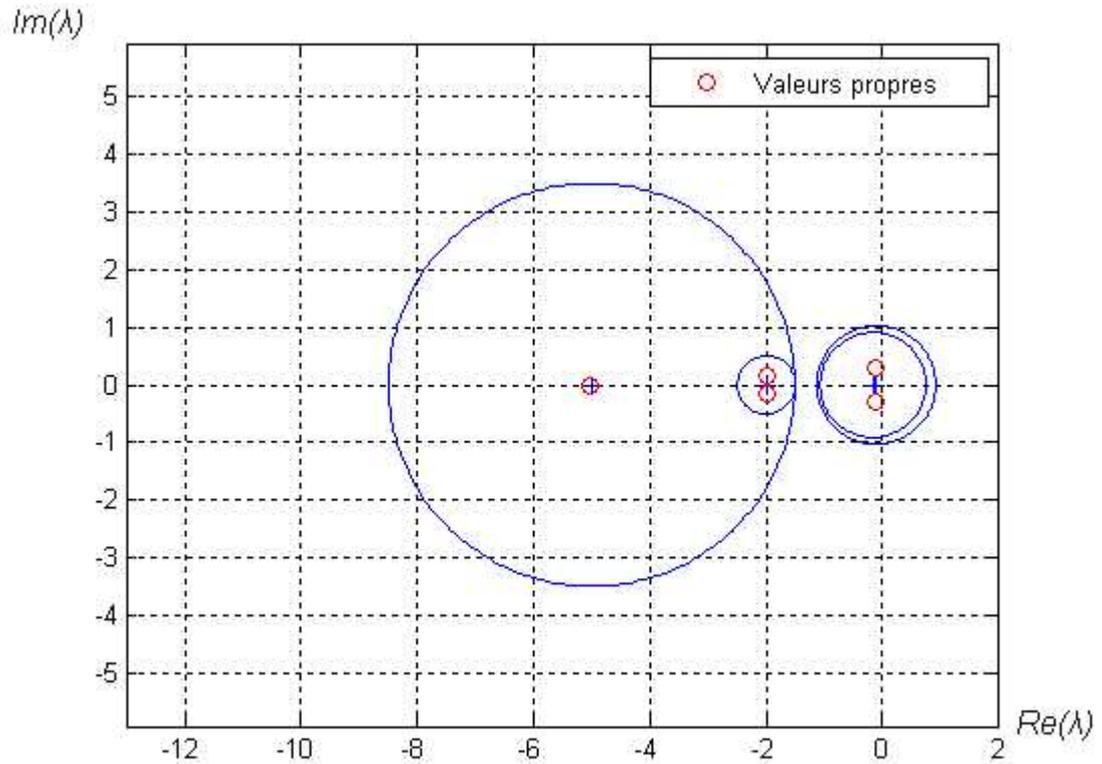


Figure 2.3 : Cercles de Gerchgorin pour le modèle linéaire (2.12)

2.3.3. 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.

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 [33-35]. 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.

2.3.4. Algorithme de séparation des dynamiques

Généralement, cette méthode appliquée directement ne permet pas de conclure immédiatement dans tous les cas. Dauphin-Tanguy [39] propose alors l'emploi d'une transformation:

$S_k = \text{diag}(1, \dots, 1, r_k, 1, \dots, 1)$, $k=1, 2, \dots, n$ sont des paramètres permettant la variation de la taille des cercles et optimise ces paramètres pour obtenir des cercles de rayons minimums. Cette méthode ne permet pas de séparer tous les systèmes.

2.3.4.1. Changement de taille des rayons des cercles

Soit la matrice :

$$S_k = \text{diag}(1, \dots, 1, r_k, 1, \dots, 1), k=1, 2, \dots, n \quad (2.13)$$

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

$$\begin{bmatrix} a_{1,1} & \cdot & \cdot & \cdot & a_{1,k-1} & a_{1,k}/r_k & a_{1,k+1} & \cdot & \cdot & \cdot & a_{1n} \\ \cdot & & & & & \cdot & & & & & \cdot \\ \cdot & & & & & \cdot & & & & & \cdot \\ \cdot & & & & & \cdot & & & & & \cdot \\ a_{k-1,1} & \cdot & \cdot & \cdot & a_{k-1,k-1} & a_{k-1,k}/r_k & a_{k-1,k+1} & \cdot & \cdot & \cdot & a_{k-1,n} \\ a_{k-1}r_k & \cdot & \cdot & \cdot & a_{k,k-1}r_k & a_{k,k} & a_{k,k+1}r_k & \cdot & \cdot & \cdot & a_{k,n}r_k \\ a_{k+1,1} & \cdot & \cdot & \cdot & a_{k+1,k-1} & a_{k+1,k}/r_k & a_{k+1,k+1} & \cdot & \cdot & \cdot & a_{k+1,n} \\ \cdot & & & & & \cdot & & & & & \cdot \\ \cdot & & & & & \cdot & & & & & \cdot \\ \cdot & & & & & \cdot & & & & & \cdot \\ a_{n1} & \cdot & \cdot & \cdot & a_{n,k-1} & a_{n,k}/r_k & a_{n,k+1} & \cdot & \cdot & \cdot & a_{nn} \end{bmatrix} \quad (2.14)$$

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

Si plusieurs cercles se coupent, les termes $B_l (l = 1, 2, \dots)$ se calculent par la même Méthode, la transformation finale est donc :

$$\begin{cases} X = T \underline{X} \\ A = T \underline{A} T^{-1} \end{cases} \quad (2.19)$$

$$\text{avec :} \quad T = \prod T_l, \quad (2.20)$$

Si deux groupes de cercles sont disjoints, la matrice de permutation P trouvée est :

$$\begin{cases} X = P \underline{X} \\ A = P \underline{A} P^{-1} \end{cases} \quad (2.21)$$

▪ **Application**

Considérons le modèle suivant [34]:

$$\dot{x} = \frac{1}{2} \begin{bmatrix} -11 & 9 & -9 \\ 1 & -3 & -1 \\ -8 & 8 & -12 \end{bmatrix} \cdot x + \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \cdot u \quad (2.22)$$

Les rayons de la matrice de base

$$\begin{array}{rcc} & & \text{rayons} \\ A = \frac{1}{2} \begin{bmatrix} -11 & 9 & -9 \\ 1 & -3 & -1 \\ -8 & 8 & -12 \end{bmatrix} & \begin{matrix} 9 \\ 1 \\ 8 \end{matrix} \\ \text{rayons} & 4.5 & 4.5 & 5 \end{array} \quad (2.23)$$

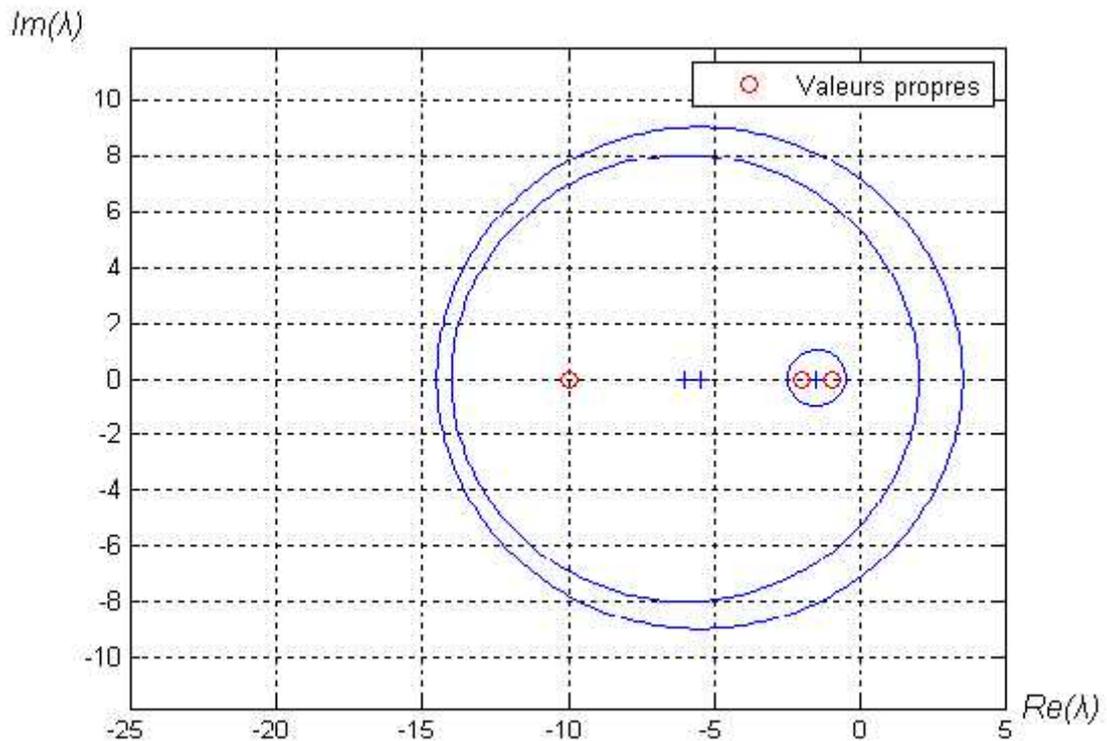


Figure 2.4 : Cercles de Gerschgorin pour le modèle (2.22)

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 :

$$A^* = \frac{1}{2} \begin{bmatrix} -3 & 1 & 0 \\ 1 & -3 & 0 \\ -8 & 8 & -20 \end{bmatrix} \quad \begin{array}{l} \text{rayons} \\ 0.5 \\ 0.5 \\ 8 \end{array} \quad (2.24)$$

rayons 4.5 4.5 0

- Le tracé des cercles de Gerschgorin après la transformation est représenté à la Fig 2.5.

- 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.

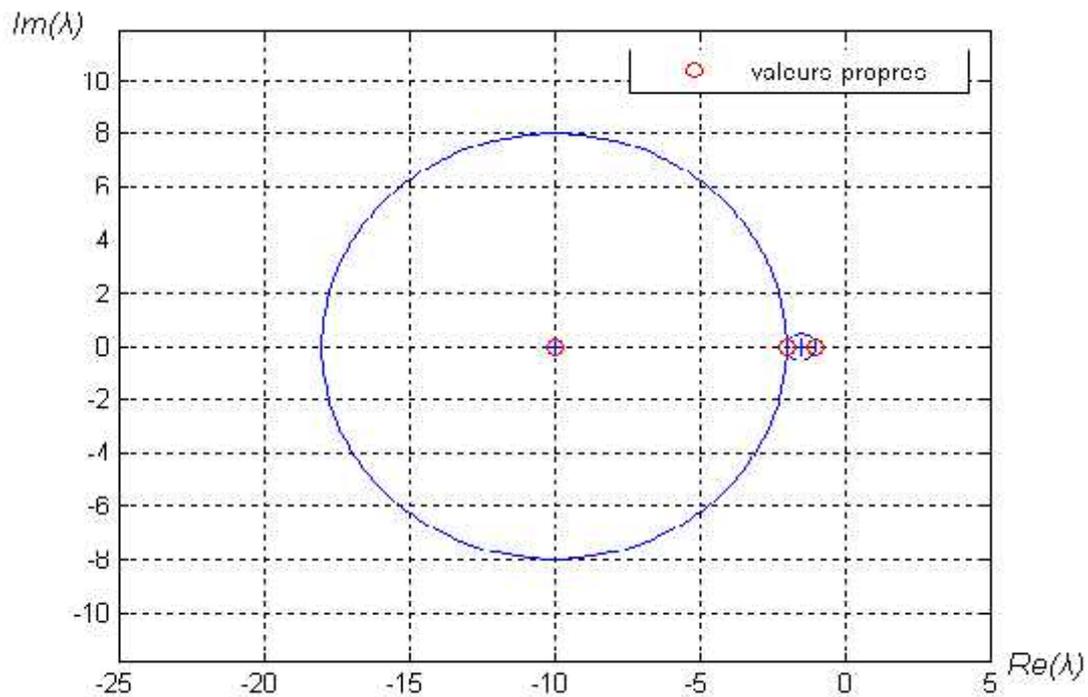


Figure 2.5 : Cercles de Gerschgorin pour le modèle (2.24)

2.3.5. 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 [33].

- 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 :

$$\begin{cases} \tilde{A} = P^T A P \\ \tilde{B} = P^T B \\ \tilde{C} = C P^T \end{cases} \quad (2.25)$$

La matrice P caractérise le changement de lignes correspondant à l'ordre croissant des valeurs propres de A et P' sa transposée. Les éléments de la matrice P seront déterminés comme suit :

$$\begin{cases} P_{ij} = 1, & \text{si la } i^{\text{ième}} \text{ ligne sera transférée à la } j^{\text{ième}}, \\ P_{ii} = 1, & \text{si la } i^{\text{ième}} \text{ ligne ne sera pas à permuter,} \\ P_{ij} = 0, & \text{autrement.} \end{cases} \quad (2.26)$$

- 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 :

$$\tilde{A} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} B_{11} \\ B_{22} \end{bmatrix}, \quad \tilde{C} = [C_1 \quad C_2] \quad (2.27)$$

Le bloc de ligne $C = [A_1 \quad A_1]$ 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.

2.3.6. Application

- Séparation des dynamiques

Considérons le modèle suivant d'ordre 5 [33] :

$$\begin{aligned} \frac{d}{dt} \underline{X} &= \underline{A} \cdot \underline{X} + \underline{B} \cdot U \\ Y &= \underline{C} \cdot \underline{X} \end{aligned} \quad (2.28)$$

avec :

$$A = \begin{bmatrix} -17 & 0.0 & 0.0 & -0.5 & -0.5 \\ -0.1 & -16.5 & -0.4 & 0.0 & -1.0 \\ 0.5 & -0.03 & -2 & -0.02 & -0.05 \\ -0.3 & 0.0 & -0.1 & -16.0 & -0.2 \\ -0.01 & -0.04 & 0.0 & 0.5 & -1.7 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

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

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

$$P = \begin{bmatrix} 0 & 0 & 0. & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (2.29)$$

et la nouvelle représentation d'état est :

$$A = \left[\begin{array}{cc|ccc} -1.7 & 0 & 0.5 & -0.04 & -0.01 \\ -0.05 & -2 & -0.02 & -0.03 & 0.5 \\ \hline -0.2 & -0.1 & -16 & 0 & -0.3 \\ -1 & -0.4 & 0 & -16.5 & -0.1 \\ -0.5 & 0 & -0.5 & 0 & -17 \end{array} \right], \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad (2.30)$$

$$C = [1 \ 1 \ | \ 0 \ 0 \ 1]$$

Le tracé des cercles est représenté sur la Fig 2.6. 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 $z_0 = [0 \ 0 \ 0]^T$.

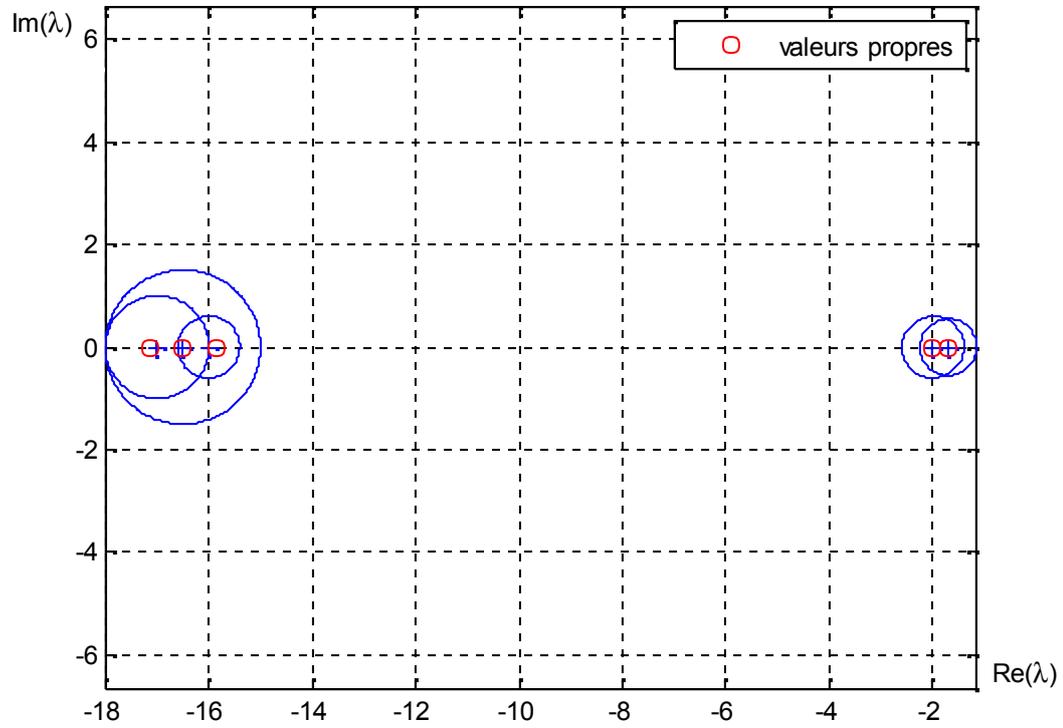


Figure 2.6: Cercles de Gerschgorin du modèle (2.30)

- Application de la technique des perturbations singulières

En appliquant les équations (2.4) à (2.9), les deux sous systèmes sont :

- Modèle réduit lent :

$$\frac{d}{dt} x_s = \begin{bmatrix} -1.7033 & 0.0022 \\ -0.0625 & -1.9991 \end{bmatrix} x_s + \begin{bmatrix} -0.9976 \\ -0.0018 \end{bmatrix} u_s,$$

$$y_s = [0.9709 \ 1.0002] x_s \tag{2.31}$$

$$z_s = \begin{bmatrix} -0.012 & -0.0063 \\ -0.0604 & -0.0242 \\ -0.0291 & 0.0002 \end{bmatrix} x_s + \begin{bmatrix} 0. \\ 0.0606 \\ 0 \end{bmatrix} u_s$$

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

- Modèle réduit rapide :

$$\frac{d}{dt} z_f = \begin{bmatrix} -16 & 0 & -0.3 \\ 0 & -16.5 & -0.1 \\ -0.5 & 0 & -17 \end{bmatrix} z_f(\ddagger) + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u_s(\ddagger)$$

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

(2.32)

$$z_{f0} = z_0 + \begin{bmatrix} 0.012 & 0.0063 \\ 0.0604 & 0.0242 \\ 0.0291 & -0.0002 \end{bmatrix} x_0$$

2.4. Conclusion

Dans ce chapitre, nous avons rappelé quelques notions sur la décomposition multi échelles de temps basée sur la technique des perturbations singulières. Son application suppose une mise sous forme standard singulièrement perturbée.

En outre, nous avons présenté une approche géométrique basée sur les cercles de Gerschgorin pour l'identification et la séparation des dynamiques. Cette méthode est améliorée par l'utilisation de transformations modales permettant la variation de la taille des cercles et le déplacement de leurs centres.

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 3

MODULARITE ET ARCHITECTURE NEUROGLIALE

3.1. Introduction

Dans le domaine de la recherche en Intelligence artificielle, différents modèles de réseaux de neurones artificiels qui émulent le système cérébral biologique sont proposés pour résoudre des tâches complexes. Ces modèles jouent un très grand rôle en traitement des données, grâce à leurs excellentes performances dans de nombreuses applications, leur robustesse et leur facilité de mise en œuvre. Néanmoins, l'analyse des modèles informatiques développés jusqu'à présent montre que les réseaux de neurones artificiels possèdent certaines limites en tant que paradigmes de traitement de l'information [2]. Les causes de ces limitations sont dues au fait que les modèles existants n'expriment pas certains comportements des neurones dans le cerveau humain et ne font pas intervenir la participation d'autres éléments dans les traitements de l'information. Ainsi, par exemple, ils ne considèrent pas que l'information soit traitée en deux échelles de temps différentes lentes et rapides.

Depuis que les RNA_s ont été utilisés pour émuler le cerveau, les chercheurs n'ont cessé de porter leur attention sur l'importance des neurones dans le système nerveux.

On remarque, ces derniers temps, que le système glial suscite par son rôle un grand intérêt dans le processus de traitement de l'information dans le système nerveux. En effet, de nouvelles découvertes révèlent que les astrocytes sont intimement liés au contrôle actif de l'activité neuronale et participent à la régulation de la neurotransmission synaptique [40]. Ils propagent de forte concentration de calcium pendant une longue durée sur une courte distance à basse vitesse, contrairement aux neurones qui étendent leurs axones sur de grandes distances et réalisent des potentiels d'action de courte durée à grande vitesse [41]. De plus, les astrocytes communiquent entre eux dans un réseau séparé, parallèle au réseau neuronal, à vitesse différente de celle de ce dernier [3-5].

L'étude du rôle du système glial devient donc un nouvel axe de recherche car certains chercheurs considèrent que les caractéristiques de ses cellules gliales peuvent être appliquées à des réseaux de neurones artificiels. Dans ce cas, il pourra être utile d'intégrer dans les réseaux de neurones artificiels les nouvelles connaissances sur le traitement de l'information dans le cerveau humain, soit en concevant des neurones plus complexes, soit en élaborant des architectures plus complexes.

3.2 Comment introduire les connaissances des cellules gliales dans les réseaux de neurones artificiels

Un modèle de neurone à cellule gliale plus sophistiqué est un modèle de neurone qui tient compte de plus de paramètres. Peu de travaux de recherche explore l'intégration des cellules gliales dans les RNA_s [42-49].

Parmi ces recherches est celle de C. Ikuta et al qui ont proposé des réseaux gliaux pour améliorer les performances du MLP. Ils ont proposé un MLP avec un réseau glial dont les cellules gliales génèrent des oscillations chaotiques [42].

Dans un second travail [43], ils ont étudié le MLP avec le réseau glial à impulsions. La cellule gliale génère uniquement une sortie impulsionnelle.

La même équipe a proposé dans [44] un réseau MLP avec une chaîne gliale à impulsions à partir des caractéristiques de la glie biologique. Dans ce modèle, les cellules gliales sont connectées aux neurones, un par un, dans la couche cachée du MLP.

Dans [45], ils ont publié une nouvelle étude dans laquelle ils ont appliqué le réseau MLP avec une chaîne gliale à impulsions au problème à deux spirales (TSP).

Enfin, dans [46], C. Ikuta et al ont proposé un réseau MLP avec une chaîne gliale à impulsions positives et négatives qui s'inspire des caractéristiques biologique des cellules gliales. Dans ce modèle La cellule gliale génère des impulsions lorsqu'elle est excitée par la sortie du neurone connecté. Si ce dernier

a une grande quantité de sortie, la glie génère une impulsion positive. De plus, si le neurone connecté a une petite quantité de sortie, la glie génère l'impulsion négative. Les impulsions positives et négatives se propagent au neurone connecté et à la glie voisine.

Dans [47], Alvarellos et al ont étudié les conséquences de l'inclusion d'astrocytes artificiels, qui présentent les propriétés biologiquement impliquées dans la communication entre les neurones et les astrocytes, sur la performance des réseaux de neurones artificiels. À l'aide de systèmes connexionnistes et d'algorithmes évolutifs, ils ont comparé les performances de réseaux de neurones artificiels et de réseaux neuroglial artificiels pour résoudre des problèmes de classification.

Dans [48], ils ont étudié, à l'aide de modèles informatiques, différentes interactions astrocyte-neurones pour le traitement de l'information; différents algorithmes neuro-glial ont été mis en œuvre pour la formation et la validation de réseaux multicouches artificiels neuro-glial orientés vers la résolution de problèmes de classification.

En 2014, Lansolsi et Kirman ont proposé un modèle de réseau neuro-glial en utilisant l'algorithme de cohonen avec l'algorithme de propagation [49].

Dans tous les modèles proposés par C. Ikuta et al, le réseau glial est connecté aux neurones de la dernière couche cachée uniquement. De l'autre côté, dans le modèle biologique, les astrocytes, type de cellule gliale, existent dans toutes les fentes synaptiques. Ainsi, le modèle mathématique proposé ne correspond pas au modèle biologique [49]. De plus, la courbe d'erreur présente des oscillations (augmentation et diminution de l'erreur) qui montrent que l'algorithme proposé n'est pas stable [49].

Il est à noter que lors de leur deuxième ouvrage publié, Alvarellos et al essayent, dans la phase d'apprentissage, de faire varier les valeurs attribuées aux paramètres utilisés, mais la question qui se pose ici est le choix de ces paramètres [49].

En plus de cela, Alvarellos et al utilisent l'algorithme génétique qui n'a aucune relation connue avec le paradigme des réseaux de neurones.

Une autre limite de l'algorithme proposé est que l'inspiration biologique ne correspond pas à l'algorithme mathématique. En fait, chaque astrocyte est en interaction avec un neurone, mais il n'y a pas d'interaction entre les astrocytes[49].

On remarque dans la réf [49] que l'algorithme proposé nécessite plusieurs couches caché et plus de neurones dans chaque couche cachée ce qui augmente considérablement la complexité de l'architecture qui ont résulte une convergence lente. En plus, l'inspiration biologique ne correspond pas à l'algorithme proposé car il ne tiens pas en compte que les astrocytes sont lents et les neurones sont rapides.

D'après ces recherches, on peut constater que L'élaboration de 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 en combinant plusieurs algorithmes. Cette complexité conduit à plusieurs inconvénients tels que; l'augmentation du cout algorithmique, une convergence plus lente et délicate de l'apprentissage ou encore un temps de réponse plus important.

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.

3.3. Modularité

La modularité peut être définie comme une subdivision d'un objet complexe en une multitude d'objets plus simples. Cette subdivision est déterminée soit par la structure, soit par la fonction de l'objet et de ses sous-parties.

La modularité peut être présente partout; dans les créatures vivantes ainsi que dans les objets inanimés, mais la subdivision en objets moins complexes n'est pas souvent évidente.

La réplication et la décomposition sont les deux principales formes de modularité ; ils se retrouvent dans des objets concrets ainsi que dans la pensée. Il est souvent difficile de distinguer nettement entre elles; la réplication et la décomposition se produisent souvent en combinaison[50].

La réplication est un moyen de réutiliser les connaissances. Une fois qu'un module développé s'est avéré utile, il est reproduit en plus grand nombre. Ce principe est souvent utilisé au niveau des organismes vivants, comme par exemple à différentes échelles en observant un humain: Deux jambes semblables, des doigts, des vertèbres de structure similaire, des milliers de modules de cheveux et des milliards de cellules [50].

