

Titre: Entraînement de réseaux de neurones récurrents à pulses appliqué à la modélisation d'un tissu neuronal biologique
Title: [Entraînement de réseaux de neurones récurrents à pulses appliqué à la modélisation d'un tissu neuronal biologique](#)

Auteur: Matthieu Gilson
Author: [Matthieu Gilson](#)

Date: 2003

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Gilson, M. (2003). Entraînement de réseaux de neurones récurrents à pulses appliqué à la modélisation d'un tissu neuronal biologique [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/8694/>

Document en libre accès dans PolyPublie Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/8694/>
PolyPublie URL: <https://publications.polymtl.ca/8694/>

Directeurs de recherche: Jean-Jules Brault
Advisors: [Jean-Jules Brault](#)

Programme: Non spécifié
Program: [Non spécifié](#)

UNIVERSITÉ DE MONTRÉAL

ENTRAÎNEMENT DE RÉSEAUX DE NEURONES
RÉCURRENTS À PULSES APPLIQUÉ À LA
MODÉLISATION D'UN TISSU NEURONAL BIOLOGIQUE

MATTHIEU GILSON

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE ÉLECTRIQUE)

NOVEMBRE 2003



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-91945-5

Our file Notre référence

ISBN: 0-612-91945-5

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this dissertation.

While these forms may be included in the document page count, their removal does not represent any loss of content from the dissertation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de ce manuscrit.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

ENTRAÎNEMENT DE RÉSEAUX DE NEURONES RÉCURRENTS
À PULSES APPLIQUÉ À LA MODÉLISATION D'UN TISSU
NEURONAL BIOLOGIQUE

présenté par : GILSON Matthieu

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment acceptée par le jury d'examen constitué de :

Mme. CHERIET Farida, Ph. D., présidente

M. LEPORÉ Franco, Ph. D., membre

M. BRAULT Jean-Jules, Ph. D., directeur

M. SAWAN Mohamad, Ph. D., co-directeur

REMERCIEMENTS

Je tiens à remercier tous ceux qui m'ont côtoyé durant ces deux ans à Montréal, tant sur le plan des études que dans la vie courante.

Merci en particulier à mes directeurs de recherche Jean-Jules et Mohamad, avec qui j'ai pris goût à la recherche, et à R. Gulrajani pour sa disponibilité.

Et merci entre autres à CIBL (la radio libre de Montréal), au Sergent Recruteur, au P'tit Bar, au Ludik et à In Da Jungle pour l'animation de la vie culturelle.

RÉSUMÉ

Les récents progrès en microélectronique permettent d'entrevoir la restauration de fonctions corporelles en excitant le système nerveux grâce à des micro-stimulateurs implantables dans le corps humain, notamment recréer une vision partielle chez des personnes devenues aveugles. Les travaux de Foerster, Dobelle et Hambrecht ont montré que la stimulation électrique du cortex visuel engendrait l'apparition de phosphènes (sensations mentales d'un point lumineux) dans le champ visuel et ont étudié l'impact de certains paramètres de stimulation sur les qualités visuelles des phosphènes ainsi générés (intensité, stabilité, etc.). Ces expériences ont été effectuées chez des personnes ayant perdu la vue (leurs voies visuelles ont déjà été organisées et fonctionnelles au niveau du cortex, ce qui n'est pas le cas chez des personnes aveugles de naissance). Le laboratoire de micro-électronique PolySTIM développe une prothèse visuelle implantable dans la couche 4 du cortex visuel primaire qui comportera une matrice d'électrodes permettant de stimuler le tissu neuronal et d'enregistrer son activité en de nombreux sites. Elle sera suffisamment versatile pour envoyer une grande diversité de signaux électriques aux cellules corticales.