En électronique, le développement des circuits intégrés repose sur la réplication d'unités simples pour construire une structure complexe.

La décomposition est souvent constatée lorsqu'il s'agit d'une tâche complexe. C'est un signe de comportement intelligent que de résoudre un problème complexe en le décomposant en tâches plus simples, plus faciles à gérer, puis à rassembler la solution à partir des résultats des sous-tâches. Construire un logiciel volumineux, construire une voiture ou résoudre une équation se fait généralement en décomposant le problème [50] .

3.4. Modularité dans le système nerveux

Le cerveau humain est une source d'inspiration pour développer des techniques de calcul plus sophistiquées qui conduisent à de meilleures performances. Une tâche cognitive peut impliquer différents processus ou mettre en œuvre dans la plupart des cas une combinaison de traitement en série et en parallèle.

Il apparait que si les humains ont la capacité de réaliser différentes tâches simultanément; certaines tâches parallèles sont plus simples à faire que d'autres. La plupart des gens n'ont aucun problème à marcher et à parler en parallèle; alors qu'ils trouvent très difficile d'écouter deux orateurs différents en même temps. Cela implique que les tâches pouvant être traitées dans différents modules peuvent être facilement effectuées en parallèle, alors que les tâches nécessitant la même unité de traitement sont difficiles à gérer simultanément [50].

Une étude intéressante de la relation entre structure et fonction des réseaux de neurones modulaires est donnée dans « Design and Evolution of Modular Neural Network Architectures » [51].

L'article examine la structure des preuves pour une architecture modulaire dans le cerveau humain, donnée par différents psychologues, biologistes et neurologues. Plusieurs niveaux de la modularité dans le cerveau sont décrits. Multitâche, capacités humaines et handicaps sont expliqués avec la structure modulaire et la structure parallèle du cerveau [51].

Dans le même article, Happle et Murre conclurent que la nature du traitement de l'information dans le cerveau est modulaire [51]. 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.

3.5. Modularité dans le réseau de neurones artificiel

Ces dernières années ont vue les réseaux neuronaux artificiels passer du stade de prototype de laboratoire à celui d'outil de traitement de données standard. Ces réseaux jouent un très grand rôle en traitement de données, ceci est du en grande partie à leurs excellentes performances, pour de nombreuses applications, leur robustesse et leur facilité de mise en Ouvre.

La maturité qui se dégage aujourd'hui de cet outil, ne signifie nullement que les réseaux de neurones artificiels soient figés, bien au contraire, ils continuent d'évoluer et une de ces évolutions est le besoin de modularité.

En effet, un grand nombre d'applications des réseaux neuronaux artificiels se caractérisent par le fait qu'ils soient relatifs à des bases de données gigantesques ; ce qui conduit inévitablement à une augmentation de la taille des réseaux mis en œuvre ; or dans ce domaine, tous les travaux de développement des réseaux neuronaux artificiels, insistent sur l'idée de se limiter à des réseaux de petites tailles et la pratique montre qu'il est très difficile de maîtriser la stabilité et la convergence des réseaux de grandes tailles.

L'utilisation du concept de modularité pour la définition de nouvelles architectures de réseaux artificiels de petites tailles, constitue une solution au problème et semble tout indiquée.

La modularité est un concept complexe, défini de diverses manières. Des classifications des approches modulaires sont proposées en littérature [52-56]. Nous retiendrons deux formes de modularité dans la conception d'architectures neuronales à plusieurs réseaux [57, 58]. La première est composée de plusieurs modules, chacun estimant une même grandeur. Toutes les sorties sont combinées pour fournir une réponse globale. Nous parlons alors de modularité à modèles multiples (ensemble-based approach). Le pendant de cette approche est la modularité vraie (neural nets ensemble). Ici, chaque réseau de l'architecture ne cherche pas à résoudre le problème dans sa globalité, mais plutôt à en résoudre un aspect. La tâche de départ est scindée en sous-tâches, plus simples à résoudre. La dépendance des modules de l'architecture peut être parallèle ou série.

Dans le cas de la modularité à modèles multiples, tous les modules sont en parallèle et prennent le même vecteur d'entrées. L'ensemble des réseaux de neurones utilisés apprend à résoudre la même tâche. L'apprentissage supervisé peut spécialiser chacun des modules dans un secteur de l'espace des réponses. La réponse finale de toute l'architecture est déterminée, soit par le vote (ou sélection exclusive winner takes all), soit par la combinaison des réponses de

chacun des modules. La figure (1.a) donne le schéma de principe de cette architecture.

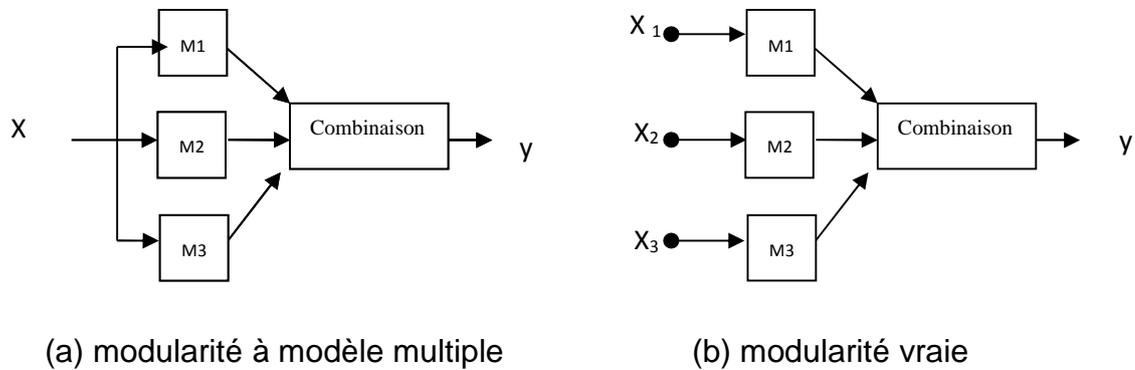


Figure 3.1 : Architectures de la modularité

L'avantage certain d'utiliser une telle architecture, est le peu de risque d'obtenir une sortie erronée, puisque celle-ci est estimée par plusieurs réseaux. Pour cela, il est possible de prendre des réseaux différents les uns par rapport aux autres. Ces différences peuvent intervenir au niveau de la taille des réseaux, des initialisations, des paramètres d'apprentissages, etc.

L'architecture mélange d'experts (ME) Fig. 3.2, proposé par Jacobs et al. (1991) [59] est une architecture proche de celle présentée Fig. 3.1.a. Elle est néanmoins, composée de deux types de réseaux : plusieurs réseaux experts et un réseau superviseur. Le superviseur détermine les coefficients de pondération de la participation de chacun des experts en fonction de l'entrée. Au cours de l'apprentissage les différents experts se spécialisent pour un groupe de réponses parmi l'ensemble des réponses possibles.

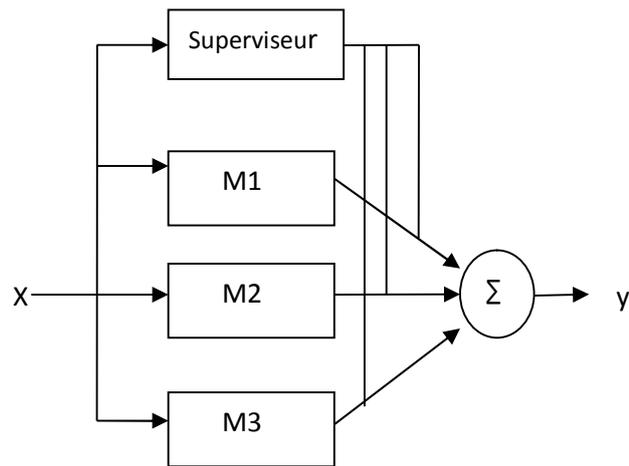


Figure 3.2 : Architecture mélange d'experts

La modularité vraie Fig. 3.1.b permet de travailler avec des réseaux de petite taille. Les objectifs d'une telle architecture sont aussi bien de permettre l'apprentissage de problèmes trop complexes, qu'améliorer les performances des réseaux (temps de traitement, coût algorithmique, etc.).

L'application du concept de modularité pour la définition de nouvelles architectures de réseaux artificiels de petites tailles implique les quatre étapes suivantes :

1. La décomposition de la tâche globale en un certain nombre de sous-tâches,
2. l'organisation de l'architecture modulaire,
3. choix du type de l'apprentissage,
4. définition de la technique de communication entre les modules.

3.5.1. Décomposition

La décomposition d'une tâche en sous-tâches est la première étape dans l'application de la modularité. Elle peut être effectuée sur l'espace d'entrée (décomposition horizontale) ou sur les variables d'entrée (décomposition verticale).

La décomposition horizontale effectue un regroupement spatial, un module pour le calcul de la valeur d'un paramètre pondéré ou d'une fonction plus complexe est généralement utilisé pour évaluer la performance de chaque groupe [60].

La décomposition verticale traite les variables d'entrée, qui sont regroupées en paquets. Ces paquets sont liés à des fonctions différentes qui sont estimées par différents modules. La combinaison des sorties de ces modules nous donnent des estimations de fonctions plus complexes [60].

La décomposition d'un système inconnu n'est pas toujours possible. Néanmoins, les connaissances physiques, à priori du système, permettent de surmonter cette étape.

- Décomposition horizontale :

La décomposition horizontale Fig.3.3 effectuer un regroupement spatial, un module pour le calcul de la valeur d'un paramètre pondéré ou d'une fonction plus complexe est généralement utilisé pour évaluer la performance de chaque groupe [60].

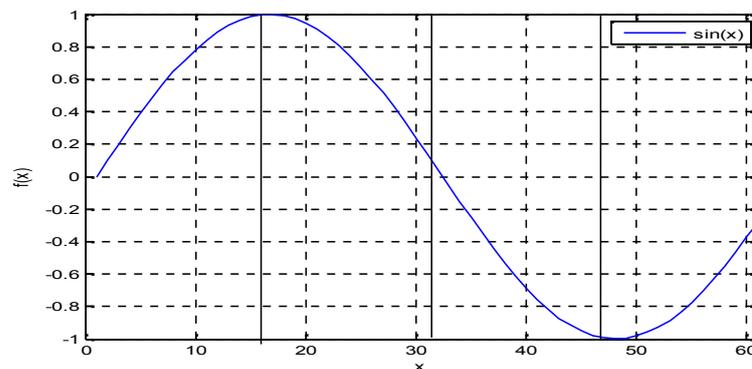


Figure 3.3 : Exemple de décomposition horizontale

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

- Décomposition verticale :

La décomposition verticale traite les variables d'entrée, qui sont regroupées en paquets. Ces paquets sont liés à des fonctions différentes qui sont estimées par différents modules. La combinaison des sorties de ces modules nous donnent des estimations de fonctions plus complexes [60]. L'exemple illustré en Fig.3.4 montre une fonction simple décomposée en deux sous fonctions d'une variable chacune.

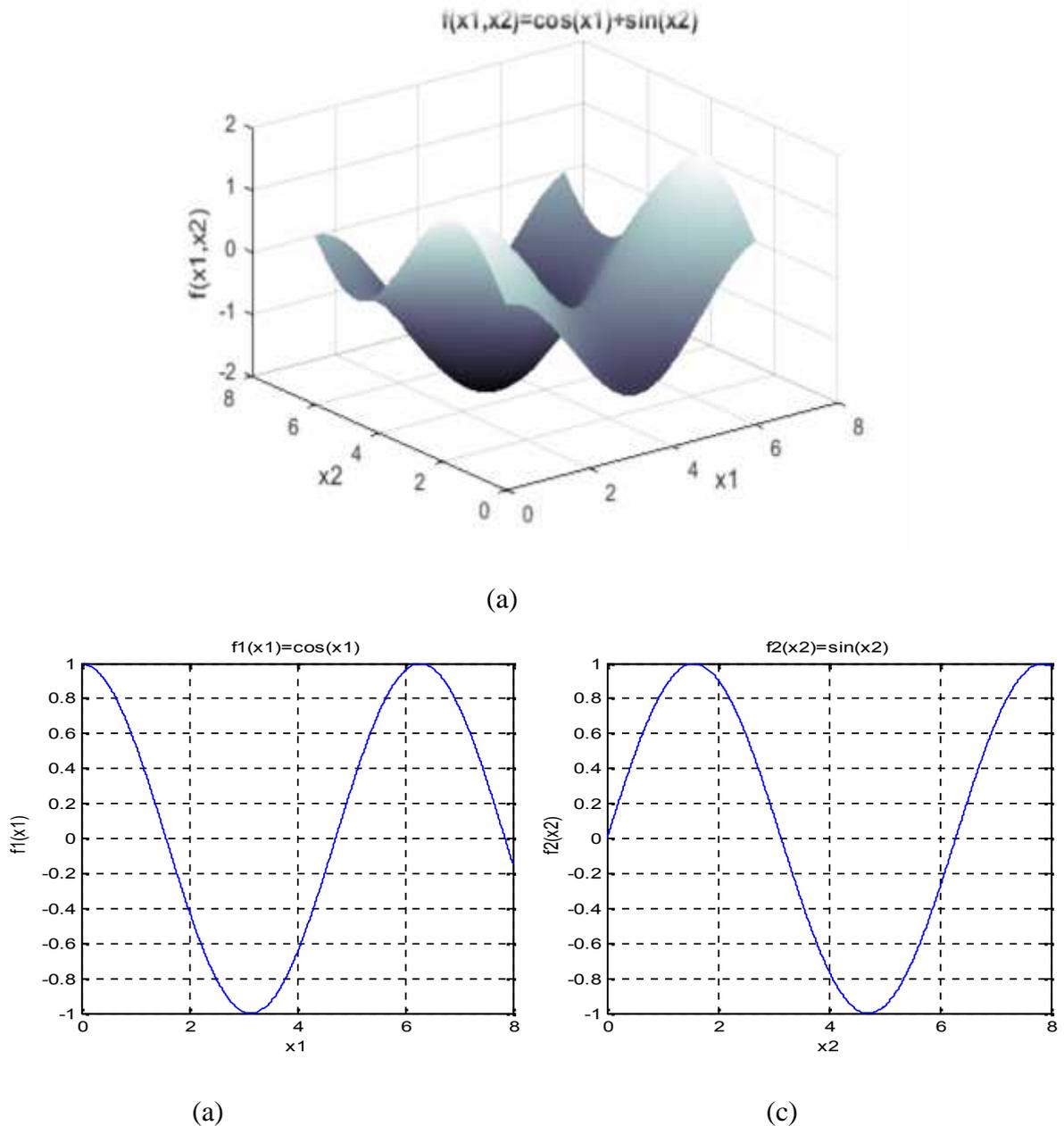


Figure 3.4 : Exemple de décomposition verticale

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.

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.

3.5.2. Organisation de l'architecture modulaire.

Nous distinguerons deux manières d'interconnecter les modules entre eux. Elles peuvent être parallèles ou 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.

Dans les architectures parallèles, tous les modules travaillent et traitent leurs informations simultanément. La sortie globale fait intervenir ou non chacun des modules, tout dépend des applications. La relation de coopération entre les modules peut être de type « et » ou de type « ou » [61]. Dans le premier cas, chacun des modules traite une partie du vecteur d'entrées et la sortie globale utilise les réponses de tous les modules, dans le second cas, c'est tout le vecteur d'entrées qui est utilisée, la réponse d'un seule module suffit pour calculer la réponse globale. La vocation première de ces combinaisons de réseaux est de diviser une tâche en sous-tâches que d'utiliser plusieurs réseaux pour améliorer les performances en termes de précision et de fiabilité.

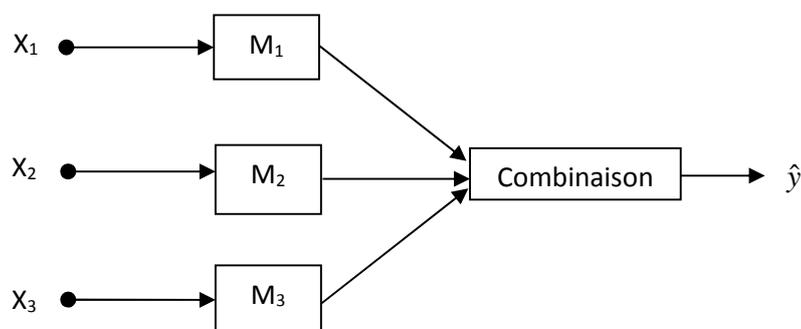


Figure 3.5 : Architecture parallèle

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 mélange d'experts (ME) Fig.3.2 qui supervise les modules et pondère leurs sorties.

Dans une architecture séquentielle, 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 ci-dessous. Les sorties du i-ème module, servent d'entrées au module i-ème+1.

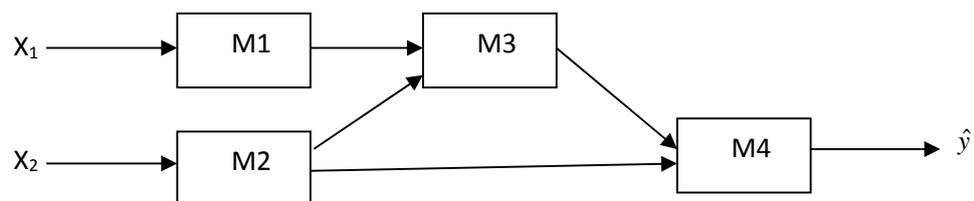


Figure 3.1 : Architecture séquentielle

3.5.3. Nature de l'apprentissage

Nous avons vu jusqu'à présent que les réseaux neuronaux ont la capacité d'apprendre, de mimer, des systèmes non linéaires complexes. Nous avons, par ailleurs, aussi vu que ces réseaux pouvaient souffrir d'une dimensionnalité trop grande du vecteur d'entrées. La décomposition semble être une solution dans la mesure où il est possible de définir des sous-tâches n'employant que quelques variables du vecteur d'entrées. Les modules sont alors organisés selon le schéma d'architecture séquentielle défini plus haut.

Bien que les règles d'adaptation propres à chacun des modules ne changent pas, l'organisation des réseaux de neurones dans une architecture modulaire va quand à elle, rendre l'apprentissage plus délicat. Les modules d'une architecture modulaire peuvent suivre des processus d'apprentissage différents.

- L'apprentissage indépendant

Entraîner les modules de manière indépendante semble être la manière la plus simple. Cela laisse supposer que les autres modules de l'architecture ne participent pas à l'apprentissage. L'interaction entre les modules n'intervient alors que lors de l'utilisation du système, dans la phase de restitution. Cette technique suppose que toutes les grandeurs sont accessibles, pour fournir des sorties désirées à chaque module dans le cas d'un apprentissage supervisé. Cette technique permet l'apprentissage de sous problèmes simples [62]. L'architecture est figée et déterminée au départ.

- L'apprentissage coopératif

Dans le cas de nombreuses applications, les seules grandeurs accessibles sont celles de l'espace d'entrées et celles de l'espace de sorties. L'idée est d'utiliser des méthodes globales qui permettent d'entraîner tous les modules en même temps. Il est nécessaire alors d'avoir une architecture figée, déterminée à l'avance. Un exemple est donné par le mélange d'experts [59], 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.

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.

3.5.4. Communication entre les modules

Les techniques pour le calcul de la sortie globale d'une architecture à plusieurs réseaux sont diverses et variées, on distingue entre autres, les techniques de vote, le calcul de la moyenne et de la moyenne pondérée et la combinaison non linéaire. Parmi les techniques de vote, on citera celle des votes pondérés [63]. Un poids est alors associé à chaque classificateur, il représente une mesure de performance. Pour la moyenne, la sortie est déterminée par la combinaison linéaire des sorties des réseaux. On peut citer aussi, une technique qui minimise l'erreur quadratique moyenne de la sortie globale [57].

L'utilisation des sorties combinées est une alternative plus efficace que celle qui consiste à choisir le réseau présentant les meilleures performances. On remarquera que, les sorties des réseaux, leurs temps de convergences et même leurs architectures sont différentes. L'élément déterminant l'efficacité d'un regroupement de réseaux de neurones est indépendant de ses membres, en termes d'erreurs [57]. Pour que les performances d'un ensemble de réseaux soient meilleures que celles d'un réseau unique, il faut faire en sorte que ces réseaux du groupement soient et demeurent indépendants.

3.6. Architecture du réseau neuroglial artificiel "ANGN"

L'architecture adoptée, pour notre réseau neuroglial artificiel, est basée partiellement sur les concepts de la modularité et demeure très proche de l'architecture « mélange d'experts » vue plus haut. Dans cette architecture, un certain nombre de réseaux de neurones (experts) sont supervisés par un réseau glial Fig.3.7 et Fig. 3.8. Le réseau glial superviseur, détermine les coefficients de pondération de la participation de chacun des experts (réseaux de neurones) en fonction de l'entrée.

Le réseau neuroglial artificiel utilise la stratégie « diviser pour mieux régner » dans laquelle les réponses des réseaux neuronaux experts sont combinées en une seule réponse rapide qui, agrégée à la réponse lente du réseau glial superviseur, nous donne la réponse globale du système.

Deux algorithmes différents, sont utilisés. Dans le premier algorithme, le réseau glial superviseur active d'une manière séquentielle les réseaux neuronaux experts, en fonction de l'entrée, sans test de performance Fig.3.7. Dans le deuxième algorithme, le réseau glial superviseur, évalue la performance de chaque réseau neuronal expert en fonction de l'entrée et sélectionne le meilleur qui sera activé Fig. 3.8.

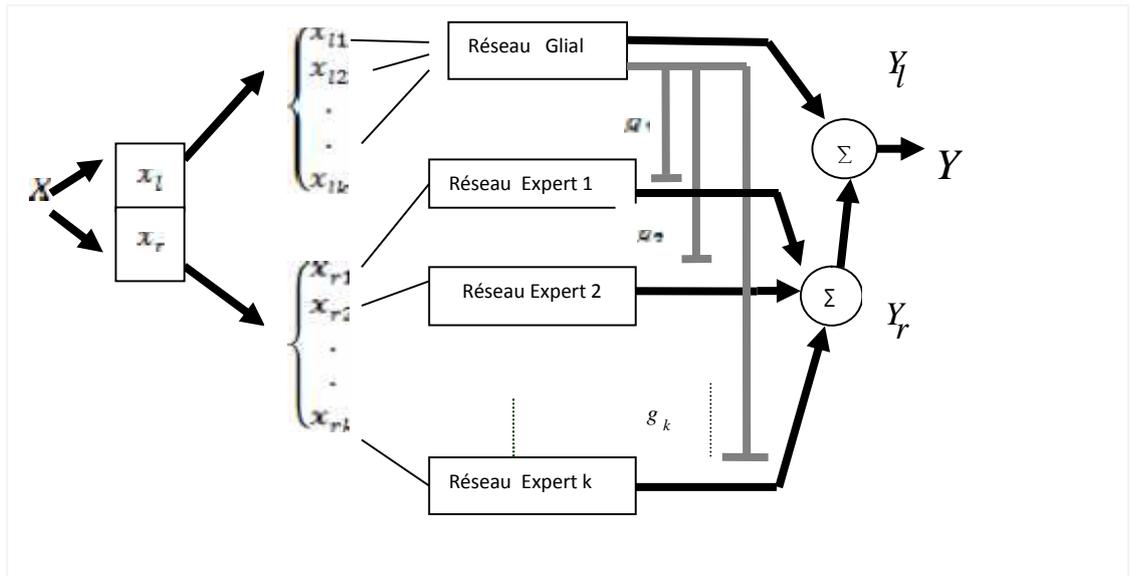


Figure 3.7 : Architecture du réseau neuroglial du 1^{er} algorithme

Comme illustré en Fig.3.7 et Fig. 3.8, notre réseau neuroglial global est composé de K modules neuronaux experts rapides et d'un module glial superviseur lent. Le vecteur des entrées est divisé en deux vecteurs X_l et X_r représentant respectivement, les entrées lentes et les entrées rapides du réseau, ils ont mêmes dimensions. Le vecteur X_l est affecté au réseau superviseur et le vecteur X_r , aux différents réseaux experts. Les réponses des modules experts sont combinées pour constituer la sortie rapide. Le module glial superviseur, possède deux sorties, la première, est utilisé pour le contrôle et la supervision des modules experts en sélectionnant le réseau le plus adéquat et en désactivant les autres, et cela pour chaque vecteur d'entrée. La deuxième sortie du module, représente la réponse lente du module glial, cette sortie est agrégée à la sortie rapide pour former la réponse globale du réseau neuroglial.

La structure du réseau neuroglial global, telle que présentée en Fig.3.7, est proche de celle de la modularité vraie ainsi la structure présentée dans la Fig.3.8 est proche de la modularité à modèles multiples. En effet, chaque réseau expert est spécialisé pour un sous problème précis, une décomposition verticale des variables d'entrée en entrées rapides et lentes, ainsi qu'une décomposition horizontale de l'espace d'entrée rapide X_r et lente X_l est utilisée.

Deux algorithmes différents sont adoptés dans ce travail, pour l'apprentissage de notre réseau neuroglial modulaire.

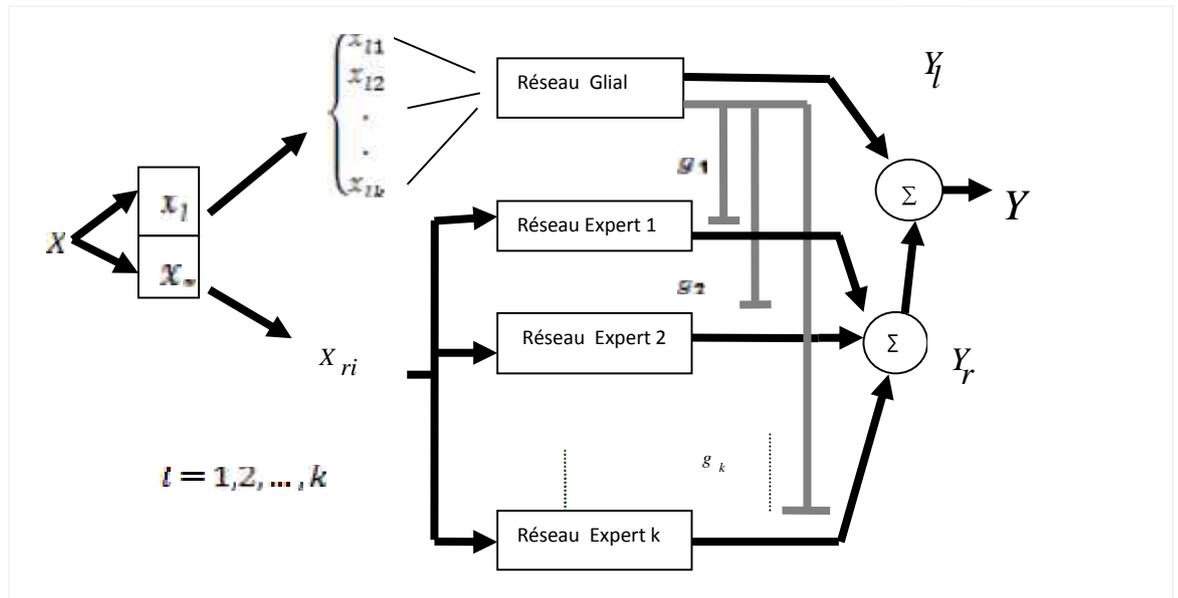


Figure 3.8 : Architecture du réseau neuroglial du 2^{ème} algorithme.

3.6.1. Premier algorithme

Dans le réseau neuroglial illustré en Fig.3.7, le vecteur des entrées rapides X_r est sectionné séquentiellement et équitablement en K vecteurs $X_{r1}, X_{r2}, \dots, X_{rk}$. Ces K vecteurs X_{ri} constituent les entrées respectives des K modules neuronaux experts.

Le vecteur des entrées lentes X_l est aussi sectionné de la même manière en K vecteurs $X_{l1}, X_{l2}, \dots, X_{lk}$ qui seront destinés d'une manière consécutive au module glial superviseur, lequel pour chaque X_{li} sélectionne et active le i ème module expert et désactive les autres et cela successivement pour i variant de 1 jusqu'à K . La sélection des modules experts par le module glial superviseur est systématique et n'utilise aucun critère de performance. On note que L'apprentissage du réseau glial ainsi que l'apprentissage des réseaux experts sont indépendants.

Cela se résume par la table 3.1, qui illustre les détails de ce premier algorithme d'apprentissage.

Table 3.1: Premier algorithme d'apprentissage du réseau neuroglial

Division de l'espace d'entrée rapide x_r en K vecteurs : $x_1(r), x_2(r), \dots, x_k(r)$ représentant les entrées respectives des experts.

Division de l'espace d'entrée lent x_l en K vecteurs : $x_1(l), x_2(l), \dots, x_k(l)$ représentant les entrées successives du module glial superviseur.

1. Initialisation des poids synaptiques des réseaux experts et du réseau glial
2. Pour chaque vecteur d'entrée lent $x_{i(l)}$ du module glial superviseur,

$i = 1, 2, 3, \dots, K$

2.1 pour $m = 1, 2, 3, \dots, q$

$$y_{i(l)}^m = X_{i(l)}^T w_{i(l)}^m(k)$$

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

$$e_{i(l)}^m(k) = d_{i(l)}^m(k) - y_{i(l)}^m(k)$$

$$w_{i(l)}^m(k+1) = w_{i(l)}^m(k) + y e_{i(l)}^m(k) X_{i(l)}$$

$$y_{i(r)}^m = X_{i(r)}^T w_{i(r)}^m(k)$$

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

$$e_{i(r)}^m(k) = d_{i(r)}^m(k) - y_{i(r)}^m(k)$$

$$w_{i(r)}^m(k+1) = w_{i(r)}^m(k) + y e_{i(r)}^m(k) X_{i(r)}$$

Fin pour

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

Si $X = X_{i(l)}$ alors $g_i = 1$

Sinon $g_i = 0$

$$\left. \begin{aligned} y_{i(r)} &= \sum_{i=1}^K g_i y_{i(r)} \\ y_i &= \sum_{i=1}^K (y_{i(l)} + y_{i(r)}) \end{aligned} \right\}$$

Agrégation des deux sorties lente et rapide

Fin pour

3.6.2 Deuxième algorithme.

Dans ce deuxième algorithme Fig.3.8, chaque vecteur des entrées rapides $x_{i(r)}$, ($i = 1, 2, \dots, K$) est destiné à tous les modules neuronaux experts. Ces modules apprennent des exemples différents de la base d'apprentissage et se spécialisent pour un groupe de réponses parmi l'ensemble des réponses possibles. Les réponses des modules neuronaux experts sont alors pondérées par le module glial superviseur, suivant leurs différences absolues par rapport à la réponse désirée. Le module expert dont la réponse est la plus proche de la réponse désirée aura le poids le plus élevé. Le réseau glial superviseur, dont l'entrée est le vecteur des entrées lentes $x_{i(l)}$, évalue la performance de chaque réseau neuronal expert en fonction de l'entrée et sélectionne le meilleur qui sera activé.

Cet algorithme, présente beaucoup de similitudes avec celui développé par Jacob [59] dans l'architecture « mélange d'experts ». Ces similitudes concernent, la supervision et la sélection des experts, avec néanmoins trois principales différences :

- Dans l'architecture « mélange d'experts », le superviseur et les experts possèdent la même échelle de temps [59]. Pour notre réseau neuroglial, les deux types de modules travaillent sur deux échelles de temps, lente et rapide.
- La tâche du réseau superviseur dans l'approche mélange d'expert consiste à superviser et à contrôler la compétition des experts. Dans notre approche, le réseau glial supervise, contrôle la compétition des experts et contribue à la réponse globale du système, en fournissant la réponse lente.
- Pour la sélection, la pondération des experts est binaire (0 ou 1). Dans l'approche de mélange d'expert les poids sont des probabilités entre 0 et 1.

Avant de présenter notre algorithme, on notera que, L'apprentissage des experts est coopératif, c'est-à-dire que les experts apprennent simultanément et divisent la

tâche en cours d'apprentissage. Les poids des réseaux neuronaux experts et ceux du réseau gliales, pour le vecteur sélectionné, sont mises à jour au même moment, en utilisant la propagation.

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

1. Séparation du vecteur d'entrée X en deux vecteurs lent et rapide : x_l et x_r
2. Chaque vecteur $x_{i(r)}$, ($i = 1, 2, \dots, K$) est destiné à tous les modules neuronaux experts.
3. Chaque vecteur $x_{i(l)}$, ($i = 1, 2, \dots, K$) est destiné au module glial superviseur.
4. Apprentissage du réseau glial pour l'obtention de la réponse lente désirée correspondante à l'entrée.
5. Sélection de l'expert suivant la valeur de la probabilité $p(i/x_l)$ (sélection du i ème expert sachant l'entrée lente $x_{(l)}$).
6. La sortie de l'expert i représente la moyenne conditionnelle de la réponse désirée sachant l'entrée et le réseau expert.

3.6.2.1. L'apprentissage des experts (réseaux rapides)

Supposons que chaque réseau expert est représenté par la Fig. 3.9.

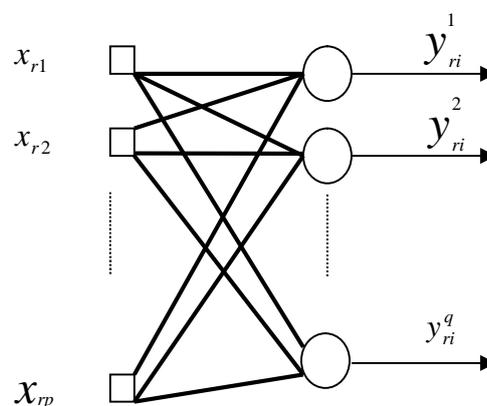


Figure 3.9 : Architecture du réseau expert

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 Fig.4.7, la sortie du neurone m est donnée par :

$$Y_{ri}^m = X_r^T W_{ri}^m \quad (3.1)$$

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 :

$$W_{ri}^m(k+1) = W_{ri}^m(k) + \Delta W_{ri}^m(k) \quad (3.2)$$

Ou

$$\Delta W_{ri}^m(k) = \gamma \frac{\partial L(W, g)}{\partial W_{ri}^m(k)} \quad (3.3)$$

γ : Le pas d'apprentissage.

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

$$\frac{\partial L}{\partial W_{ri}^m} = \frac{\partial L}{\partial y_{ri}^m} \frac{\partial y_{ri}^m}{\partial W_{ri}^m} \quad (3.4)$$

Après des simplifications, nous obtenons :

$$\frac{\partial L}{\partial y_{ri}^m} = h_i(d_r^m - y_{ri}^m) \quad (3.5)$$

D'après l'équation (3.1), nous avons :

$$\frac{\partial y_{ri}^m}{\partial W_{ri}^m} = X_r \quad (3.6)$$

En utilisant les équations (3.2) à (3.6) et en posant $e_{ri}^m = d_r^m - y_{ri}^m$, nous aurons :

$$W_{ri}^m(k+1) = W_{ri}^m(k) + y_{hi} e_{ri}^m(k) X_r \quad (3.7)$$

3.6.2.2. L'apprentissage du réseau glial

On considère le réseau glial représenté par la Fig. 3.10.

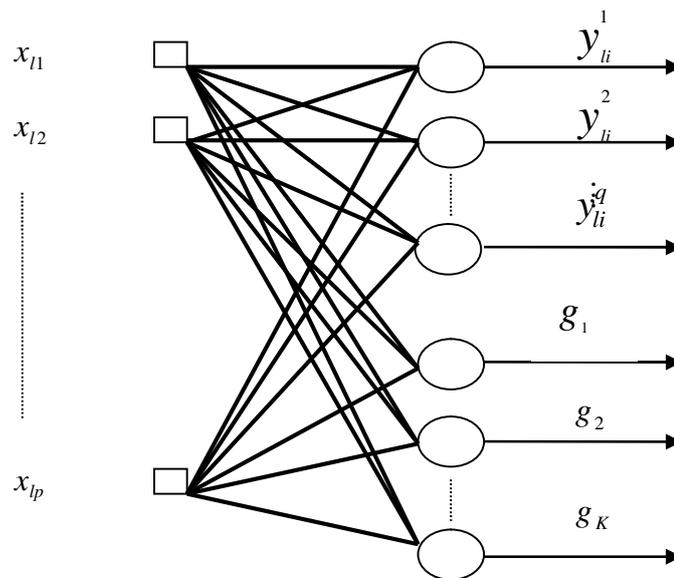


Figure 3.10 : Architecture du réseau glial

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 .

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

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

▪ 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 Fig. 3.8, on a :

$$u_i = X_l^T a_i \quad (3.8)$$

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 ».

$$g_i = \frac{\exp(u_i)}{\sum_j^K \exp(u_j)} \quad (3.9)$$

Le vecteur u_i est alors transformé par la fonction « softmax » afin de fournir des valeurs, 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 (3.9) doit satisfaire les conditions suivantes :

$$\begin{cases} 0 \leq g_i \leq 1 \\ \sum_{i=1}^K g_i = 1 \end{cases} \quad (3.10)$$

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

$$y_r = \sum_{i=1}^K g_i y_{ri} \quad (3.11)$$

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 :

$$L(W, g) = \ln \sum_{i=1}^k g_i \exp\left(-\frac{1}{2} \|d_r - y_{ri}\|^2\right) \quad (3.12)$$

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

$$h_i = \frac{g_i \exp\left(-\frac{1}{2}\|d_r - y_{ri}\|^2\right)}{\sum_{j=1}^k g_j \exp\left(-\frac{1}{2}\|d_r - y_{rj}\|^2\right)} \quad (3.13)$$

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 :

$$a_i(k+1) = a_i(k) + \Delta a_i(k) \quad (3.14)$$

$$\Delta a_i(k) = y \frac{\partial L(W, g)}{\partial a_i(k)} \quad (3.15)$$

ou

$$\frac{\partial L}{\partial a_i} = \frac{\partial L}{\partial u_i} \frac{\partial u_i}{\partial a_i} \quad (3.16)$$

Après des simplifications nous avons :

$$\frac{\partial L}{\partial a_i} = h_i - g_i \quad (3.17)$$

$$\frac{\partial u_i}{\partial a_i} = X_l \quad (3.18)$$

$$a_i(k+1) = a_i(k) + y(h_i(k) - g_i(k))X_l \quad (3.19)$$

- Apprentissage de la réponse lente

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 :

$$e(k) = d_i(k) - y_i(k) \quad (3.20)$$

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

$$W_i(k+1) = W_i(k) - \eta e(k) X_i(k) \quad (3.21)$$

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

A la présentation de $k^{\text{ième}}$ 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.

La table 3.2, illustre les détails de ce deuxième algorithme d'apprentissage.

Table 3.2 : Deuxième algorithme d'apprentissage du réseau neuroglial.

1. Initialisation des poids synaptiques des réseaux experts et du réseau glial.

2. Pour chaque vecteur d'entrée lent $x_{i(l)}$ du module glial superviseur :

2.1 Calculer pour $i = 1, 2, \dots, K$

Pour $m = 1, 2, \dots, q$

$$u_i(k) = X_{i(l)}^T a_i(k)$$

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

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

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

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

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

$$e_{i(l)}^m(k) = d_i^m(k) - y_{i(l)}^m(k)$$

$$W_{i(l)}^m(k+1) = W_{i(l)}^m(k) + y e_{i(l)}^m(k) X_{i(l)}$$

$$e_{i(r)}^m(k) = d_{i(r)}^m - y_{i(r)}^m(k)$$

$$W_{i(r)}^m(k+1) = W_{i(r)}^m(k) + y h_i(k) e_{i(r)}^m(k) X_{i(r)}$$

$$a_i(k+1) = a_i(k) + y (h_i(k) - g_i(k)) X_{i(l)}$$

Fin pour

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

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

sinon $g_i = 0$

$$y_{i(r)} = \sum_{i=1}^K g_i y_{i(r)}$$

2.4. $Y_i = \sum_{i=1}^K (y_{i(l)} + y_{i(r)})$ } agrégation des deux sorties lente et rapide

Fin pour

3.7. Conclusion

Deux thèmes ont été abordés dans ce chapitre. Le premier concerne l'intégration de nouvelles connaissances des cellules gliales dans les réseaux de neurone artificiels, le second concerne les différents aspects de la modularité ainsi le développement d'une nouvelle approche neuronale à réseau glial "ANGN" qui émule la réalité biologique des cellules gliales.

Il ne faut pas voir les réseaux de neurones comme des solutions miracles à tous les problèmes. Ils constituent en fait de bons outils d'estimation, d'approximation, capables de s'adapter à toutes sortes de situations. Face à des problèmes complexes, un seul réseau à une seule échelle de temps n'est souvent pas suffisant, il faut leur préférer une assemblée de réseaux à deux échelles de temps.

Le puissant concept de la modularité permet de fournir l'information des cellules gliales 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 différents, sont développés pour l'adaptation de cette structure :

- Dans le premier algorithme, le réseau glial superviseur active d'une manière séquentielle les réseaux neuronaux experts, en fonction de l'entrée, sans test de performance.
- En outre, dans le deuxième algorithme, le réseau glial superviseur, évalue la performance de chaque réseau neuronal expert en fonction de l'entrée et sélectionne le meilleur qui sera activé.

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 leurs caractéristiques, nous avons choisi le moteur à courant continu (MCC) et la machine asynchrone (MAS).

Chapitre 4

APPLICATION AUX SYSTEMES SINGULIEREMENT PERTURBES

4.1. Introduction

Les systèmes singulièrement perturbés doivent avoir une forme spéciale dite standard pour être modélisés et identifiés. Cette forme standard s'écrit comme suit :

$$\begin{cases} \dot{x} = f(x, z, u, t, \nu) & x \in \mathcal{R}^n \\ \nu \dot{z} = g(x, z, u, t, \nu) & z \in \mathcal{R}^m \end{cases} \quad (4.1)$$

Où $x \in \mathcal{R}^n$ et $z \in \mathcal{R}^m$ sont des variables d'état lentes et rapides, $u \in \mathcal{R}^{n+m}$ est le vecteur d'entrée et ν est un petit paramètre. Les fonctions f, g sont supposées suffisamment continues et dérivables par rapport à leurs arguments x, z, u, ν, t .

L'algorithme "ANGN" de modélisation et d'identification est proposé pour une classe de système non linéaire singulièrement perturbé avec différentes échelles de temps. La structure du schéma d'identification est illustrée à la Fig. 4.1.

Deux exemples sont également donnés pour démontrer l'efficacité du schéma d'identification. En outre, le 2^{ème} algorithme proposé de "ANGN" est appliqué pour identifier le moteur électrique à courant continu (MCC) et la machine asynchrone (MAS), y compris les aspects de phénomène rapide et lent.

Comparées à d'autres méthodes d'identification à multi-échelles de temps proposées à la réf [17], l'architecture "ANGN" proposée, a amélioré les performances d'identification.

En revanche, les deux algorithmes proposés sont appliqués pour modéliser la machine asynchrone. Les résultats obtenus sont comparés aux résultats du modèle à réseau de neurones unique, modulaire à deux réseaux de neurones et mélange d'experts (ME).

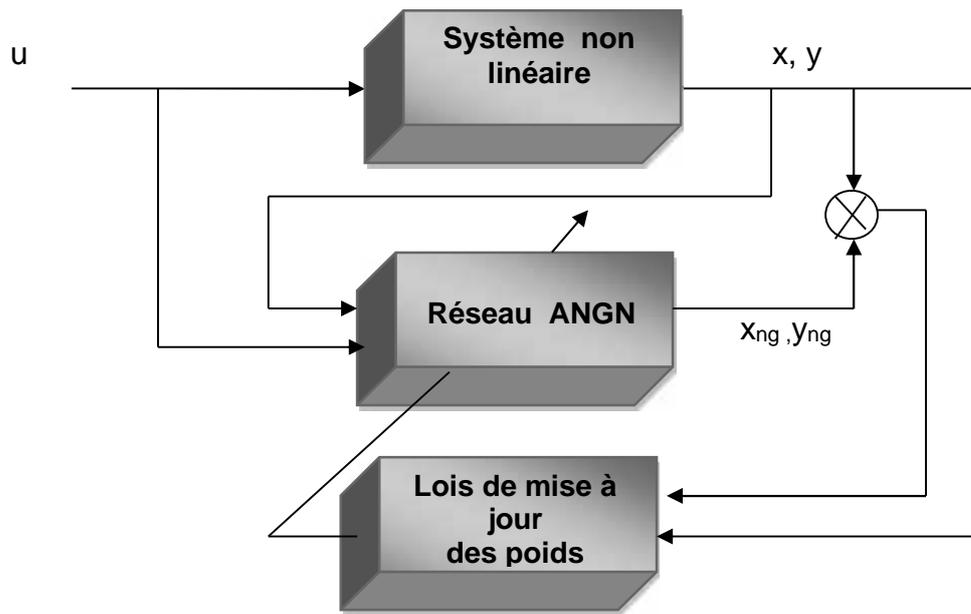


Figure 4.1 : Schéma d'identification "ANGN"

4.2. Identification du moteur à courant continu en utilisant le second algorithme proposé

Le moteur à courant continu (MCC) est un exemple typique de système à deux échelles de temps. La modélisation du MCC peut être séparée en deux sous-systèmes électrique et mécanique.

Il est bien connu que la constante de temps du système électrique est beaucoup plus petite que celle du système mécanique. Par conséquent, le sous-système électrique est le sous-système rapide et le système mécanique est le sous-système lent. Le modèle du moteur à courant continu est présenté comme suit:

$$J \frac{d}{dt} \omega = k_t i \quad 4.2$$

$$L \frac{d}{dt} i = -k_b \omega - R i - V \quad 4.3$$

où i, V, R et L sont respectivement le courant, la tension, la résistance et l'inductance de l'induit, J est le moment d'inertie de l'arbre, ω est la vitesse

angulaire de rotation, $k_t i$ et $k_D \omega$ sont respectivement le couple et la force contre électromotrice et ε est l'échelle de temps.

Définir $u = v/k_D$, $T_m = J/s(k_t k_D)$, $T_e = L/R$, les équations d'état (4.2) et (4.3) peuvent être amenées sous une forme standard d'un système singulièrement perturbé (4.4) et (4.5) :

$$\frac{a}{a} = a_1 i \quad 4.4$$

$$\varepsilon \frac{a}{a} = -b_1 \omega - b_2 i + u \quad 4.5$$

Où $\varepsilon = T_e/T_m$, $a_1 = b_1 = b_2 = 1$, La constante de temps électrique T_e étant très inférieure à la constante de temps mécanique T_m , le paramètre ε est très petit.

4.2.1. Séparation des modes électromagnétiques

L'application de l'algorithme des cercles de Gerschgorin à la matrice d'état A :

$$A = \begin{bmatrix} 0 & 1 \\ -1/\varepsilon & -1/\varepsilon \end{bmatrix} \quad 4.6$$

avec $\varepsilon = 0.5$

donne la figure 4.2 :

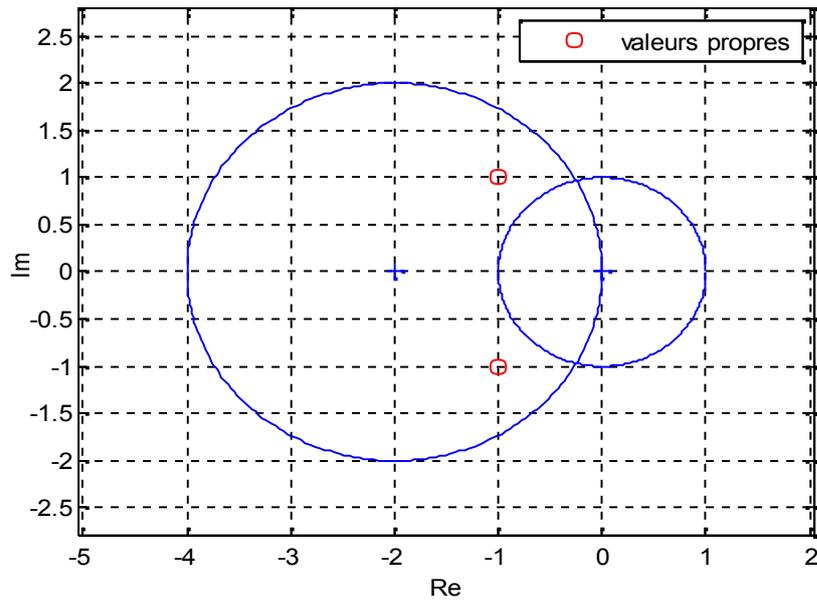


Figure 4.2 : Cercles de Gerschgorin pour le modèle 4.6 du MCC

1. Les cercles sont centrés en $C_1 = 0$ et $C_2 = -2$, et leurs rayons sont respectivement $R_1 = 1$ et $R_2 = 2$.

On remarque que Les cercles se coupent. On effectue un changement de la taille des rayons par la transformation 1.

$$P_1 = \begin{bmatrix} -1.541 & 0 \\ 0 & -0.237 \end{bmatrix}$$

Pour obtenir :

$$A_1 = \begin{bmatrix} 0 & 6.5021 \\ -0.3076 & -2 \end{bmatrix}$$

4.7

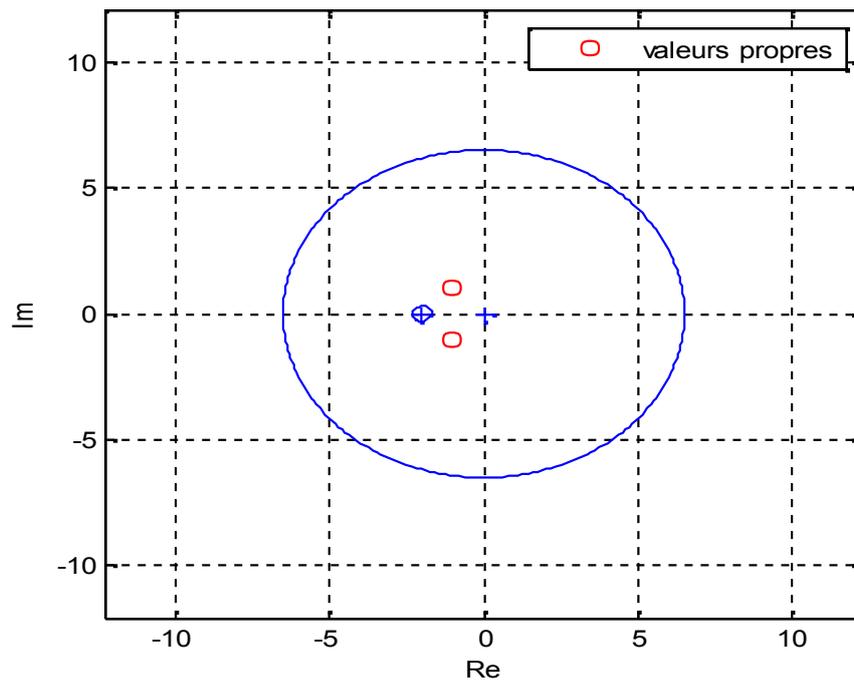


Figure 4.3 : Cercles de Gerschgorin pour le modèle 4.7 du MCC

2- Les nouveaux cercles Fig. 4.3 sont centrés en $C_1 = 0$ et $C_2 = -2$, et leurs rayons sont respectivement $R_1 = 6.5021$ et $R_2 = 0.3076$.

On remarque que les cercles ne sont pas disjoints.

Les centres des cercles peuvent être déplacés par la transformation :

$$x_2 = P_2 x_1, P_2 = \begin{bmatrix} 1 & 0 \\ -0.8 & 1 \end{bmatrix},$$

et :

$$A_2 = \begin{bmatrix} 5.2017 & 6.5021 \\ -6.0689 & -7.2017 \end{bmatrix}$$

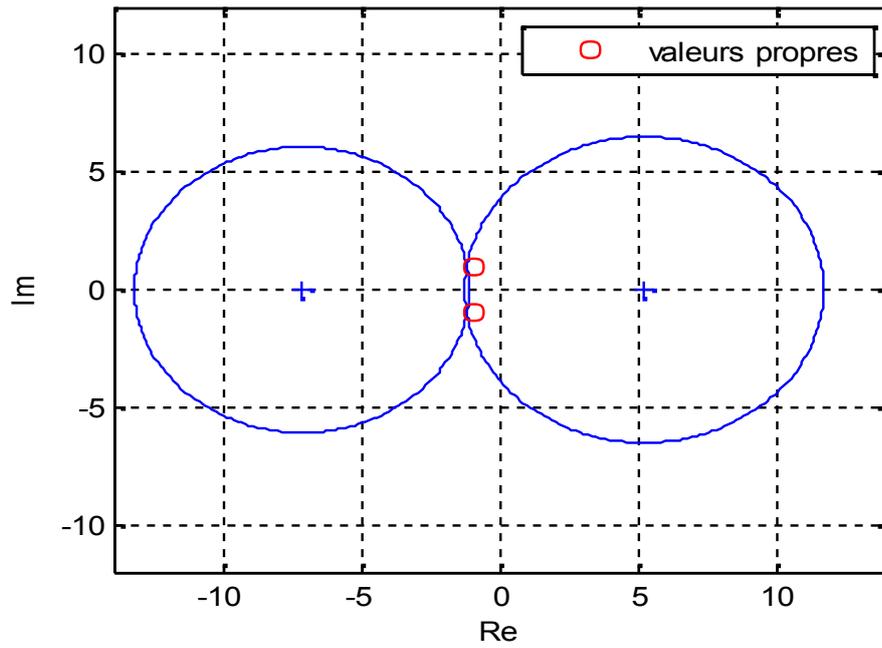


Figure 4.4 : Cercles de Gerschgorin pour le modèle 4.8 du MCC

3- Les nouveaux cercles-ligne sont disjoints et sont centrés en $C_1 = 5.2017$ et $C_2 = -7.2017$, Fig 4.4; leurs rayons sont respectivement : $R_1 = 6.5021$ et $R_2 = 6.0689$.

La transformation finale est :

$$x = p_2 p_1 \begin{bmatrix} \omega \\ i \end{bmatrix}, \quad P_2 P_1 = \begin{bmatrix} -1,541 & 0 \\ 12,328 & -0,237 \end{bmatrix}$$

4.2.2. Simulation

Afin d'illustrer l'efficacité du schéma d'identification proposé à la Fig. 4.1 le modèle du MCC mis sous forme standard singulièrement perturbé est utilisé.

Pour démontrer les performances du 2^{ème} algorithme d'identification de la métaheuristique "ANGN" proposé dans la section 3.6.2, les résultats de simulation obtenus sont comparés aux résultats de la méthode d'identification basée sur les réseaux de neurones dynamiques à multi couches proposée dans la réf [17]. Nous avons utilisé les mêmes valeurs du paramètre α .

Les résultats d'identification des grandeurs \hat{I}_T et les erreurs d'identification sont présentés aux Figs. 4.5 - 4.10.

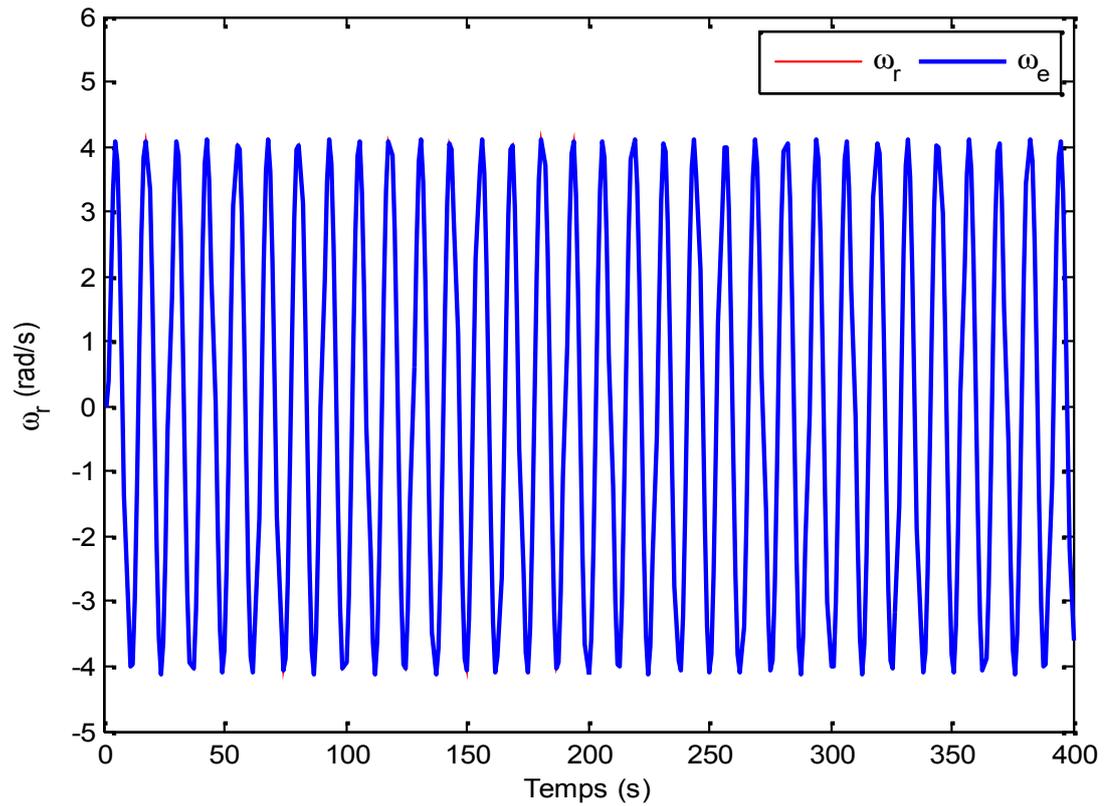


Figure 4.5 : Résultat d'identification de ω_r : 2^{ème} algorithme proposé de "ANGN"

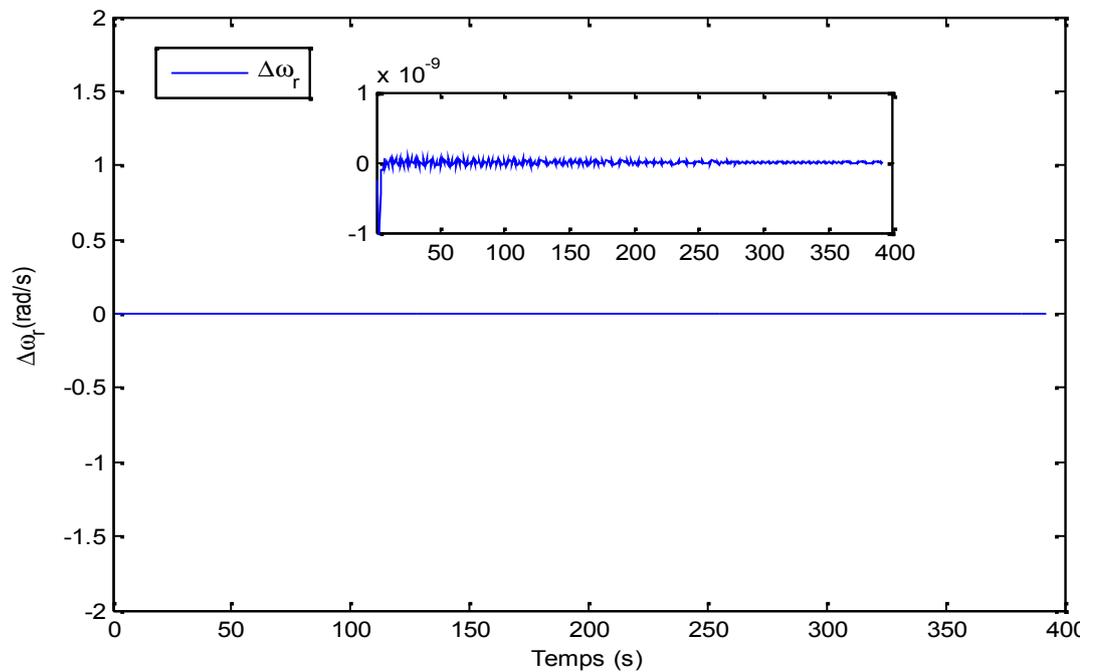


Figure 4.6 : Erreur d'identification de ω_r : 2^{ème} algorithme proposé de "ANGN"

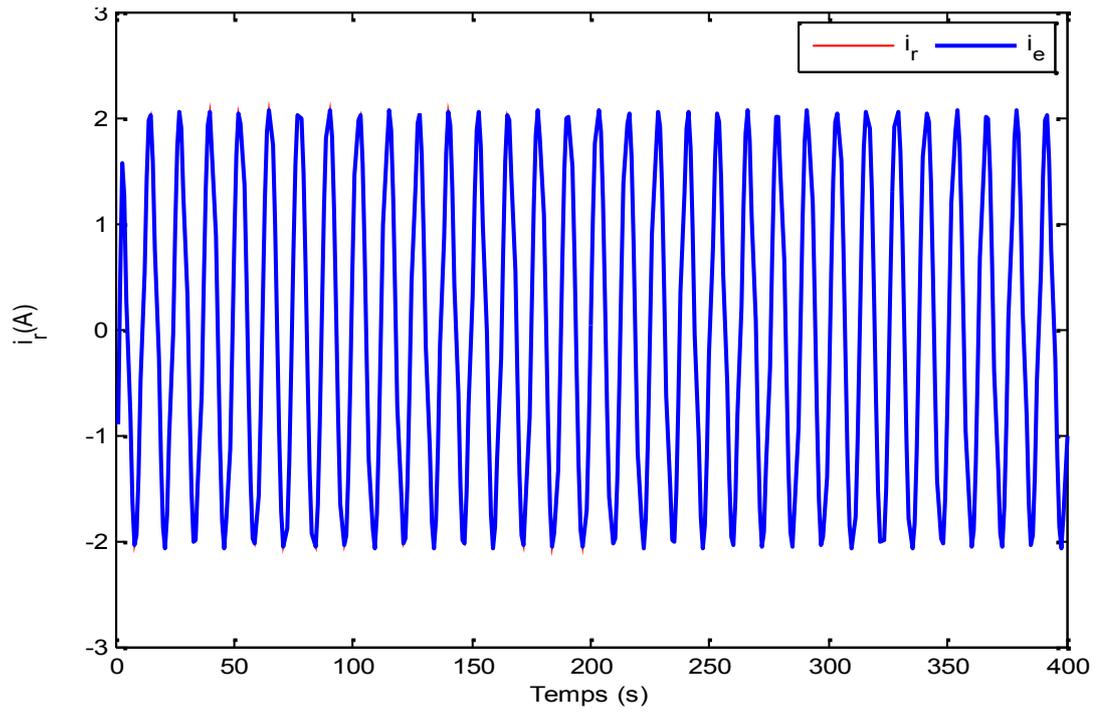


Figure 4.7 : Résultat d'identification de i_r : 2^{ème} algorithme proposé de "ANGN"

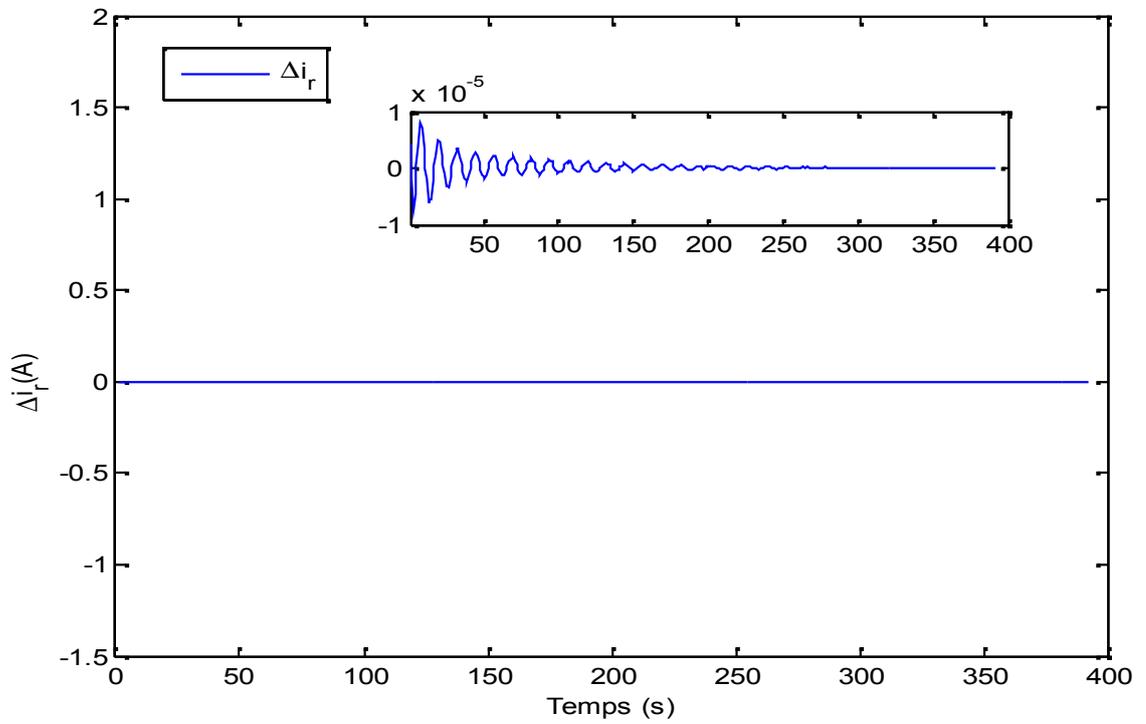


Figure 4.8 : Erreur d'identification de i_r : 2^{ème} algorithme proposé de "ANGN"

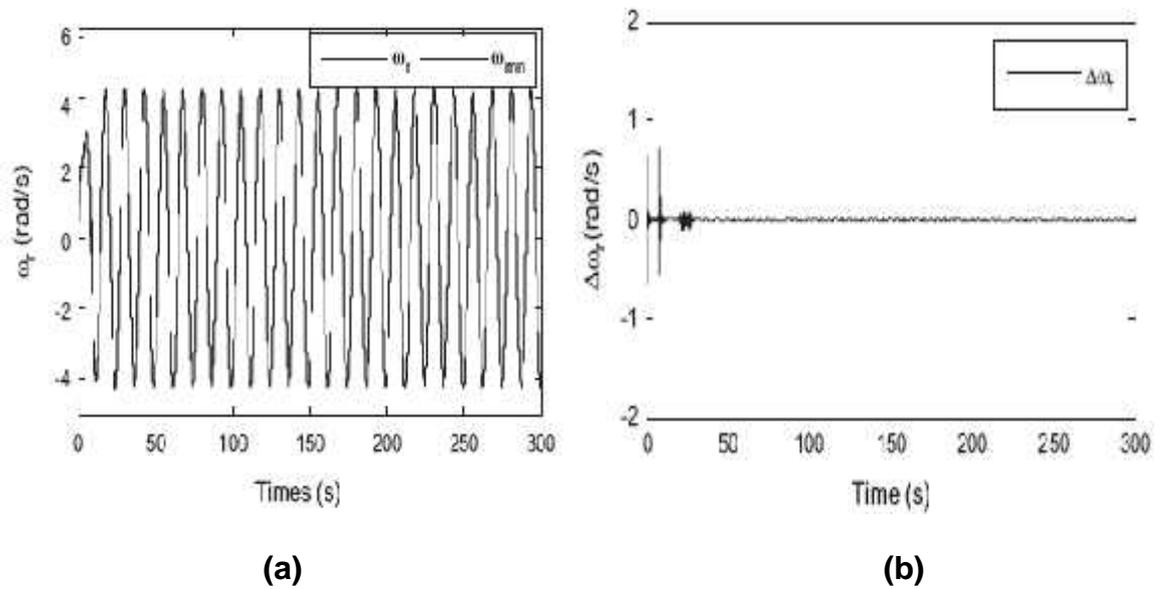


Figure 4.9 : (a) Résultat d'identification de ω_r réf.[17] (b) Erreur d'identification de ω_r réf. [17]

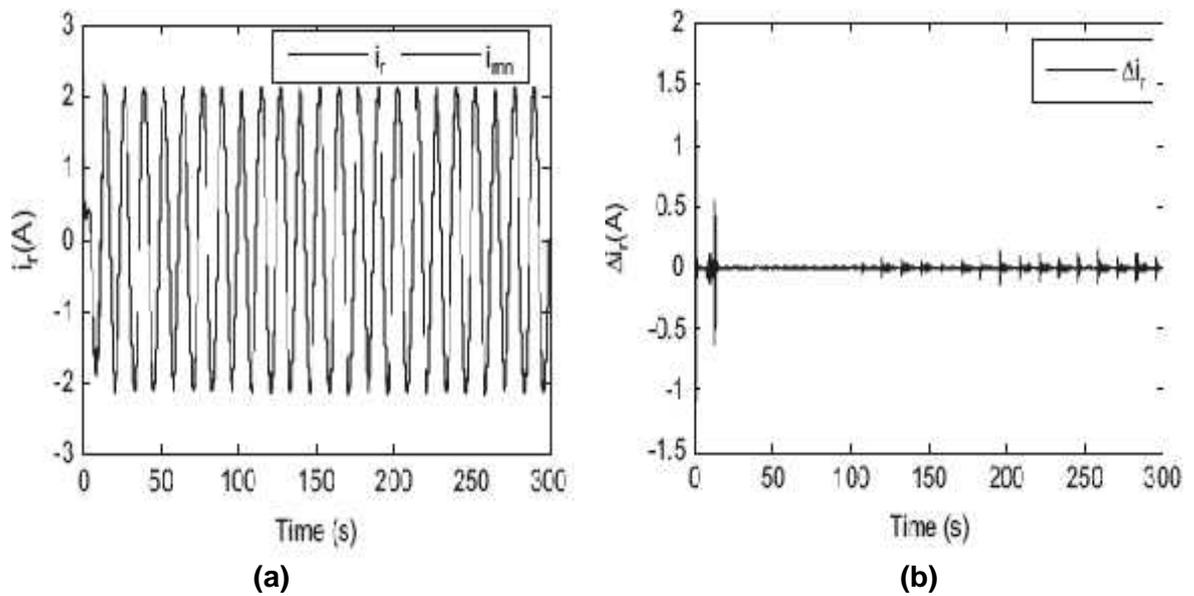


Figure 4.10 : (a) Résultat d'identification i_r réf.[17] (b) Erreur d'identification i_r réf.[17]

Les Figs 4.5 et 4.7 montrent clairement que les performances d'identification sont très bonnes, les états estimés ω_e et i_e suivent parfaitement et rapidement les états réels ω_r et i_r du système depuis le tout début du processus d'identification

lorsque le schéma d'identification à base du 2^{ème} algorithme proposé de l'architecture "ANGN" est utilisé. Dans les Figs 4.6 et 4.8, les tracés d'agrandissement des erreurs des premières phases du processus d'identification sont présentés dans les coins supérieurs droits. Il ressort clairement de ces graphiques d'agrandissement que les erreurs d'identification ϵ_r et Δi_r sont beaucoup plus petites que les erreurs d'identification obtenues dans la réf [17] comme présenté aux Fig. 4.9(a) et Fig. 4.10(a). En effet des erreurs plus importantes peuvent être observées pendant tout le processus d'identification dans la Fig. 4.10.

Par conséquent, il est clair que les résultats d'identification du second algorithme proposé de "ANGN" sont bien meilleurs que ceux dans la réf [17].

Pour comparer d'avantage les performances de différentes méthodes d'identification, l'indice de performance de la racine de l'erreur quadratique moyenne (RMS) des états est défini comme suit:

$$R = \sqrt{(\sum_{i=1}^n e^2(i))/n} \quad 4.9$$

où n est alors le nombre de pas de simulation, e est la différence entre les variables d'état dans le modèle et le système à l' i ème étape.

Les valeurs RMS de ϵ_r et i_r pour le MCC sont présentées dans le tableau 4.1.

Tableau 4.1 : Les valeurs RMSE pour les variables d'états ϵ_r et i_r

| R | $\omega_r(w)$ | $i_r(A)$ |
|----------|----------------------|--------------------|
| Réf [17] | 0.0062 | 0.0045 |
| ANGN | $3,29 \cdot 10^{-5}$ | $15 \cdot 10^{-6}$ |

D'après le tableau 4.1, les erreurs RMS de i_r et les erreurs RMS de i_r du second algorithme proposé à la section (3.6.2) sont beaucoup plus petites que celles obtenues dans la réf [17], ce qui signifie que l'architecture "ANGN" proposée donne des résultats plus précis.

4.3. Modélisation de la machine asynchrone par l'approche "ANGN"

La machine asynchrone constitue un système non-linéaire complexe, fortement couplée. Son modèle possède une dynamique à deux échelles de temps. Les performances de l'architecture "ANGN" proposée précédemment, sont évaluées sur les modèles réduits, lent et rapide de la machine asynchrone (MAS). Cette application se résume aux étapes suivantes :

1. Présentation de la machine sous sa forme singulièrement perturbée en présentant les modèles réduits lent et rapide.
2. La décomposition des variables d'entrées : Les variables d'entrées v_{sr}, v_{ss} aux variables lentes $v_{sr(l)}, v_{ss(l)}$ et aux variables rapides $v_{sr(r)}, v_{ss(r)}$. Cette décomposition est réalisée en utilisant les ondelettes.
3. L'apprentissage du réseau "ANGN".
4. La phase de reconnaissance des deux modèles "lent et rapide".
5. La combinaison des deux sorties $C_{em(l)}, C_{em(r)}$ du réseau glial (lent) et du réseau neuronal (rapide) pour l'obtention de la sortie globale du couple C_{em} .

Dans ce qui va suivre, nous présentons le modèle de la machine asynchrone, ainsi qu'une procédure itérative utilisant l'approche des cercles de Gerschgorin pour séparer les deux modes : lent et rapide, ainsi l'obtention d'une forme standard singulièrement perturbée. On applique ensuite la technique des perturbations singulières pour développer les modèles réduits lent et rapide. Notre architecture "ANGN" est testée suivant les deux algorithmes présentés dans les sections 3.6.1 et 3.6.2.

Dans le but d'avoir une erreur optimale, une meilleure vitesse de convergence et une configuration neuronale réduite, plusieurs études comparatives des performances vont être effectuées.

4.3.1. Séparation des modes électromagnétiques

Les équations d'état électriques en flux et le couple électromagnétique de la machine à induction en repère $\underline{\beta}$ s'écrivent :

$$\frac{d}{dt} \begin{bmatrix} W_{srs} \\ W_{rrs} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\dagger T_s} I_2 & \frac{B_r}{\dagger T_s} R(-\alpha) \\ \frac{1-\dagger}{\dagger T_r B_r} R(\alpha) & -\frac{1}{\dagger T_r} I_2 \end{bmatrix} \begin{bmatrix} W_{srs} \\ W_{rrs} \end{bmatrix} + \begin{bmatrix} v_{srs} \\ 0 \end{bmatrix} \quad (4.10)$$

$$C_{em} = P \frac{B_r}{\dagger L_s} \{ {}^T_{srs} R(-\alpha) J_2 W_{rrs} \}$$

avec :

$$\begin{aligned} R(\alpha) \cdot R(-\alpha) &= I_2 \\ R(\alpha) \cdot J_2 \cdot R(-\alpha) &= J_2 \\ R(\alpha) \cdot R(-\alpha) &= -\tilde{S} \cdot J_2 \end{aligned}$$

D'où :

$$A = \begin{bmatrix} -\frac{1}{T_{sp}} I_2 & \frac{B_r}{T_{sp}} R(-\alpha) \\ \frac{1-\dagger}{T_{rp} B_r} R(\alpha) & -\frac{1}{T_{rp}} I_2 \end{bmatrix} \quad (4.11)$$

$$\text{avec : } T_{sp} = \dagger T_s, \quad T_{rp} = \dagger T_r, \quad B_r = \frac{M}{L_r} \quad e \quad \sigma = (1 - M^2)/(L_s * L_r)$$

a) L'application de la technique des cercles de Gerschgorin à la matrice d'état A, donne :

1- Les cercles sont doubles et centrés en $C_{1,2} = -\frac{1}{T_{sp}}$ et $C_{3,4} = -\frac{1}{T_{rp}}$, et leurs rayons sont respectivement $R_{1,2} = \frac{B_r}{T_{sp}} (|\cos(\alpha)| + |\sin(\alpha)|)$ et $R_{3,4} = \frac{1-\dagger}{B_r T_{rp}} (|\cos(\alpha)| + |\sin(\alpha)|)$.

2- Pour toutes les valeurs de $R_{1,2}$ et $R_{3,4}$, les cercles se coupent.

b) On peut effectuer un changement de la taille des rayons par la transformation P_1 .

$$W_1 = P_1 \begin{bmatrix} W_{srs} \\ W_{rrs} \end{bmatrix}, P_1 = \begin{bmatrix} I_2 & 0 \\ 0 & B_r R(-\alpha) \end{bmatrix} \quad (4.12)$$

avec :

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

Pour obtenir :

Tapez une équation ici.

$$A_1 = \begin{bmatrix} -\frac{1}{T_{sp}} I_2 & \frac{1}{T_{sp}} I_2 \\ \frac{1-\dagger}{T_{rp}} I_2 & -\frac{1}{T_{rp}} I_2 - \check{S} J_2 \end{bmatrix}. \quad (4.13)$$

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 $R_{3,4} = \frac{1-\dagger}{T_{rp}} + \check{S}$.

2- Les cercles se coupent même pour les petites vitesses.

.

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

$$W_2 = P_2 W_1, P_2 = \begin{bmatrix} I_2 & 0 \\ -I_2 & I_2 \end{bmatrix},$$

qui donne :

$$A_2 = \begin{bmatrix} 0 & 0 & \frac{1}{T_{sp}} & 0 \\ 0 & 0 & 0 & \frac{1}{T_{sp}} \\ -\frac{r\ddagger}{T_{sp}} & -\check{S} & -\frac{r+1}{T_{sp}} & \check{S} \\ \check{S} & -\frac{r\ddagger}{T_{sp}} & -\check{S} & -\frac{r+1}{T_{sp}} \end{bmatrix}. \quad (4.14)$$

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

On représente sur la figure (4.11), les cercles de Gerschgorin qui sont encore doublés et centrés en $C_{1,2} = 0$ et $C_{3,4} = -\frac{1+r}{T_{sp}}$. Leurs rayons sont respectivement

$$R_{1,2} = \frac{1}{T_{sp}} \text{ et } R_{3,4} = \frac{r\ddagger}{T_{sp}} + 2\check{S}.$$

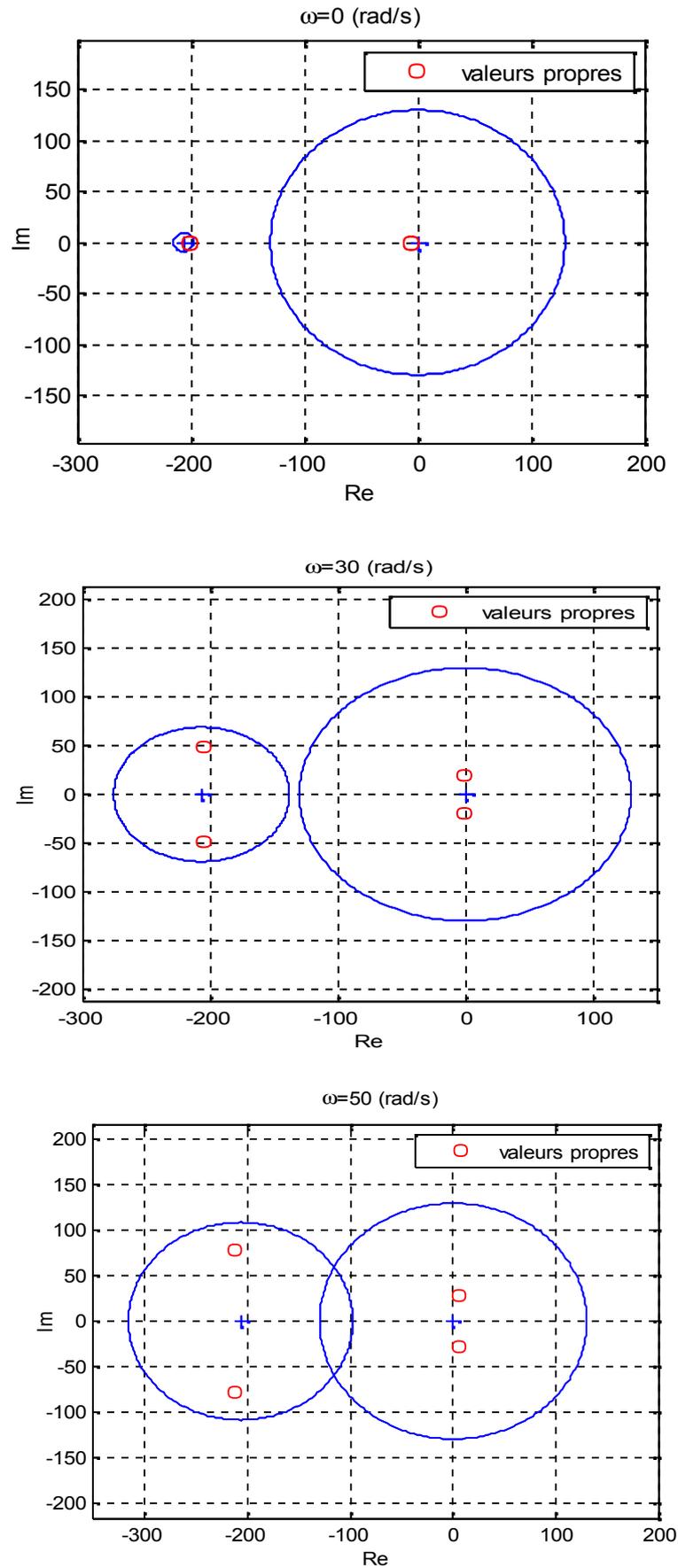


Figure 4.11 : Cercles de Gerschgorin pour différentes vitesses

On remarque que les nouveaux cercles-ligne sont disjoints pour les petites vitesses Fig. 4.11.; La transformation finale est :

$$W = P_2 P_1 \begin{bmatrix} W_{srS} \\ W_{rS} \end{bmatrix}, \quad P_2 P_1 = \begin{bmatrix} I_2 & 0 \\ -I_2 & B_r R(-\alpha) \end{bmatrix}, \quad (4.15)$$

4.3.2 Mise sous forme standard singulièrement perturbée

Dans ce cas, les composantes lentes et rapides sont identifiées. Nous pouvons alors appliquer la technique des perturbations singulières pour développer les sous-modèles lent et rapide.

En posant : $w = \begin{bmatrix} x \\ z \end{bmatrix}$ on obtient :

$$\begin{bmatrix} x \\ z \end{bmatrix} = P_2 P_1 \begin{bmatrix} W_{srS} \\ W_{rS} \end{bmatrix} \quad (4.16)$$

et la nouvelle représentation d'état s'écrit :

$$\frac{d}{dt} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{T_{sp}} I_2 \\ -\frac{1}{T_{rp}} I_2 & -\frac{r+1}{T_{sp}} I_2 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} I_2 & 0 \\ -I_2 & B_r R(-\alpha) \end{bmatrix} \begin{bmatrix} v_{srS} \\ 0 \end{bmatrix}, \quad (4.17)$$

Le couple électromagnétique s'exprime par l'expression

$$C_{em} = \frac{P}{L_{sp}} x^T J_2 z \quad (4.18)$$

La forme est standard le flux x est lent et le flux z est rapide. On remarque que le couple reste cependant non-linéaire.

▪ Sous-modèles réduits en repère β

En décomposant les flux, les tensions et le couple on obtient :

$$\begin{aligned} x &= x_l(t) \\ z &= z_l(t) + z_r(\ddagger) \\ v_{srs} &= v_{srs(l)}(t) + v_{srs(r)}(\ddagger) \end{aligned} \quad (4.19)$$

$$\text{et} \quad C_{em} = C_{em(l)} + C_{em(r)}$$

• **Modèle réduit lent :**

Ce modèle s'écrit :

$$\begin{cases} \frac{d}{dt} x_l = -\frac{1}{T_r(1+r)} x_l + \frac{r}{1+r} v_{srs(l)} \\ Z_l = -\frac{T_{sp}}{1+r} \left(\frac{1}{T_r} x_l + v_{srs(l)} \right) \\ C_{em} = \frac{P}{L_{sp}} x_l^T J_2 z_l \end{cases} \quad (4.20)$$

• **Modèle réduit rapide**

Ce modèle s'exprime par :

$$\begin{cases} \frac{d}{dt} z_r(\ddagger) = -\frac{1+r}{T_s} z_r - \ddagger v_{srs}(r) \\ C_{em}(r) = \frac{P}{L_{sp}} x_l^T J_2 z_r \end{cases} \quad (4.21)$$

▪ Modèles réduits en flux et en courants

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

1. D'après l'Eq (4.16), on écrit :

$$\begin{aligned} W_{srs} &= x \\ W_{r's} &= \frac{1}{B_r} R(r)(x + z) \end{aligned} \quad (4.22)$$

Les flux w_{srs} sont donc des variables lentes. Les flux w_{rrs} peuvent être décomposés en des flux lents w_{rrsl} et des flux rapides w_{rrsr}

Soit :

$$w_{srs} = w_{srs(l)} \quad (4.23)$$

$$w_{rrs} = w_{rrsl} + w_{rrsr} \quad (4.24)$$

avec (4.19) et (4.22), l'équation (4.24) devient :

$$w_{rrs} = \frac{1}{B_r} R(n) \left(\frac{1+r-\tau r}{1+r} x_l - \frac{T_{sp}}{1+r} v_{srs(l)} + z_r \right) \quad (4.25)$$

En identifiant les parties lentes et rapides, l'équation (4.24) permet d'écrire :

$$w_{rrsl} = \frac{1}{B_r} R(n) \left(\frac{1+r-\tau r}{1+r} x_l - \frac{T_{sp}}{1+r} v_{srs(l)} \right) \quad (4.26)$$

$$w_{rrsr} = \frac{1}{B_r} R(n) z_r \quad (4.27)$$

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

$$\begin{aligned} L_{sp} I_{srs} &= w_{srs} - B_r R(-n) w_{rrs} \\ L_{ip} I_{rrs} &= -\frac{1-\tau}{B_r} R(n) w_{srs} + w_{rrs} \end{aligned} \quad (4.28)$$

En décomposant les courants en composantes lentes et rapides :

$$\begin{aligned} I_{srs} &= I_{srs(l)} + I_{srs(r)} \\ I_{rrs} &= I_{rrsl} + I_{rrsr} \end{aligned} \quad (4.29)$$

On aboutit aux expressions suivantes :

$$\begin{aligned} L_{sp} I_{srs} &= w_{srs} - B_r R(-n) w_{rrsr} \\ L_{ip} I_{rrsr} &= -B_r R(-n) w_{rrsr} \end{aligned} \quad (4.30)$$

et :

$$\begin{aligned} L_{rp} I_{rrs} &= -\frac{1-\dagger}{B_r} R(\dagger) W_{srS} + W_{rrS(r)} \\ L_{rp} I_{rrS(r)} &= W_{rrS(r)} \end{aligned} \quad (4.31)$$

Les sous systèmes (4.20) et (4.21) deviennent :

$$\frac{d}{dt} \begin{bmatrix} W_{sr} \\ W_{sS} \end{bmatrix} = \begin{bmatrix} -\frac{1}{(1+r)T_r} & 0 \\ 0 & -\frac{1}{(1+r)T_r} \end{bmatrix} \cdot \begin{bmatrix} W_{sr} \\ W_{sS} \end{bmatrix} + \frac{r}{1+r} \cdot \begin{bmatrix} v_{sr}(t) \\ v_{sS}(t) \end{bmatrix} \quad (4.32)$$

$$C_{em(l)} = \frac{pT_s}{(1+r)L_s} (W_{sr} v_{sS}(t) - W_{sS} v_{sr}(t)), \quad (4.33)$$

et :

$$\frac{d}{dt} \begin{bmatrix} I_{sr(r)} \\ I_{sS(r)} \end{bmatrix} = \begin{bmatrix} -\frac{1+r}{T_s} & 0 \\ 0 & -\frac{1+r}{T_s} \end{bmatrix} \cdot \begin{bmatrix} I_{sr(r)} \\ I_{sS(r)} \end{bmatrix} - \frac{1}{L_s} \cdot \begin{bmatrix} v_{sr}(r) \\ v_{sS}(r) \end{bmatrix} \quad (4.35)$$

$$C_{em(r)} = p (W_{sr} I_{sS(r)} - W_{sS} I_{sr(r)}), \quad (4.36)$$

Les modèles rapide et lent sont illustrés respectivement dans les Figs 4.12(a) et 4.12(b). On notera que les variables rapides sont nulles à l'instant $t = 0,2$ s, puis le couple global est donné par la composante lente. Une deuxième remarque concerne l'influence des valeurs initiales. Ainsi, avec un choix judicieux des valeurs initiales, le couple global obtenu à partir des modèles réduits Fig. 4.14 suit parfaitement le couple global du modèle de la machine asynchrone Fig. 4.13.

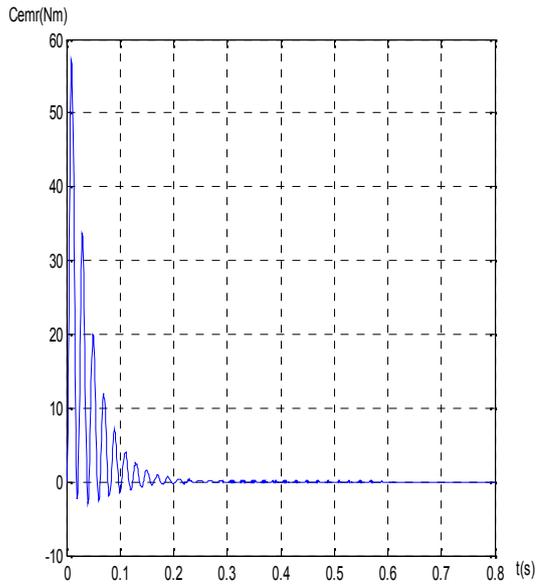
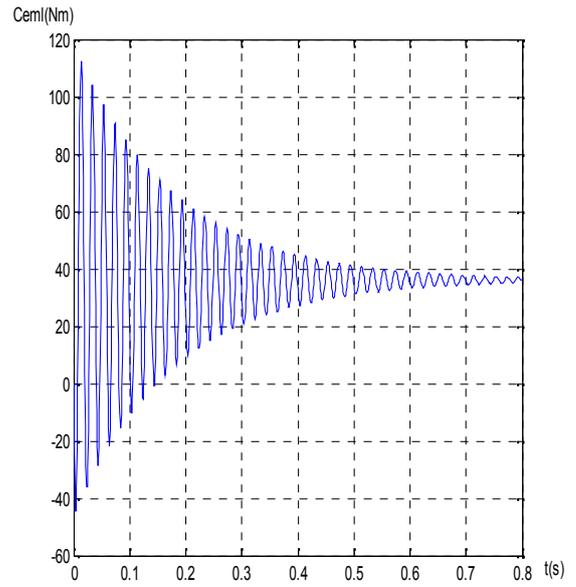


Figure 4.12 : (a) Le couple rapide



(b) Le couple lent

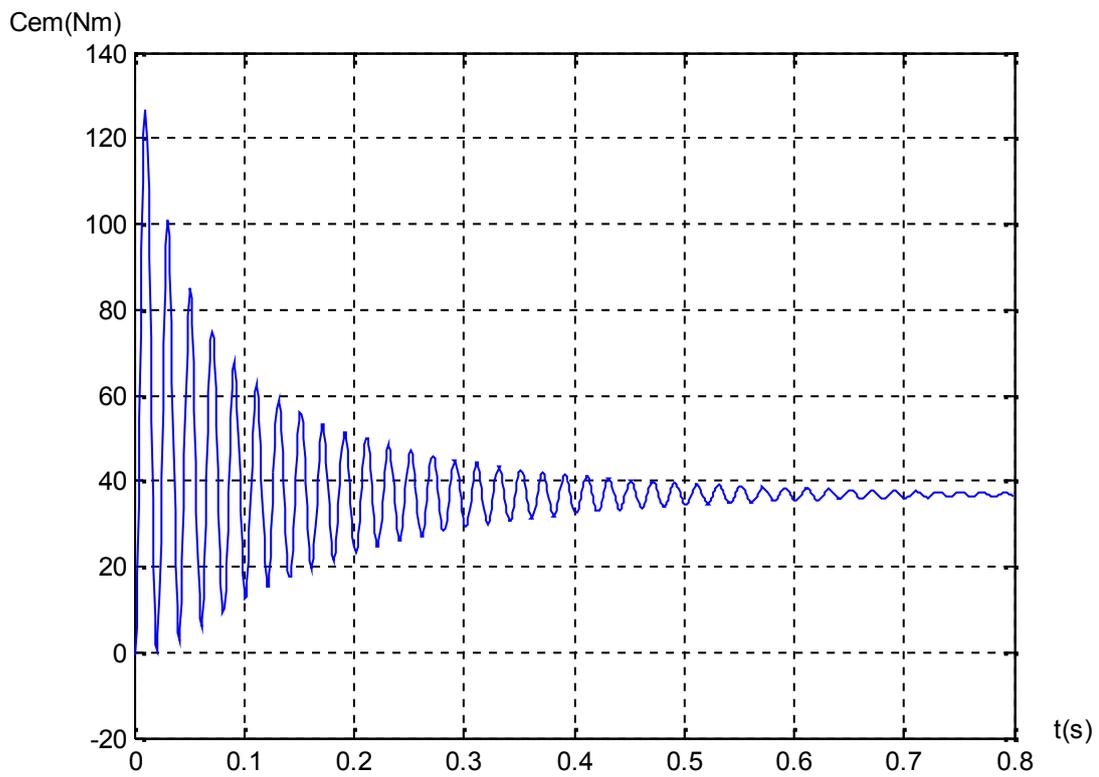


Figure 4.13 : Le couple global électromagnétique

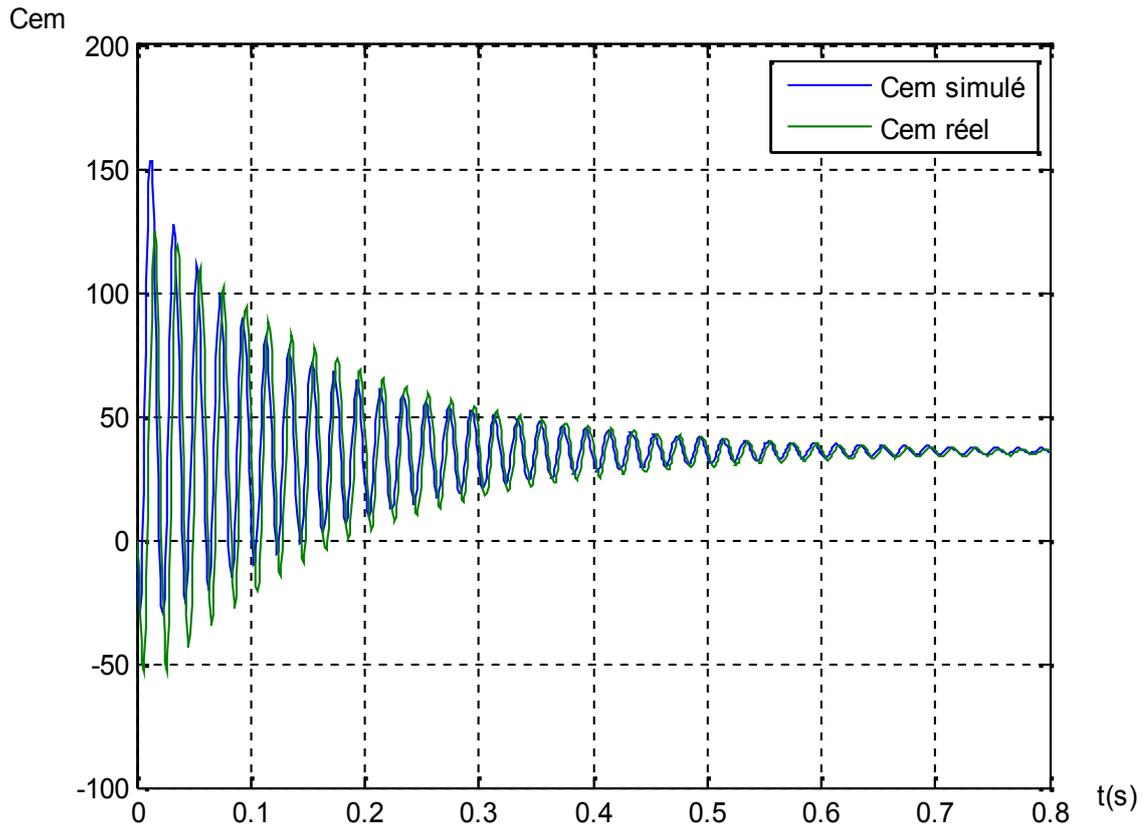


Figure 4.14 : Superposition du couple du modèle réel et du couple global estimé à partir des modèles réduits lent et rapide

4.3.3 Résultats des tests.

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

Notre système est composé de deux entrées : les tensions sinusoïdales v_s , v_s et d'une sortie représentant le couple C_{em} . Les deux entrées v_s , v_s sont décomposées en deux entrées lentes $v_s (l)$, $v_s (l)$ et deux entrées rapides $v_s (r)$, $v_s (r)$.

Dans notre réseau "ANGN", on fait varier, dans un premier temps, le nombre de réseaux neuronaux experts (sous-réseaux rapides), afin de déterminer le

nombre de régimes. On remarque que quelle que soit l'architecture adoptée pour les réseaux experts et pour le réseau superviseur glial, seulement quatre régimes ont été détectés. Dans un deuxième temps, différentes architectures de sous réseaux experts et superviseur, ont été testées, ce qui nous a conduit à une architecture optimale pour chaque type de sous réseau.

On notera qu'il est difficile de déterminer au départ, le nombre de sous réseaux experts pour un problème donné et qu'un nombre insuffisant de ces sous réseaux limite les performances du réseau global, en revanche, un nombre d'experts plus élevé que nécessaire, n'améliore aucunement ces performances. Pour notre application, le réseau neuroglial global, se compose de quatre experts et d'un réseau superviseur glial.

Les quatre réseaux experts possèdent des architectures semblables, composées d'une couche d'entrée de quatre neurones et d'un neurone de sortie, estimant les réponses rapides du réseau "ANGN" global. Le réseau superviseur glial est composé d'une couche d'entrée de quatre neurones et d'un neurone de sortie pour la sélection des experts et pour l'estimation de la réponse lente du réseau global.

Pour l'obtention des performances optimales (erreur, vitesse de convergence, configuration neuronale réduite,...) on adopte, comme mentionné plus haut, deux algorithmes d'apprentissage différents. On terminera notre application par une étude comparative des résultats obtenus par notre réseau neuroglial et ceux obtenus par utilisation d'un réseau de neurones classique et modulaire. Pour l'évaluation de nos résultats, on utilise l'erreur quadratique moyenne (EQM), donnée par :

$$EQM = \frac{1}{2} \sum_{i=1}^m (y_d - y_e)^2 \quad (4.36)$$

Où : y_d représente la sortie désirée et y_e la sortie estimée.

4.3.4 Premier algorithme d'apprentissage

Le réseau glial superviseur, estime le couple lent et sélectionne un expert parmi les quatre, séquentiellement en lui affectant un poids unité ; des poids de valeurs zéro sont, alors affectés aux trois autres experts et cela pour chaque entrée $(v_{s(l)i}, v_{s(r)i})$, $i = 1,2,3,4$. La sortie globale des réseaux neuronaux rapides est alors égale à la somme des sorties y_{ri} de chaque expert pondérées par leurs coefficients respectifs g_i .

$$y_r = g_1 y_{r1} + g_2 y_{r2} + g_3 y_{r3} + g_4 y_{r4}$$

La sortie, du réseau expert sélectionné y_{ri} représente le couple rapide $cem_{(r)i}$ correspondant à la sortie du réseau glial superviseur qui représente le couple lent $cem_{(l)i}$.

▪ Apprentissage du réseau glial superviseur :

Le réseau glial superviseur apprend la relation entre l'espace des entrées $v_{s(l)}$, $v_{s(r)}$ et le couple lent $C_{em(l)}$ de la machine, représentant la sortie. D'après la Fig. 4.15(a), on remarque que la courbe de l'erreur se stabilise à une valeur de : $7.23E-32$ et qu'une soixantaine d'itérations sont suffisantes pour l'apprentissage du modèle lent.

Ces résultats ne sont pas exceptionnels. Ils s'expliquent par le fait que la fonction soit 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 globale de la machine.

▪ Apprentissage des experts:

Ces réseaux, au nombre de quatre, vont apprendre le modèle réduit rapide de la machine. Les entrées sont les tensions $V_{s(r)}$ et $V_{s(l)}$, la sortie représente le couple rapide $C_{em(r)}$. La Fig. 4.15(b) montre les résultats de performance du réseau expert 4. On remarque que la valeur minimale, qui est de l'ordre de $3.0205E-29$, est atteinte juste après soixante itérations. Les performances des

quatre sous réseaux experts et celles du réseau neuroglial global sont illustrées en tableau 4.2:

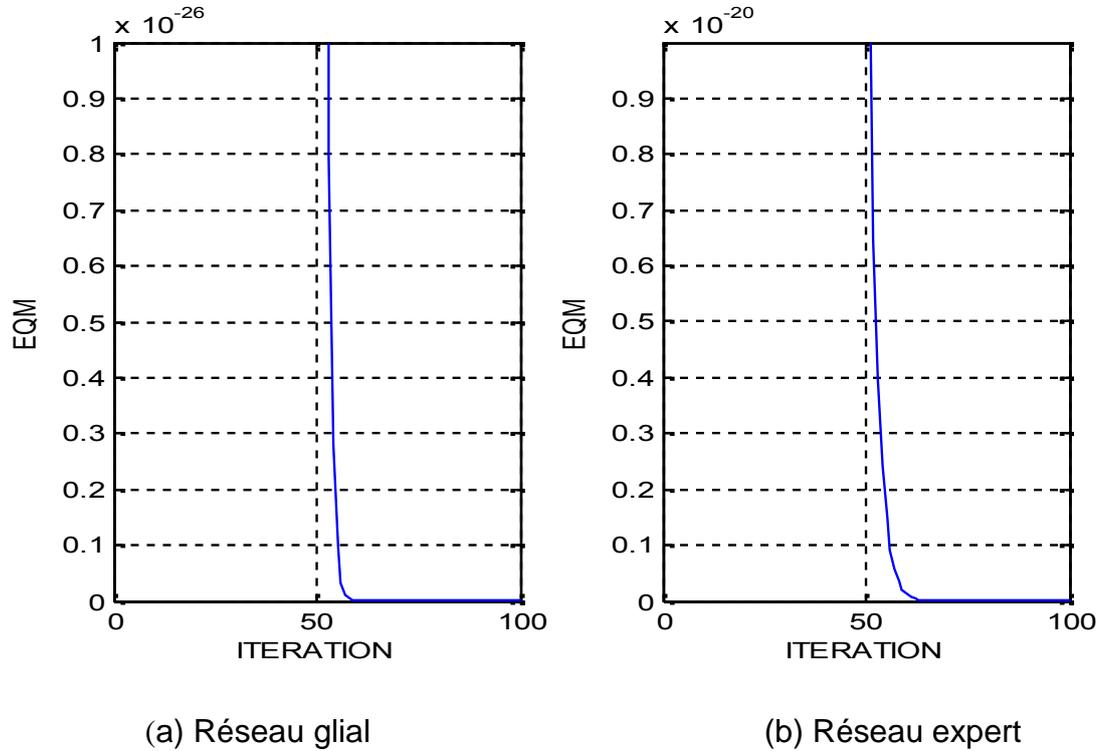


Figure 4.15 : Evolution des EQM_s

Tableau 4.2 : Performances du réseau "ANGN" relatives au premier algorithme.

| | <i>Nombre d' iterations</i> | <i>EQM</i> |
|---------------|-----------------------------|------------|
| Expert1 | 70 | 1.7014E-28 |
| Expert2 | 60 | 2.0550E-28 |
| Expert3 | 70 | 2.1211E-28 |
| Expert4 | 70 | 3.0205E-29 |
| Réseau rapide | | 6.1980E-28 |
| Réseau glial | 60 | 7.2300E-32 |
| Réseau ANGN | | 6.1980E-28 |

D'après le tableau 4.2, on remarque que pour notre application, l'erreur quadratique moyenne du modèle Neuroglial est très faible et est de l'ordre de $6.1980E-28$. Pour améliorer les performances du couple de la machine asynchrone, on adopte, dans ce qui va suivre, un deuxième algorithme appelé : réseau glial softmax.

4.3.5 Deuxième algorithme d'apprentissage (réseau glial softmax).

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.

Pour chaque entrée présentée, les quatre sous réseaux experts sont mis en compétition, le réseau glial estime la réponse lente de la machine et sélectionne le sous réseau expert vainqueur comme source de la réponse rapide estimée. Chaque expert se spécialise dans une région de l'espace d'entrée. Les valeurs des coefficients g_i déterminent le domaine de validité de chaque expert. Le coefficient g_i peut être pris comme la probabilité, à posteriori, de l'expert i sachant x_{ij} . Le domaine de validité de l'expert i correspond au domaine où g_i est égal à 1.

La Fig. 4.16(a) montre la bonne convergence de cet algorithme d'apprentissage. La valeur minimale, qui est très proche de zéro, de l'ordre de $7.627E-30$, se produit après seulement 2 itérations. L'algorithme a convergé parfaitement et rapidement.

La Fig. 4.16(b) montre l'évolution de l'erreur quadratique moyenne pour chacun des quatre sous réseaux experts. Ces informations permettent d'estimer la première partie de la réponse rapide (x_{r1}). On remarque que c'est le quatrième expert qui est le gagnant de la compétition ($g_4=1$) donc la réponse rapide correspondante est la bonne.

On remarque aussi dans la même Fig. 4.16(b) la grande vitesse de convergence de ce deuxième algorithme. La valeur minimale de l'erreur est de l'ordre de $3.8447E-31$ et elle est atteinte après 2 itérations seulement, par le quatrième expert qui est le vainqueur de la compétition dans la première région de l'espace d'entrée x_{r1} .

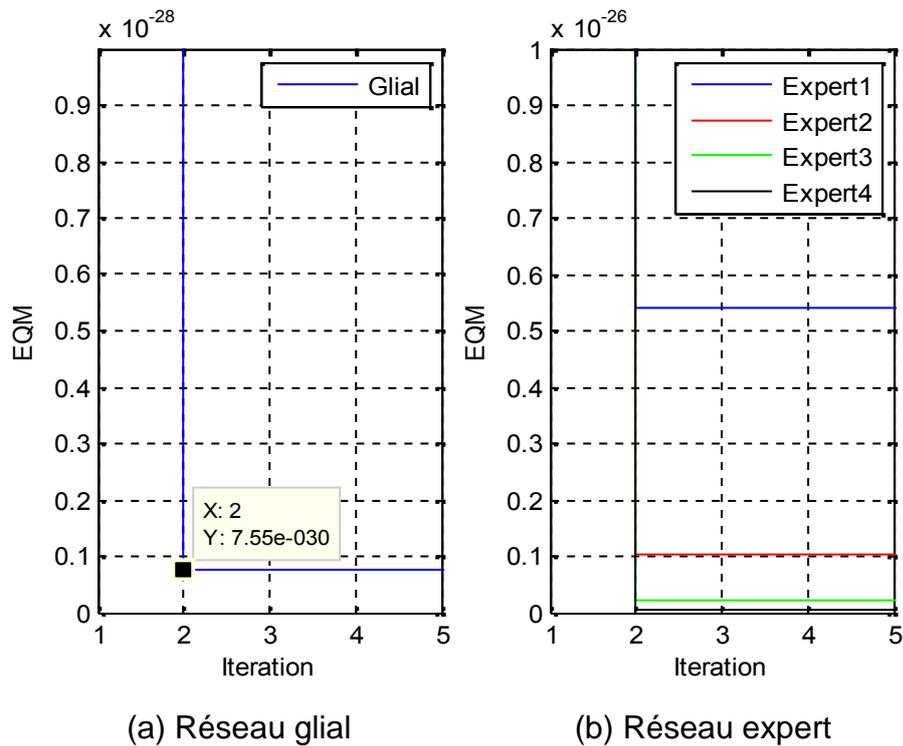
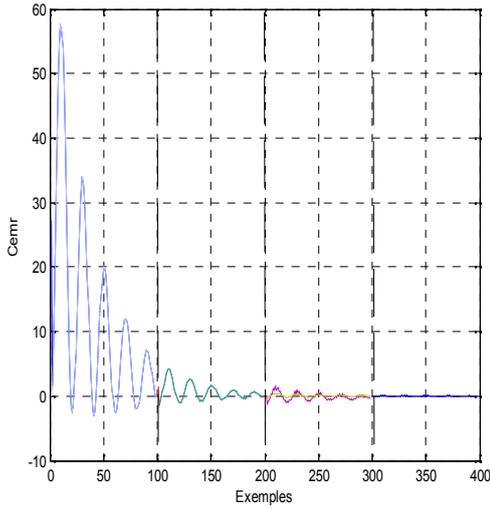
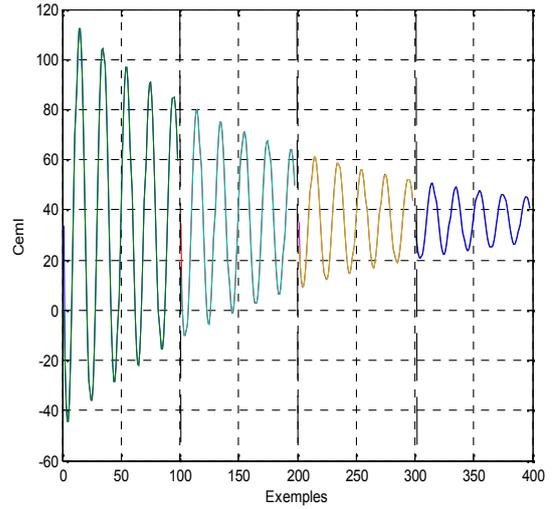


Figure 4.16 : L'évolution des EQM_s



(a) Le couple rapide



(b) le couple lent

Figure 4.17 : Les couples réduits estimés par le 2^{ème} algorithme

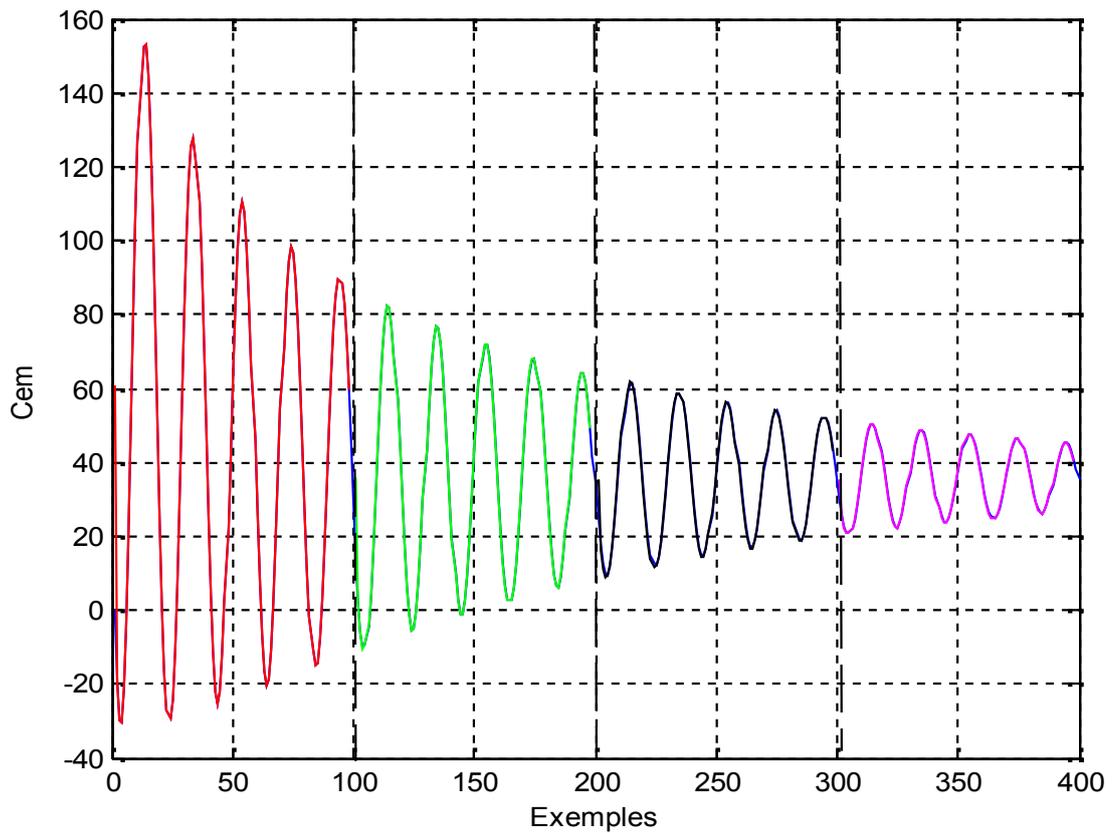


Figure 4.18 : Le couple global du réseau "ANGN" estimée par le 2^{ème} algorithme

La combinaison du réseau glial Fig. 4.17(b) et du réseau rapide fig. 4.17(a) donne le réseau Neuroglial Fig. 4.18. L'erreur est de l'ordre de $21.0974E^{-30}$. Ceci nous amène à conclure que les excellentes performances du système en termes de vitesse de convergence et de diminution de l'erreur quadratique moyenne sont en grande partie dues à la taille réduite des sous réseaux, ainsi qu'au nombre d'exemples, très limités, présentés à l'entrée de chaque sous réseau expert. Ces atouts, constituent les principales propriétés de la modularité. En outre, la répartition binaire des coefficients g_i , s'est avérée très efficace pour la sélection du meilleur expert. Pour le quatrième expert, on constate que dès que $g_4 = 1$, la sortie de cet expert (y_{r4}) correspond bien à la portion de la réponse rapide désirée. Ce qui fait de cet expert, le gagnant de la compétition. Inversement, lorsque que la sortie de ce quatrième expert est incorrecte, cet expert est désélectionné et g_4 s'annule automatiquement. Les autres experts se comportent d'une manière similaire. Les résultats de la simulation sont illustrés dans le tableau 4. 3 suivantes :

Tableau 4.3 : Poids et performances des experts pour chaque domaine de validité des données

| Domaine de validité | Expert vainqueur | g_1 | g_2 | g_3 | g_4 | EQM |
|----------------------------|------------------|-------|-------|-------|-------|------------|
| La 1 ^{ère} partie | Expert4 | 0 | 0 | 0 | 1 | 7.7759E-30 |
| La 2 ^{ème} partie | Expert3 | 0 | 0 | 1 | 0 | 3.7791E-30 |
| La 3 ^{ème} partie | Expert2 | 0 | 1 | 0 | 0 | 5.0512E-31 |
| La 4 ^{ème} partie | Expert1 | 1 | 0 | 0 | 0 | 8.9189E-30 |

Les résultats de simulation classés dans le tableau 4.3 représentent les EQMs 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 calculée par la somme des erreurs des quatre experts vainqueurs pour chaque partie. Cette erreur est de l'ordre de $9.8063E-30$.

Ces résultats montrent une nette amélioration des performances du modèle et on peut affirmer la grande efficacité de ce deuxième algorithme par rapport au premier.

4.3.6 Comparaison entre les différentes architectures

Pour mettre en évidence les avantages des modèles modulaires neuroglial par rapport à d'autres modèles neuronaux modulaires et classiques. On procède dans ce qui va suivre, en gardant la même base de données, à une étude comparative entre :

- Le modèle à réseau de neurone unique.
 - Le modèle modulaire à deux réseaux de neurones lent et rapide.
 - Et le modèle mélange d'experts (ME).
- Modèle à réseau de neurone unique

La Fig. 4.19 montrent les performances d'apprentissage du réseau de neurone unique, on remarque que l'erreur quadratique moyenne est de l'ordre de 0,000174207 obtenus après 5000 itérations avec une architecture composée de trois couches : une couche d'entrée de dix neurones; une couche cachée de dix autres neurones et une couche de sortie constituée d'un seul neurone. On remarque que la convergence est lente à cause de la taille du réseau.

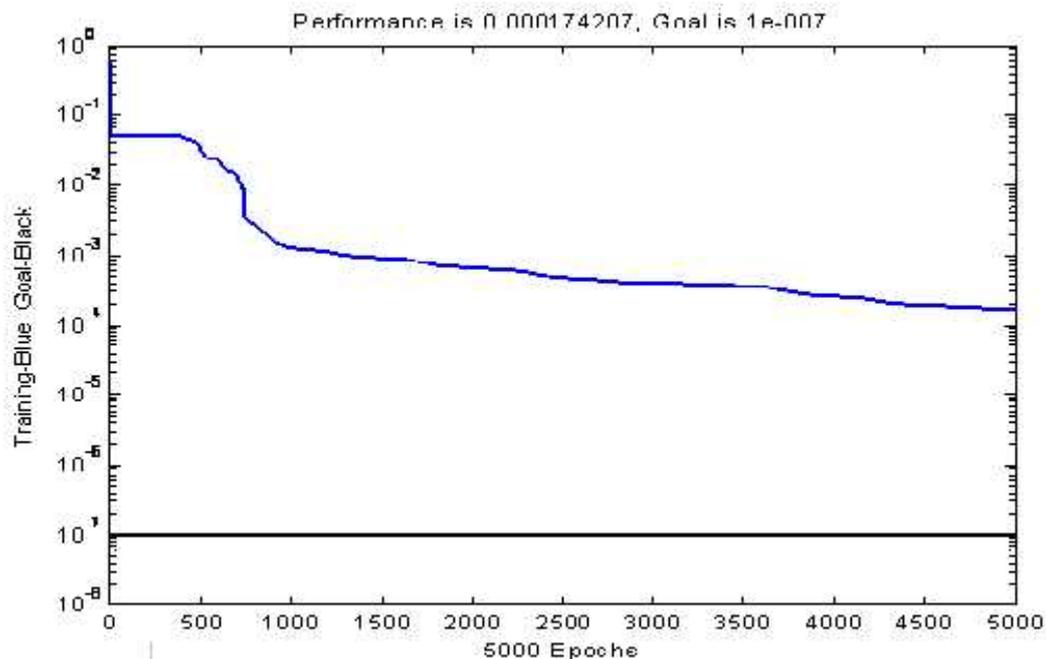


Figure 4.19 : Evolution du EQM du réseau de neurones unique

▪ Modèle modulaire à deux réseaux de neurones lent et rapide.

L'apprentissage du modèle modulaire à deux réseaux de neurones de la Fig 4.20, le premier estimant la réponse lente, le deuxième la réponse rapide, donne les résultats illustrés dans les Figs. 4.21(a) et 4.21(b). Les architectures des deux réseaux, lent et rapide, sont semblables. Chacun des deux réseaux est composé d'une couche d'entrée contenant quatre neurones et d'une couche de sortie d'un seul neurone. On remarque dans la Fig. 4.21(a), que l'erreur du réseau lent se stabilise à une valeur de : $7.23E-32$ et qu'une centaine d'itérations sont suffisantes pour apprendre le modèle lent. La Fig. 4.21(b) montre que l'erreur du réseau rapide atteint la valeur $1.082E-27$ après soixante dix itérations. Les performances du modèle global dépendent des performances du modèle réduit lent et celles du modèle réduit rapide.

Ces résultats mettent en évidence les grands avantages offerts par le modèle modulaire à deux réseaux de neurones par rapport au modèle à réseau de neurones unique classique. Ces avantages concernent, d'une part, la vitesse de convergence et d'autre part, l'erreur minimale quadratique, sans compter la complexité de l'architecture, qui se trouve, considérablement réduite.

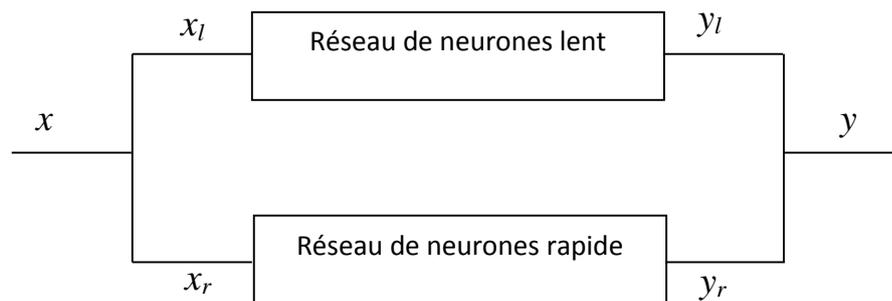
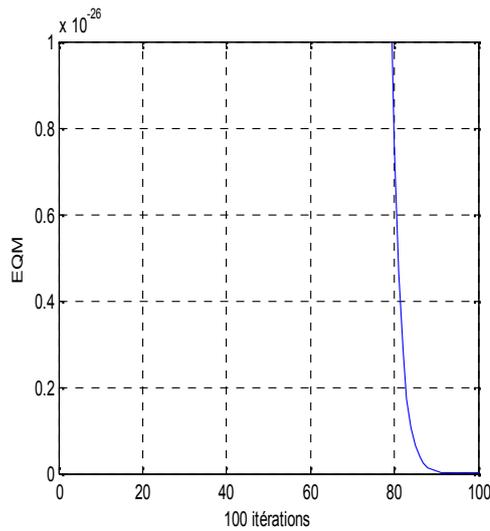
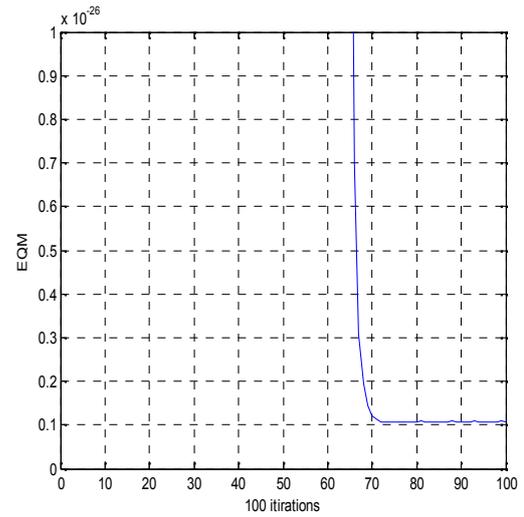


Figure 4.20 : Architecture modulaire à deux réseaux



(a) Réseau lent



(b) Réseau rapide

Figure 4.21: Evolution de l'EQM

- le modèle mélange d'experts (ME).

Dans cette section, nous allons apprendre le modèle de la machine asynchrone que nous avons décrit plus haut en utilisant l'architecture mélange d'experts. Le modèle est composé de quatre réseaux de neurones de type MLP qui sont mis en compétition. Le modèle à apprendre est fonction de deux entrées: les tensions sinusoïdales ϵ_{sR} , ϵ_{sS} . La sortie est le couple C_{em} .

Les résultats du mélange d'experts obtenus sont illustrées dans la Fig. 4.22, on remarque que le gagnant de la compétition du mélange d'experts à converger à une valeur minimale de $7.934 \text{ E-}24$ au bout de 90 itérations.

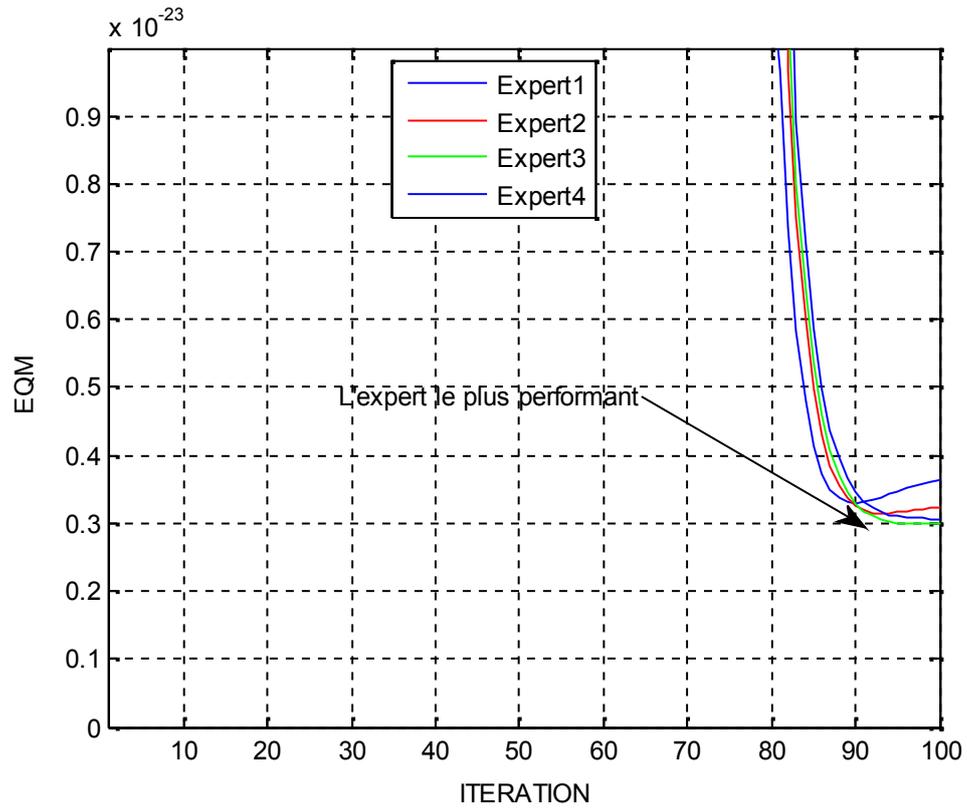


Figure 4.22 : Evolution des EQM_s des experts du réseau ME

Les performances des différents modèles sont présentées dans la Fig. 4.23 à des fins de comparaison.

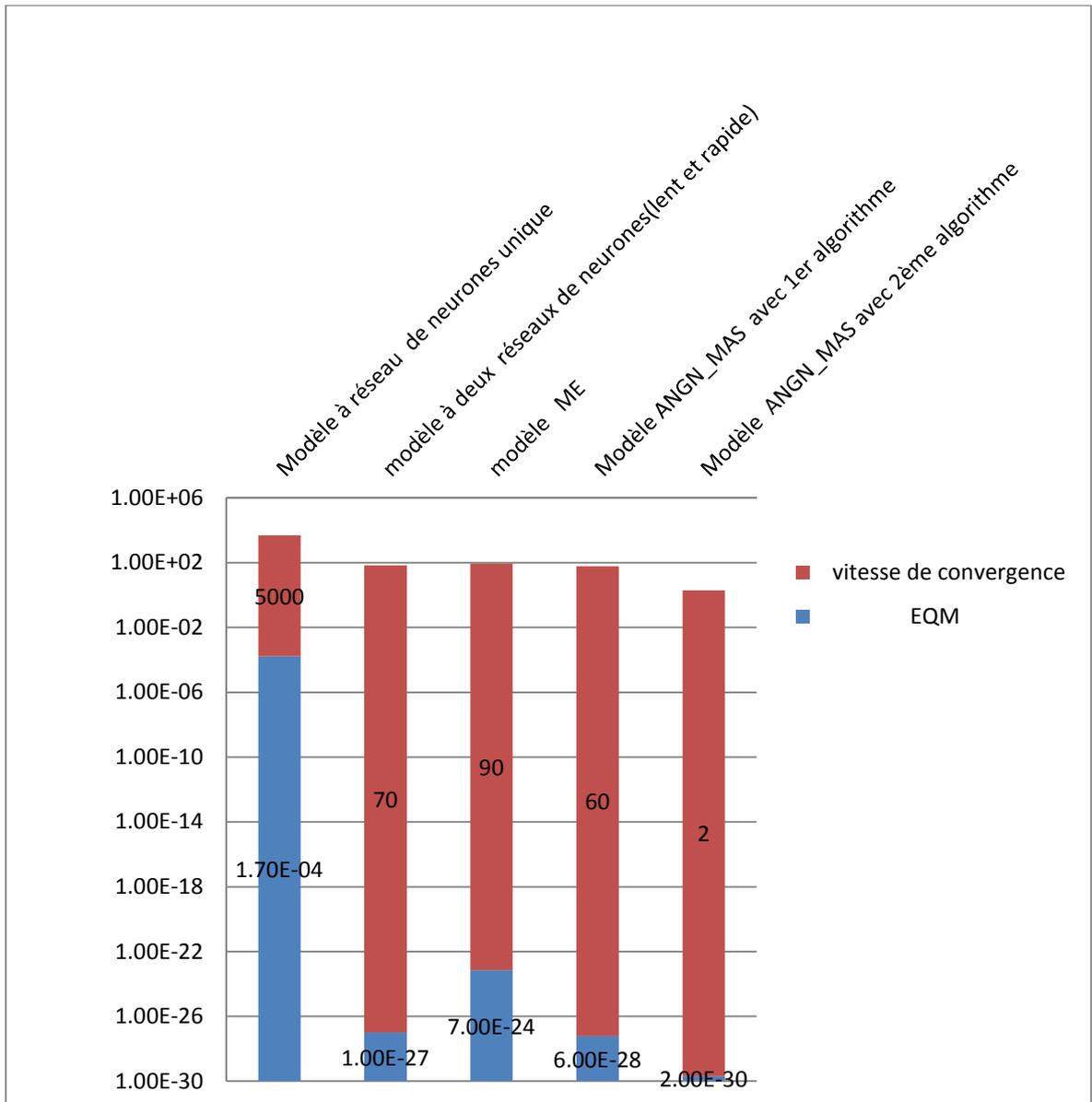


Figure 4.23 : Les performances des différents modèles

D'après l'histogramme de la Fig. 4.23, il est très clair que les valeurs des EQM_s de l'architecture "ANGN" sont beaucoup plus petites que les valeurs des autres modèles, ce qui signifie que les algorithmes proposés dans ce travail peut produire des résultats plus précis. Aussi, on peut constater l'excellente performance du modèle proposé en termes de vitesse de convergence et de réduction de l'erreur EQM optimale par rapport aux performances des autres

modèles. Cela est dû en grande partie à la petite taille des sous-réseaux et au nombre très limité d'exemples présentés à l'entrée de chaque expert. Ces avantages sont les principales propriétés de la modularité.

Le choix du «meilleur» algorithme n'est pas évident et peut se faire uniquement sur des considérations de performances concernant l'erreur d'évolution du réseau, vitesse de convergence, la taille du réseau et l'interprétabilité, 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. Le modèle neuroglial a l'avantage de pouvoir remédier partiellement à ce problème. Il a l'avantage indéniable, de par leur nature, de pouvoir effectuer des partitions du modèle selon des critères physiques, 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 le 2^{ème} algorithme proposé, donne de bonnes performances. En effet l'erreur quadratique atteint 17.455E-30 après 2 itérations. Cette valeur est obtenue avec un minimum de calcul et une grande vitesse de convergence.

4.4. Identification des paramètres de la machine asynchrone

Afin de démontrer l'efficacité de l'architecture " ANGN " proposée, une simulation est effectuée sur les modèles réduits lents et rapides de la machine asynchrone (MAS) pour identifier les grandeurs suivants :

$$\omega \quad \varphi_r \quad \varphi_r \quad i_{\alpha} \quad i_{\beta}$$

Où

$$\begin{cases} I_{rs(l)} = \frac{1}{(1+r)B_r L_r} R(n) (w_{srs(l)} - T_s v_{srs(l)}) \\ I_{rs(r)} = -\frac{L_s}{M} R(n) I_{srs(r)} \end{cases} \quad (4.37)$$

$$\begin{cases} W_{rrs}(t) = \frac{1}{B_r} R(r) \left(\left(1 - \frac{rt}{1+r} \right) W_{srs}(t) - \frac{T_{sp}}{1+r} v_{srs}(t) \right) \\ W_{rrs}(r) = -\frac{L_s}{B_r} R(r) I_{srs}(r) \end{cases} \quad (4.38)$$

$$\frac{d}{dt} = \frac{p}{j} C_e - \frac{p}{j} C_r - \frac{f}{j} \omega \quad (4.39)$$

Les résultats de simulation du 2^{ème} algorithme d'identification proposé dans la section (3.6.2) et l'algorithme d'identification utilisant le RN dynamique et la technique des perturbations singulières proposé dans la réf [17] sont présentés aux Figs. 4.24- 4.28. Pour une comparaison la plus juste, nous choisissons les mêmes paramètres de la machine asynchrone.

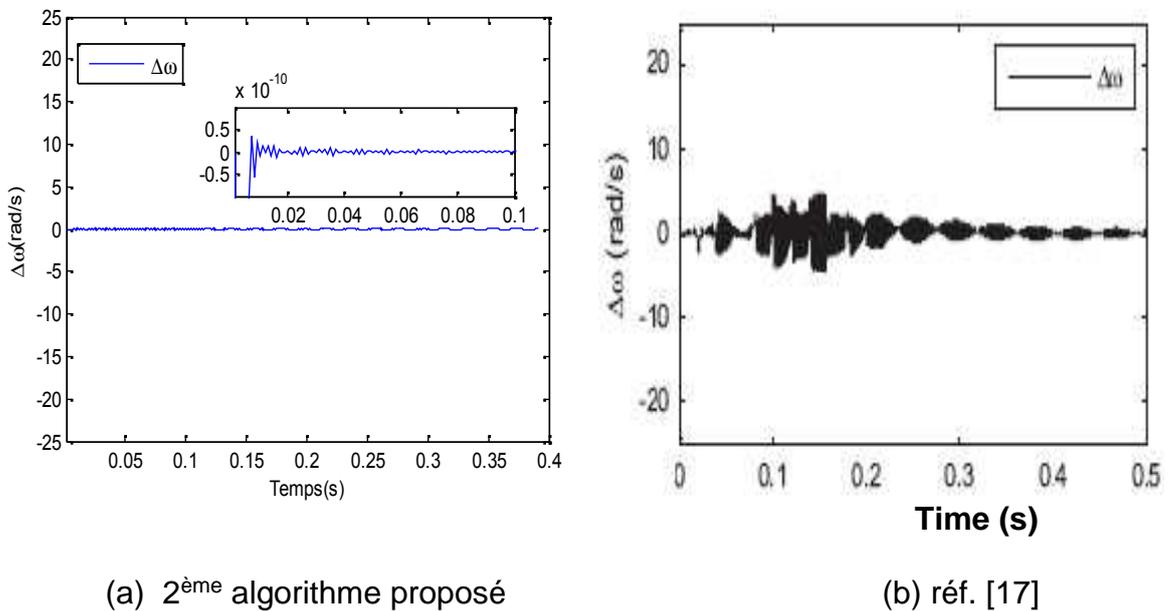
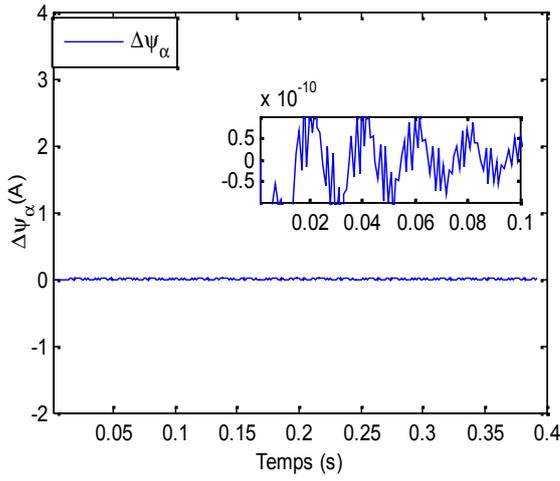
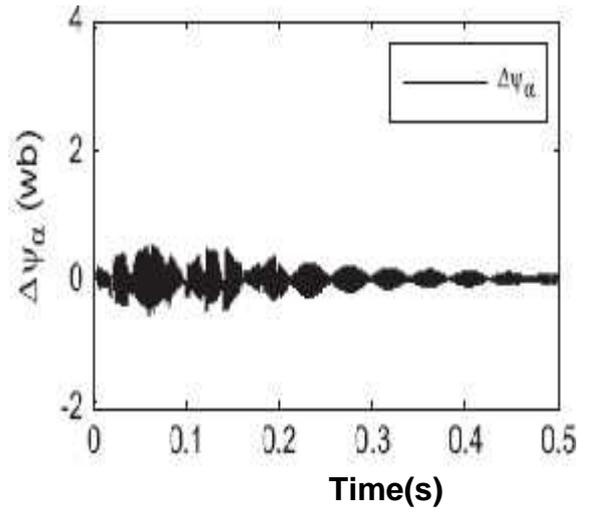


Figure 4.24 : L'erreur de la variable d'état ω

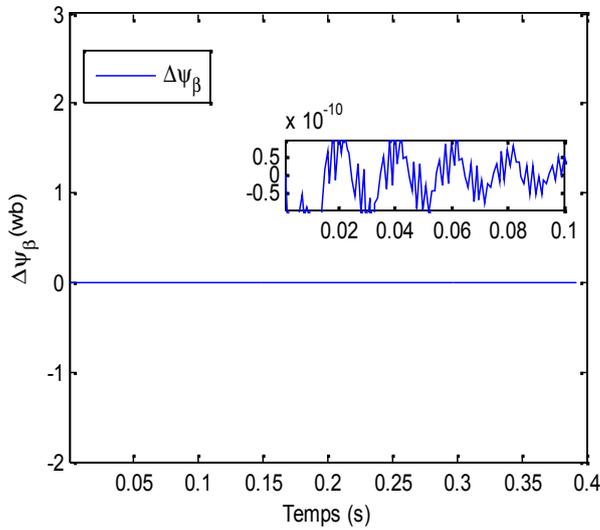


(a) 2^{ème} algorithme proposé

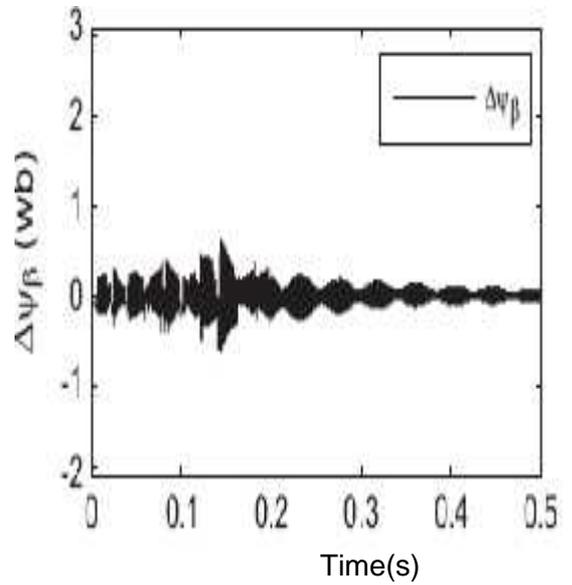


(b) réf. [17]

Figure 4.25: L'erreur de la variable d'état φ_r

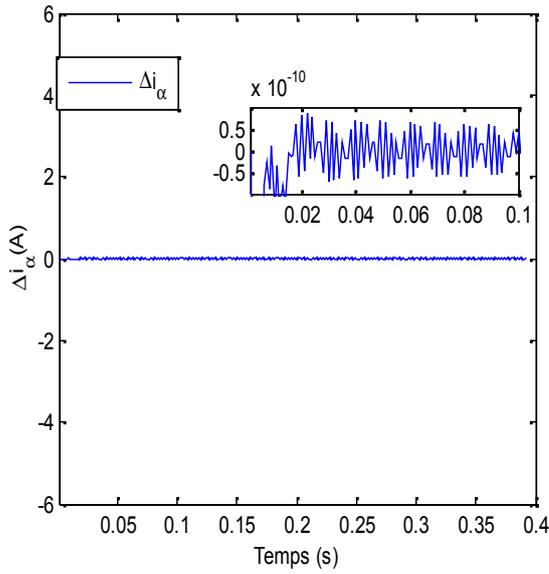


(a) 2^{ème} algorithme proposé

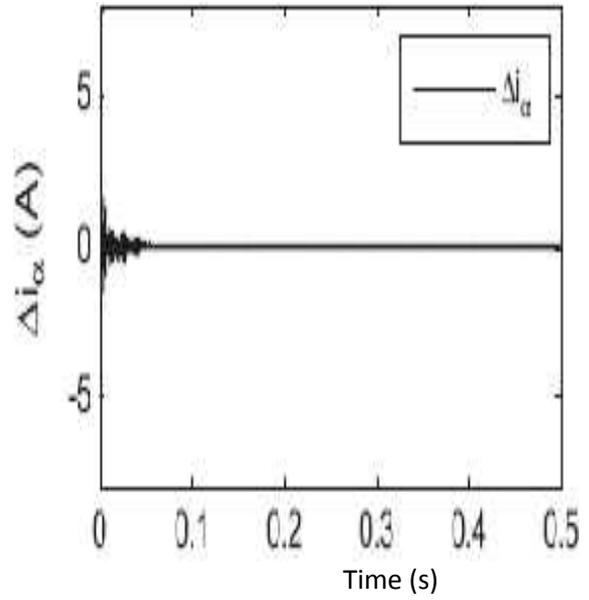


(b) réf. [17]

Figure 4.26 : L'erreur de la variable d'état φ_r

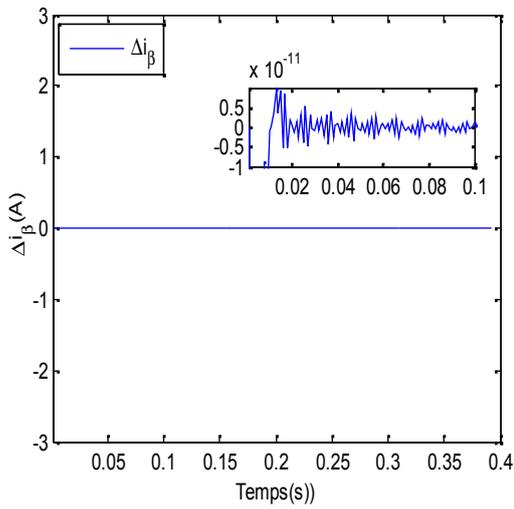


(a) 2^{ème} algorithme proposé

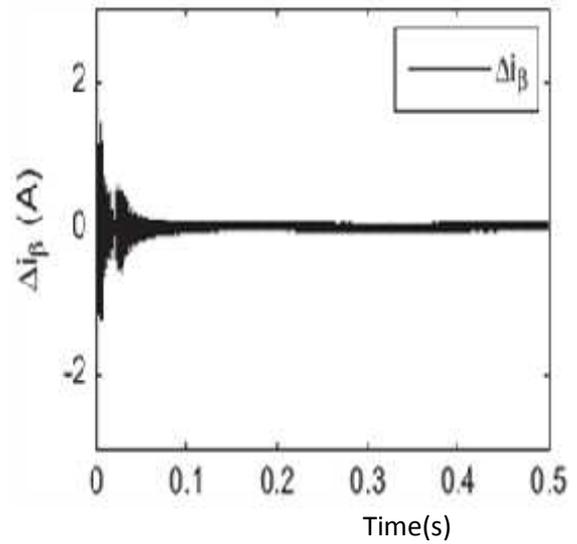


(b) réf. [17]

Figure 4.27 : L'erreur de la variable d'état \bar{i}_α



(a) 2^{ème} algorithme proposé



(b) réf. [17]

Figure 4.28 : Erreur de la variable d'état \bar{i}_β

Dans les Figs. 4.24(a) - 4.28(a), les sous-figures de gauche sont les résultats du 2^{ème} algorithme de l'architecture " ANGN " proposé dans la section (3.6.2), et les sous-figures de droite sont les résultats des erreurs de la méthode proposée dans la réf. [17].

Dans la Fig. 4.24(a), le tracé d'agrandissement des premières phases du processus d'identification montre que l'erreur d'identification de ω est presque égale à 0 depuis le début du processus d'identification, ce qui signifie que la précision de l'algorithme d'identification proposé dans cette section est très élevée et que la vitesse de convergence est très rapide. Par contre, dans la Fig 4.24(b) qui montre le résultat de l'algorithme d'identification proposé dans la réf [17] , de nombreuses oscillations sur l'erreur de ω peuvent être observées au début du processus d'identification. En outre, des erreurs plus importantes peuvent être observées pendant tout le processus d'identification.

Des phénomènes similaires existent aux Figs 4.25 - 4.28. On peut conclure que les résultats d'identification de $\omega, \varphi_r, \varphi_r, i_\alpha$ e i_β sont beaucoup plus précis lorsqu'on utilise le 2^{ème} algorithme de " ANGN " proposé dans la section (3.6.2). Les erreurs sont grandement réduites par rapport aux résultats obtenus avec l'algorithme proposé dans la réf. [17]. De plus, le 2^{ème} algorithme proposé de l'architecture "ANGN" converge beaucoup plus rapidement que l'algorithme d'identification proposé dans [17].

L'indice de performance RMS (la racine de l'erreur quadratique moyenne) peut également être utilisé pour illustrer l'efficacité du 2^{ème} algorithme de l'architecture "ANGN". Les RMS_s d'identification $\Delta\omega, \Delta\varphi_r, \Delta\varphi_r, \Delta i_\alpha$ e Δi_β sont calculés comme suit:

$$\text{RMS} = \sqrt{\left(\sum_{i=1}^n e^2(i)\right)/n} \quad 4.5$$

où n est alors le nombre de pas de simulation, e est la différence entre les variables d'état dans le modèle et le système à la $i^{\text{ème}}$ étape.

Les valeurs des erreurs RMS_s pour les variables d'état $\omega, \varphi_r, \varphi_r, i_\alpha$ e i_β sont présentées dans le tableau 5.

D'après le tableau 4.4, on remarque que les performances ont été améliorées par rapport à celles de la réf. [17], il est très clair à partir de ces chiffres que les erreurs d'identification de l'architecture "ANGN" sont considérablement réduites. Nous pouvons confirmer que les variables d'état de l'architecture "ANGN" suivent celles du système non linéaire MAS avec plus de précision et de rapidité. Ceci est dû à la bonne séparation des modes lent et rapide, ce qui a pour effet de réduire considérablement la complexité des réseaux de l'architecture "ANGN".

Tableau 4.4 : Les valeurs des erreurs RMS des variables d'état.

| RMS | ω | $\varphi_r(w)$ | $\varphi_r(w)$ | $i_{\alpha(A)}$ | $i_{\beta(A)}$ |
|----------|-----------------------|------------------------|-----------------------|-----------------------|---------------------|
| Réf [17] | 0.1431 | 0.0575 | 0.05334 | 0.0446 | 0.0452 |
| ANGN | $3.65 \cdot 10^{-15}$ | $1.029 \cdot 10^{-23}$ | $3.33 \cdot 10^{-24}$ | $3.33 \cdot 10^{-23}$ | $.8 \cdot 10^{-23}$ |

4.5. Conclusion

Dans ce chapitre, l'architecture "ANGN" est appliquée à la modélisation et l'identification des systèmes singulièrement perturbés, les deux algorithmes

développés sont testés sur le modèle de la machine asynchrone et le modèle du moteur MCC. Les modèles originaux sont mis sous des formes singulièrement perturbées.

Les modèles lents et rapides sont obtenus après deux étapes:

- Application de l'approche géométrique basée sur les cercles de Gerschgorin pour découpler les variables lentes et rapides.
- Application de la technique des perturbations singulières pour le développement des modèles réduits.

Les modèles réduits "lent et rapide" du moteur MCC ainsi de la machine asynchrone constituent des modèles adéquats pour notre travail. Pour la modélisation de la machine asynchrone, Les résultats de simulation de l'apprentissage du modèle ANGN_MAS avec le premier et le deuxième algorithme ont montré des performances satisfaisantes en termes de vitesse de convergence, de réduction de la complexité, et l'erreur quadratique moyenne minimale, par rapport au modèle à réseau unique, au modèle à deux réseaux de neurones lent et rapide et au modèle modulaire ME. Ceci met en évidence qu'une telle architecture est la mieux adaptée aux systèmes complexes présentant deux échelles de temps. De plus, à partir des résultats de simulation, on a observé que la convergence du second algorithme basé sur la fonction softmax est plus rapide que celle du premier algorithme.

Pour l'identification de la machine asynchrone et du moteur MCC, Les résultats de la simulation montrent que les résultat obtenus par le 2^{ème} algorithme proposé ont de meilleures performances que celui de la réf. [17] . Une convergence plus rapide est obtenue avec une plus grande précision.

CONCLUSION GENERALE

Dans cette thèse de doctorat, une nouvelle architecture, qui émule une réalité biologique, découverte récemment appelée réseau neuroglial artificiel "ANGN" basée sur le concept puissant modularité est proposée. Deux algorithmes modulaires sont développés pour tester son efficacité et sa performance.

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

En premier lieu, 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 retenu que dans notre cerveau l'information circule selon deux voies : une rapide, réseau de neurones et l'autre lente, réseau glial et que le réseau de neurones est alimenté et contrôlé par le réseau glial.

Nous avons utilisé deux formes de modularité dans la conception des architectures neuronales : la modularité à modèle multiples et la modularité vraie. La première utilise plusieurs modules pour traiter le même problème. Les sorties de tous les modules sont combinées pour calculer la sortie globale. La seconde est une composition de plusieurs modules qui apprennent séparément tous les sous-problèmes d'un problème complexe.

La validation de l'approche "ANGN" a été effectuée sur les systèmes complexes à multi_échelles de temps, ce sont des systèmes singulièrement perturbés.

Pour la mise sous forme standard singulièrement perturbée des systèmes, nous avons appliqué l'approche géométrique basée sur les cercles de Gerschgorin pour le découplage et l'identification des variables lentes et rapides, ainsi que la méthode des perturbations singulières pour le développement des modèles réduits.

La modélisation et l'identification des systèmes non linéaires singulièrement perturbés à l'aide de réseaux neurogliaux à plusieurs échelles de temps sont étudiées et comparées aux autres travaux de recherche. Deux nouvelles architectures d'identification sont proposées pour identifier les systèmes non linéaires singulièrement perturbés avec des modèles de systèmes ou des paramètres inconnus. Ces nouvelles architectures d'identification peuvent atteindre une précision d'identification élevée avec une convergence rapide. Sur la base des modèles réduits lents et rapides qui sont obtenus par la mise en forme des systèmes singulièrement perturbés, les deux identificateurs sont validés par des simulations.

Les principaux résultats de cette recherche sont énumérés comme suit:

- Une mise sous forme standard du moteur à courant continu est établie puis une 'identification utilisant le réseau "ANGN" est appliquée à la section 4.2 pour identifier le moteur à courant continu. Lorsque cette architecture d'identification de "ANGN" est utilisée, les sorties du "ANGN" suivent celles du système non linéaire de manière plus précise et rapide par rapport à ceux de la littérature.

- Dans la section 4.3, les deux architectures "ANGN" sont utilisées pour la modélisation de la machine asynchrone mise sous la forme standard singulièrement perturbée. Les deux algorithmes développés sont appliqués pour tester l'efficacité et la performance de l'architecture ANGN_MAS. Les modèles réduits lent et rapide sont obtenus après deux étapes :
 - Application de l'approche géométrique basée sur les cercles de Gerschgorin pour découpler les variables lentes et rapides.

 - Application de la technique des perturbations singulières pour le développement des modèles réduits.

Les résultats de simulation de l'apprentissage du modèle ANGN_MAS avec le premier et le second algorithme ont montré des performances satisfaisantes en

termes de vitesse de convergence, réduction de la complexité, et réduction de l'erreur quadratique moyenne minimale, par rapport au modèle à réseau unique, au modèle à deux réseaux de neurones lent et rapide et au modèle modulaire ME. Ceci met en évidence qu'une telle architecture est la mieux adaptée aux systèmes complexes présentant deux échelles de temps.

De plus, à partir des résultats de simulation, on a observé que la convergence du second algorithme basé sur la fonction softmax est plus rapide et plus précise que celles du premier algorithme.

- Sur la base des modèles réduits lent et rapide de la machine asynchrone, un identificateur "ANGN" utilisant le second algorithme basé sur la fonction softmax, est appliqué pour identifier les grandeurs de la machine à induction.

La simulation montre que, lorsque l'identification, proposée dans la section 4.4, est utilisée, les états estimés du système peuvent converger vers les états réels du système plus rapidement et plus précisément, par rapport aux résultats obtenus lorsque le schéma d'identification RN multicouches proposé dans [17] est utilisé. De plus, les erreurs d'identification peuvent être considérablement réduites lorsque l'algorithme d'identification basé sur la fonction softmax est utilisé. Cet algorithme peut atteindre une précision d'identification élevée avec une convergence rapide.

Nos perspectives concernent le développement d'une architecture " ANGN " plus sophistiquée qui donnera clairement de meilleures performances pour l'identification et le contrôle des systèmes à multi échelles de temps qui ont une dynamique complexe et qui demandent une grande précision telles que :

- Le domaine de la robotique où se présentent des dynamiques différentes
- les systèmes de contrôle de la trajectoire des vols d'essais

ANNEXE A

METHODE DES PERTURBATIONS SINGULIERES APPLIQUEE AUX SYSTEMES NON-LINEAIRES

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 [37]. 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 :

$$\begin{cases} \dot{x} = f(x, z, u, t, v) & x \in \mathfrak{R}^n \\ v \dot{z} = g(x, z, u, t, v) & z \in \mathfrak{R}^m \\ y = h(x, z) \end{cases} \quad (\text{A.1})$$

avec :

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

Les fonctions f , g sont supposées suffisamment continues et dérivables par rapport à leurs arguments x, z, u, v, t , et $t_0 \in [0, T]$ et en notant $x(t, v)$ et $z(t, v)$ la solution du problème original (A.1).

Le modèle (A.1) est dit sous forme singulièrement perturbée car :

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

$$g(x, z, u, t, 0) = 0 \quad (\text{A.2})$$

Propriété:

On dit que le modèle (A.1) est sous la forme standard singulièrement perturbée si et seulement si l'équation (A.2) admet k racines réelles distinctes $k \geq 1$ définies par :

$$z_i = q_i(x_i, u_i, t), \quad i = 1, 2, \dots, k \quad (\text{A.3})$$

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_i de l'équation (A.3).

L'indice « i » sur les variables indique qu'elles appartiennent au modèle écrit avec $v = 0$. En développant le modèle réduit caractérisé par $v = 0$, 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 :

$$\dot{x}_i = f(x_i, q(x_i, t), u_i, t, 0) \quad (\text{A.4})$$

On note la solution de (A.4) par $x_i(t, u_i)$. La variable z est exclu du modèle réduit et elle est remplacée par son régime quasi-permanent $q(x_i, t)$. La seule information qu'on peut obtenir sur z est de calculer :

$$z_i = q(x_i(t), t) \quad (\text{A.5})$$

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

La commande u est devenue u_i , car écrire $v = 0$ 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 (A.5) dans (A.1), soit :

$$\begin{aligned} \dot{x}_i &= f(x_i, q_i(x_i, u_i, t), u_i, t, 0) \\ y &= h(x_i, q_i(x_i, u_i, t)) \end{aligned} \quad (\text{A.6})$$

En gardant la même condition initiale sur x_l que sur x :

$$x_l(t_0) = x_0$$

Dans la suite nous utiliserons la notation plus compacte :

$$x_l = f(x_l, u_l, t) \quad (\text{A.7})$$

$$y = h(x_l, u_l, t) \quad (\text{A.8})$$

$$z_l(t_0) = q(x_l(t_0), u_l(t_0), t_0)$$

et :

$$z_l = q(x_l, u_l, t) \quad (\text{A.9})$$

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 :

$$v \frac{dz}{dt} = \frac{dz}{d\ddagger} \quad \text{avec} \quad \frac{d\ddagger}{dt} = \frac{1}{v}$$

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

$$\ddagger = \frac{t - t_0}{v}, \ddagger = 0, \text{ pour } t = t_0 \quad (\text{A.10})$$

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

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

$$z_r = z - z_l$$

Qui nous permet de représenter le système original comme suit :

$$\frac{dx}{dt} = f(x, z_r + q(x, u_l, t), u, t, v)$$

$$v \frac{dz_f}{dt} = g(x, z_r + q(x, u_l, t), u, t, v) - v \frac{\partial}{\partial x} q(x, u_l, t) \cdot \frac{\partial x}{\partial \dagger} - v \frac{\partial}{\partial t} q(x, u_l, t)$$

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

$$\begin{aligned} \frac{dx}{d\dagger} &= v f(x, z_r + q(x, u_l, t), u, t, v) \\ v \frac{dz_f}{d\dagger} &= g(x, z_r + q(x, u_l, t), u, t, v) - \frac{\partial}{\partial x} q(x, u_l, t) \cdot \frac{\partial x}{\partial \dagger} - v q(x, u_l, t) \end{aligned}$$

Le modèle réduit dans cette échelle de temps est obtenu en posant $v = 0$ et on suppose que $\frac{\partial u_l}{\partial \dagger} = O(v)$. Par conséquent, le système réduit vérifiant l'équation de couche limite s'écrit :

$$\begin{aligned} \frac{dz_r}{d\dagger} &= g(x_0, z_r(\dagger), z_l(\dagger), u(\dagger), t_0) \\ y_r &= h(x_0, z_r(\dagger) + z_l(\dagger)) \\ z_r(0) &= z_0 - z_l(t_0) \end{aligned} \tag{A.11}$$

Cette équation comporte le terme $u(\dagger)$ qui, à cette échelle de temps \dagger , correspond au comportement rapide de la variable u , défini par : $u_r = u - u_l$. Avec z_l, z_r décrivant respectivement les composantes lentes et rapides de z .

ANNEXE B

RESEAUX DE NEURONES ARTIFICIELS

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 des systèmes et les connaissances améliorées 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.

B.1 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 [64]. Les neurones sont des cellules du système nerveux spécialisées dans la transmission de l'information.

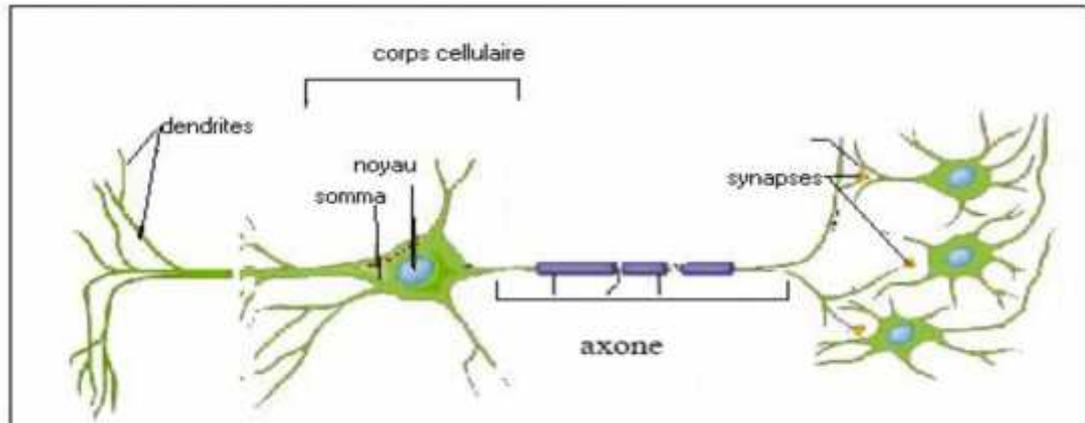


Figure B.1: Neurone biologique [64]

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 Fig. B.1.

B.1.1 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.1.2 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).

B.1.3 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 [64].

B.2. Fonctionnement du neurone biologique

L'information, sous forme de signaux électriques 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 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.

B.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 Fig. B.2.

La plupart des algorithmes connexionnistes en ce moment s'inspirent du modèle de la Fig. B.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 dont la valeur est celle de son activation. Si le neurone n'est pas activé, il ne transmet rien.

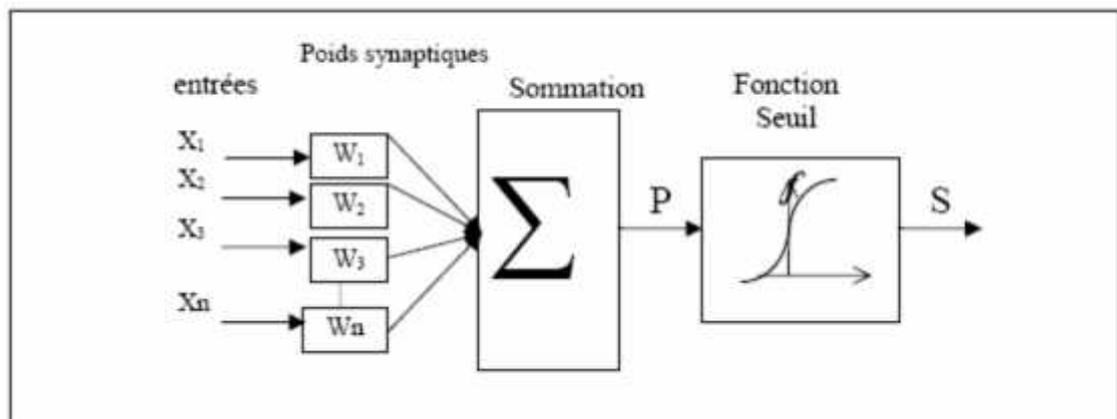


Figure B.2 : Modèle mathématique

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
$$P = \sum_{i=1}^n X_i W_i \quad (\text{B.1})$$

Où 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 et la fonction signe.

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

$$S = f(p) \quad (\text{B.2})$$

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

B.3.1 Fonctions d'activations

- Fonction binaire à seuil

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

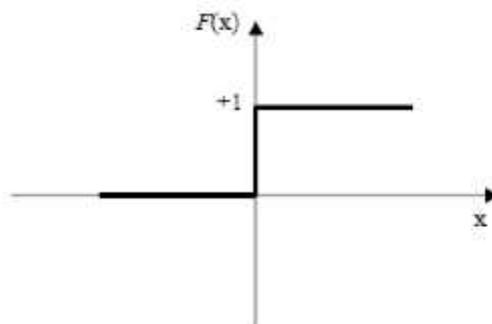


Figure B.3 : Fonction d'activation binaire à seuil.

- 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 B.4).

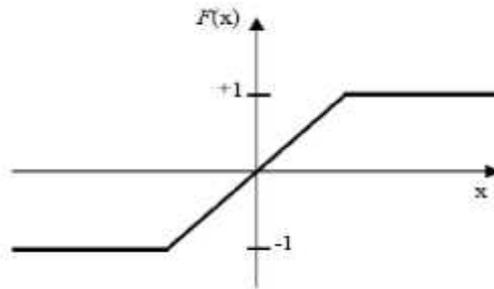


Figure B.4 : Fonction d'activation à rampe avec saturation.

- Fonction sigmoïde

La fonction sigmoïde est une fonction continue qui maintient la sortie dans l'intervalle $[0,1]$ Fig. B.5. Son avantage principal est l'existence de sa dérivée en tout point. Elle est employée en général dans le perceptron multicouche.

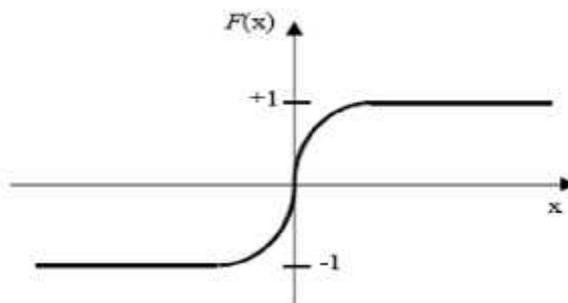


Figure B.5 : Fonction d'activation sigmoïde.

B.4. Réseaux de neurones

Un réseau de neurones peut être représenté par un graphe direct composé d'un ensemble de nœuds 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.

B.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.

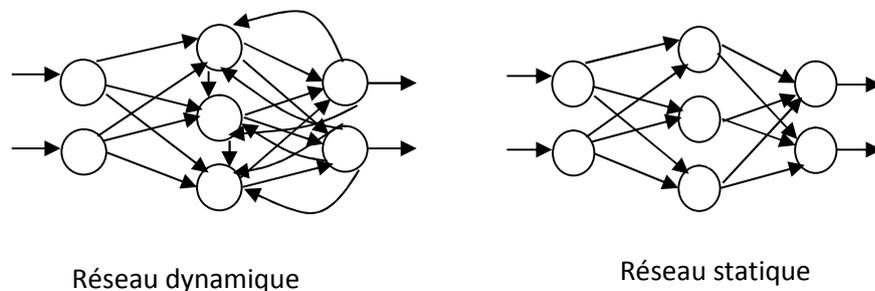


Figure B.6 : Réseau dynamique et statique

B.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.

B.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 [64]. 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.

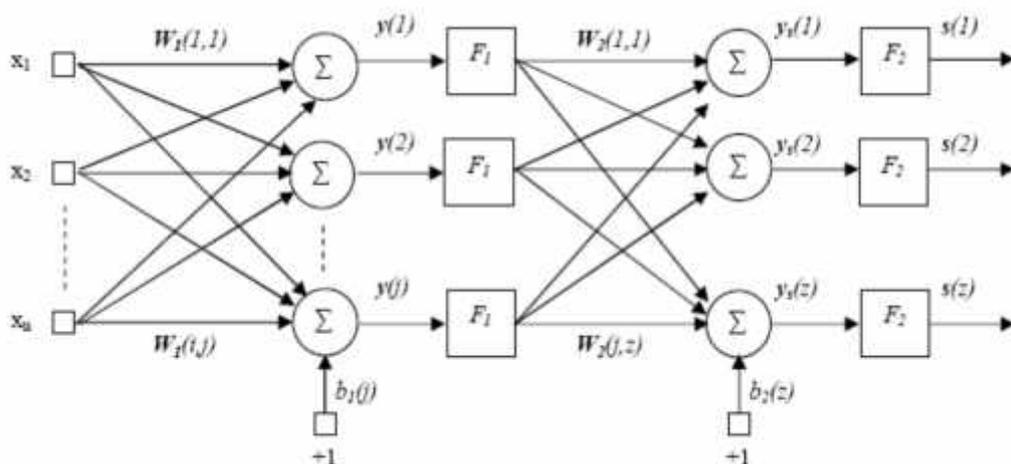


Figure B.7 : Réseau de neurones multicouches.

B.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 [64]. 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 seuil 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.

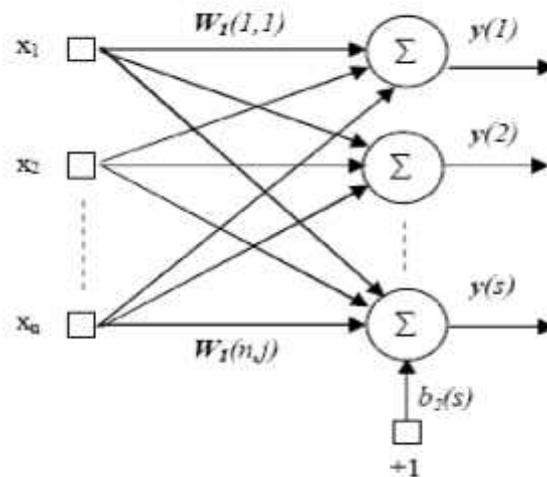


Figure B.8 : Réseau de neurones mono-couche.

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 œuvre 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 où la suppression intervient lorsque le réseau converge.

B.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.

- 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.

B.4.2.1 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 :

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

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 :

$$W_{ij}(t+1) = W_{ij}(t) + y O_i O_j \quad (\text{B.4})$$

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

y : 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 Rosenblate 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 :

$$W_{ij}(t+1) = W_{ij}(t) + y x_i \quad \text{si la sortie actuelle est 0 et doit être égale à 1.} \quad (\text{B.5})$$

$$W_{ij}(t+1) = W_{ij}(t) - y x_i \quad \text{si la sortie actuelle est égale à 1} \quad (\text{B.6})$$

et doit être égale à 0.

$$W_{ij}(t+1) = W_{ij}(t) \quad \text{si la sortie est correcte.} \quad (\text{B.7})$$

Ou, $y > 0$ 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 :

$$E = \sum_{j=1}^n (d_j - y_j)^2 \quad (\text{B.8})$$

avec :

$$y = \sum_{i=1}^n x_i W_{ij} \quad (\text{B.9})$$

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

$$W_{ij}(t+1) = W_{ij}(t) + \eta x_i (d_j - y_j) \quad (\text{B.10})$$

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

η : 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 :

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

$y = (y_1; y_2; \dots; y_n)$ 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.

$$S_i = f\left(\sum_{i=1}^n W_{ji} x_i\right) \quad (\text{B.11})$$

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 :

$$E = \sum_{j=1}^n (y_j - s_j)^2 \quad (\text{B.12})$$

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 :

$$\begin{aligned} W_{ji}^s(t+1) &= W_{ji}^s(t) + \Delta W_{ji}^s(t) \\ \Delta W_{ji}^s(t) &= \gamma u_j^s o_i^{[s-1]} + r \Delta W_{ji}^s(t-1) \end{aligned} \quad (\text{B.13})$$

avec $\gamma > 0$, $0 \leq r \leq 1$

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

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

: le coefficient d'apprentissage

: 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.

B.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.

B.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....

B.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.

ANNEXE C

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 (triphase 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 :

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

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.

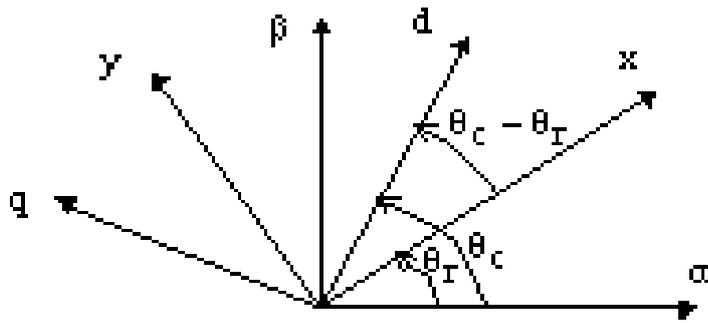


Figure C.1 : Représentation de la transformation dans le repère tournant

Donc, on peut définir une matrice de passage, du repère $(\alpha, \beta, 0)$ au repère $(d, q, 0)$, figure (C.1).

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

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

Soit, sous la forme condensée :

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

Où θ 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 :

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

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

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

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 :

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

Sous la forme condensée, nous avons :

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

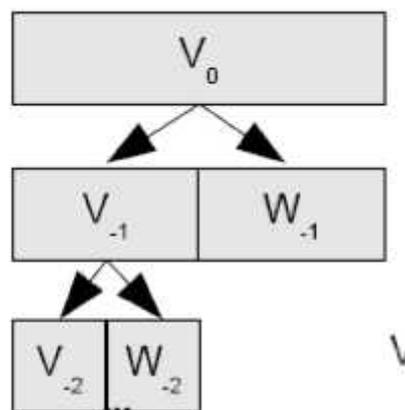
Nous remarquons que : $P^{-1} = P^t$, cette transformation est orthogonale.

Annexe D

LES COMMANDES MATLAB DES 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 multi résolution 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[65] .



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

ANNEXE E

DONNEES RELATIVES AUX MACHINES ASYNCHRONES TRIPHASEES 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 Ω |
| Résistance statorique par phase R_s | 5,793 Ω |
| 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 ² |
| La fréquence du réseau f | 50 Hz |

| | |
|-------------------------------------------------|--------------------------|
| Tension nominale V_n | 220/380 V |
| Nombre de paires de pôles p | 2 |
| Résistance rotorique par phase R_r | 2.658 Ω |
| Résistance statorique par phase R_s | 1.85 Ω |
| Inductance cyclique statorique L_s | 0,294 H |
| Inductance cyclique rotorique L_r | 0,2898H |
| Inductance mutuelle cyclique stator-rotor L_M | 0,2838 H |
| Moment d'inertie de la partie tournante J | 0,1284 Kg.m ² |
| La charge T_l | 5Nm |
| La fréquence du réseau f | 50 Hz |

REFERENCES

1. *Astrocytes, in science & vie.* 2005. p. 67.
2. Porto, A.B. and A. Pazos, *Neuroglial behaviour in computer science*, in *Artificial neural networks in real-life applications*. 2006, IGI Global. p. 1-21.
3. Antanitus, D.S., *A theory of cortical neuron-astrocyte interaction*. The Neuroscientist, 1998. **4**(3): p. 154-159.
4. Araque, A. and M. Navarrete, *Glial cells in neuronal network function*. Philosophical Transactions of the Royal Society of London B: Biological Sciences, 2010. **365**(1551): p. 2375-2381.
5. Morris, G.P., et al., *Microglia: a new frontier for synaptic plasticity, learning and memory, and neurodegenerative disease research*. Neurobiology of learning and memory, 2013. **105**: p. 40-53.
6. Zheng, D., et al., *Identification and control for singularly perturbed systems using multitime-scale neural networks*. IEEE transactions on neural networks and learning systems, 2017. **28**(2): p. 321-333.
7. Gandhi*, P.S. and F. Ghorbel, *High-speed precision tracking with harmonic drive systems using integral manifold control design*. International Journal of Control, 2005. **78**(2): p. 112-121.
8. Menon, P., et al., *Nonlinear flight test trajectory controllers for aircraft*. Journal of Guidance, Control, and Dynamics, 1987. **10**(1): p. 67-72.
9. Vakil, M., R. Fotouhi, and P.N. Nikiforuk, *End-effector trajectory tracking of a flexible link manipulator using integral manifold concept*. International Journal of Systems Science, 2011. **42**(12): p. 2057-2069.
10. Winkelman, J.R., et al., *Multi-time-scale analysis of a power system*. Automatica, 1980. **16**(1): p. 35-43.
11. Naidu, D., *Singular perturbations and time scales in control theory and applications: an overview*. Dynamics of Continuous Discrete and Impulsive Systems Series B, 2002. **9**: p. 233-278.
12. Zhang, Y., et al., *Singular perturbations and time scales in control theories and applications: an overview 2002–2012*. Int. J. Inf. Syst. Sci, 2014. **9**(1): p. 1-36.
13. Khosravi, M.A. and H.D. Taghirad, *Dynamic modeling and control of parallel robots with elastic cables: singular perturbation approach*. IEEE Transactions on Robotics, 2014. **30**(3): p. 694-704.
14. Yu, H. and B. Zhang. *Stabilizability of a class of singularly perturbed systems via switched output feedback*. in *Proceedings of 2013 Chinese Intelligent Automation Conference*. 2013. Springer.

15. Debnath, S., J. Qin, and M. Saeedifard, *Control and stability analysis of modular multilevel converter under low-frequency operation*. IEEE Transactions on Industrial Electronics, 2015. **62**(9): p. 5329-5339.
16. Fu, Z.-J., et al., *Nonlinear systems identification and control via dynamic multitime scales neural networks*. IEEE transactions on neural networks and learning systems, 2013. **24**(11): p. 1814-1823.
17. Fu, Z.-J., W.-F. Xie, and W.-D. Luo, *Robust on-line nonlinear systems identification using multilayer dynamic neural networks with two-time scales*. Neurocomputing, 2013. **113**: p. 16-26.
18. Fu, Z.-J., W.-F. Xie, and W.-D. Luo. *Induction motor identification using dynamic two-time scales neural networks with sliding mode learning*. in *Electrical & Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on*. 2012. IEEE.
19. Yu, W. and A.C. Sandoval, *Some new stability properties of dynamic neural networks with different time-scales*. International journal of neural systems, 2006. **16**(03): p. 191-199.
20. Li, X. and W. Yu, *Dynamic system identification via recurrent multilayer perceptrons*. Information sciences, 2002. **147**(1-4): p. 45-63.
21. Meyer-Bäse, A., R. Roberts, and S. Joshi. *Robust stability analysis of a class of noise perturbed two-time scale neural networks*. in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. 2006. IEEE.
22. Leung, C.-S. and L.-W. Chan, *Dual extended Kalman filtering in recurrent neural networks*. Neural Networks, 2003. **16**(2): p. 223-239.
23. Wang, X. and Y. Huang, *Convergence study in extended Kalman filter-based training of recurrent neural networks*. IEEE Transactions on Neural Networks, 2011. **22**(4): p. 588-600.
24. Sanchez, E.N., A.Y. Alanís, and A.G. Loukianov, *Discrete-time high order neural control: trained with Kalman filtering*. Vol. 112. 2008: Springer Science & Business Media.
25. Ramirez, A.F. and C. Aviles-Cruz, *Nonlinear system identification with a feedforward neural network and an optimal bounded ellipsoid algorithm*. 2008.
26. Yu, W. and J. de Jesús Rubio, *Recurrent neural networks training with stable bounding ellipsoid algorithm*. IEEE Transactions on Neural Networks, 2009. **20**(6): p. 983.
27. Zheng, D., W.-f. Xie, and S.-L. Dai. *Identification of singularly perturbed nonlinear system using recurrent high-order neural network*. in *Intelligent Control and Automation (WCICA), 2014 11th World Congress on*. 2014. IEEE.
28. Zheng, D.-D., W.-F. Xie, and C. Luo. *Robust identification for singularly perturbed nonlinear systems using multi-time-scale dynamic neural network*. in *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. 2017. IEEE.

29. Fu, Z., *Identification and Trajectory Tracking Control of Nonlinear Singularly Perturbed Systems*.
30. Kokotovic, P., H.K. Khali, and J. O'reilly, *Singular perturbation methods in control: analysis and design*. Vol. 25. 1999: Siam.
31. Dauphin-Tanguy, G., *Sur la représentation multi-modèle des systèmes singulièrement perturbés*. Dr Sc. Dissertation, Lille, France, 1983.
32. Chow, J. and P. Kokotovic, *A decomposition of near-optimum regulators for systems with slow and fast modes*. IEEE Transactions on Automatic Control, 1976. **21**(5): p. 701-705.
33. Ansary, A. and J. De Abreu-Garcia. *A simple algorithm for time scale separation*. in *Industrial Electronics Society, 1989. IECON'89., 15th Annual Conference of IEEE*. 1989. IEEE.
34. Guesbaoui, H. and C. lung. *Multi-time scale modelling in electrical machines*. in *Systems, Man and Cybernetics, 1993.'Systems Engineering in the Service of Humans', Conference Proceedings., International Conference on*. 1993. IEEE.
35. Touhami, O., et al. *Dynamics separation of induction machine models using gerschgorin's circles and singular perturbations*. in *Electrical and Electronics Engineering, 2004.(ICEEE). 1st International Conference on*. 2004. IEEE.
36. Hote, Y.V. and A.N. Jha, *New approach of Gerschgorin theorem in model order reduction*. International Journal of Modelling and Simulation, 2015. **35**(3-4): p. 143-149.
37. Borne, P. and J.-P. Richard, *Modélisation et identification des processus*. Vol. 1. 1992: Editions Technip.
38. Kokotovi , P., H.K. Khalil, and J. O'reilly, *Singular perturbation methods in control: analysis and design*. 1999: SIAM.
39. Dauphin-Tanguy, G., *Sur la représentation multi-modèle des systèmes singulièrement perturbés: application à l'analyse et à la synthèse*. 1983.
40. Perea, G. and A. Araque, *Astrocytes potentiate transmitter release at single hippocampal synapses*. Science, 2007. **317**(5841): p. 1083-1086.
41. Perea, G., M. Navarrete, and A. Araque, *Tripartite synapses: astrocytes process and control synaptic information*. Trends in neurosciences, 2009. **32**(8): p. 421-431.
42. Ikuta, C., Y. Uwate, and Y. Nishio. *Chaos glial network connected to multi-layer perceptron for solving two-spiral problem*. in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*. 2010. IEEE.
43. Ikuta, C., Y. Uwate, and Y. Nishio. *Performance and features of multi-layer perceptron with impulse glial network*. in *Neural Networks (IJCNN), The 2011 International Joint Conference on*. 2011. IEEE.
44. Ikuta, C., et al., *Multi-layer perceptron with pulse glial chain*. IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, 2016. **99**(3): p. 742-755.

45. Ikuta, C., Y. Uwate, and Y. Nishio. *Multi-Layer perceptron with pulse glial chain for solving two-spiral problem*. in *IEEE Workshop on Nonlinear Circuit Networks*. 2011. Citeseer.
46. Ikuta, C., Y. Uwate, and Y. Nishio. *Multi-layer perceptron with positive and negative pulse glial chain for solving two-spirals problem*. in *Neural Networks (IJCNN), The 2012 International Joint Conference on*. 2012. IEEE.
47. Porto-Pazos, A.B., et al., *Artificial astrocytes improve neural network performance*. PloS one, 2011. **6**(4): p. e19109.
48. Alvarillos-González, A., A. Pazos, and A.B. Porto-Pazos, *Computational models of neuron-astrocyte interactions lead to improved efficacy in the performance of neural networks*. Computational and mathematical methods in medicine, 2012. **2012**.
49. Landolsi, H. and K. Marzouki. *Self organizing neuro-glial network, SONG-Net*. in *Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB), 2014 IEEE Symposium on*. 2014. IEEE.
50. Schmidt, A. and Z. Bandar. *Modularity, a concept for new neural network architectures*. in *Proc. IASTED International Conf. Computer Systems and Applications*. 1998.
51. Happel, B.L. and J.M. Murre, *Design and evolution of modular neural network architectures*. Neural networks, 1994. **7**(6): p. 985-1004.
52. Ding, Y., et al., *A modular neural network architecture with concept*. Neurocomputing, 2014. **125**: p. 3-6.
53. Melin, P., et al., *A hybrid modular neural network architecture with fuzzy Sugeno integration for time series forecasting*. Applied Soft Computing, 2007. **7**(4): p. 1217-1226.
54. Chen, K. and H. Chi, *A modular neural network architecture for pattern classification based on different feature sets*. international journal of neural systems, 1999. **9**(06): p. 563-581.
55. Pillutla, S. and A. Keyhani, *Power system stabilization based on modular neural network architecture*. International Journal of Electrical Power & Energy Systems, 1997. **19**(6): p. 411-418.
56. Sharma, S., et al., *Learning soft computing control strategies in a modular neural network architecture*. Engineering Applications of Artificial Intelligence, 2003. **16**(5-6): p. 395-405.
57. Buessler, J., et al. *Multiple self-organizing maps to facilitate the learning of visuo-motor correlations*. in *Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on*. 1999. IEEE.
58. Caelli, T., L. Guan, and W. Wen, *Modularity in neural computing*. Proceedings of the IEEE, 1999. **87**(9): p. 1497-1518.
59. Jacobs, R.A., et al., *Adaptive mixtures of local experts*. Neural computation, 1991. **3**(1): p. 79-87.

60. Ronco, E., H. Gollee, and P.J. Gawthrop, *Modular neural network and self-decomposition*. Connection Science (special issue: COMBINING NEURAL NETS).(To appear), 1996.
61. Bennani, Y., *A modular and hybrid connectionist system for speaker identification*. Neural Computation, 1995. **7**(4): p. 791-798.
62. Kirschning, I., H. Tomabechi, and J.-I. Aoe. *A parallel recurrent cascade-correlation neural network with natural connectionist glue*. in *Neural Networks, 1995. Proceedings., IEEE International Conference on*. 1995. IEEE.
63. Perrone, M.P. and L.N. Cooper, *When networks disagree: Ensemble methods for hybrid neural networks*. 1992, BROWN UNIV PROVIDENCE RI INST FOR BRAIN AND NEURAL SYSTEMS.
64. Jain, A.K., J. Mao, and K.M. Mohiuddin, *Artificial neural networks: A tutorial*. Computer, 1996. **29**(3): p. 31-44.
65. LANDRÉ, J., *Boîte à outils Ondelettes sous Matlab*.