On ne peut pas reproduire l'information visuelle afférente via le nerf optique, mais on veut exploiter les circuits neuronaux existants dans un cortex structuré. Jusqu'à présent, l'induction d'autres formes géométriques que des points n'a pas été maîtrisée, de même que le mouvement ou la couleur. Il est nécessaire de contrôler ces composantes visuelles : restaurer une vision partielle ne se limite pas à reproduire la version pixellisée d'une scène visuelle. On est donc amené à déterminer des protocoles de stimulation adéquats (quels types de signaux électriques choisir et comment les combiner au niveau des nombreuses électrodes?), de sorte que les aires visuelles supérieures « interprètent » une scène conformément à ce que l'on veut représenter dans le champ visuel restauré. Dans un premier temps, on utilisera comme base des protocoles de stimulation des trains

de pulses, dont les paramètres (amplitude et fréquences des pulses, durée des trains, délai inter-trains, etc.) seront optimisés.

Le nombre de paramètres à ajuster est immense même pour une scène très simple et il faut déterminer lesquels sont significatifs, ce qui ne peut être fait uniquement de manière expérimentale. On désire donc construire un réseau neuronal à pulses (symboles des potentiels d'action) de style biologique modélisant le tissu cortical (on exclue les autres cellules du tissu qui ne participent pas à priori au traitement de l'information visuelle). Un tel modèle doit valider 4 objectifs :

- son **architecture** s'inspire de la **physiologie** (réseau massivement récurrent, densités statistiques de neurones et de connexions, etc.),
- son **comportement** doit être **globalement réaliste**, par exemple son activité de repos (les paramètres synaptiques doivent se situer dans certaines domaines de variations),
- il doit **s'auto-structurer** en fonction des signaux reçus, copiant l'organisation du cortex en colonnes grâce aux connexions internes à la couche 4 (Hubel et Wiesel),
- enfin, il doit être **fonctionnel**, au sens où certains neurones doivent accomplir des **tâches spécifiques**, comme reconnaître certains motifs parmi les signaux reçus afin de tester des protocoles de stimulation associés à des scènes visuelles par exemple.

Les deux derniers points nécessitent d'optimiser les paramètres synaptiques du réseau.

Nous étudions dans ce mémoire la faisabilité de construire un tel réseau neuronal et comment l'optimiser, ce qui constitue un problème non classique du fait des contraintes liées à la physiologie du tissu réel. Par exemple, l'entraînement classique d'un réseau de neurones sigmoïdaux orienté pour approcher une fonction mathématique par rétro-propagation de l'erreur va de paire avec la recherche du réseau minimal pour limiter le nombre de fonctions générées et ainsi éviter le sur-apprentissage. Dans le cas de réseaux de style biologique on est constraint de gérer le très grand nombre de signaux générés, et il faut en plus découvrir parmi ces signaux des informations pertinentes relatives à la représentation d'une scène visuelle.

On étudie une première méthode basée sur la connaissance de l'activité du réseau en des neurones de sortie, en réponse à des stimulations aux paramètres connus en des

neurones d'entrée (modélisant une électrode). On entraîne le réseau à copier des activités-objectifs (supposées réalistes) pour un ensemble d'entraînement (stimuli en entrée). On définit une mesure sur les trains de pulses pour évaluer une fonction d'erreur entre les activités en sortie des réseaux entraîné et objectif. Idéalement, cette mesure doit être cohérente avec le codage de l'information visuelle par les pulses, qui n'est que très partiellement connu pour le moment. On étudie donc deux cas à priori : une mesure qui correspond à des densités d'activité avec un lissage temporel important, et une autre qui respecte les positions des pulses prises individuellement dans les trains. Pour cette étude, on se limite à de petits réseaux et les objectifs correspondent aux sorties d'un réseau de même architecture avec des paramètres synaptiques fixés (une solution existe). Un **algorithme génétique** recherche des paramètres synaptiques du réseau entraîné afin de trouver le minimum de la fonction d'erreur. En particulier, on voulait contraindre le comportement global (i.e. les activités de tous les noeuds et pas seulement ceux de sorties) du réseau entraîné à converger vers celui du réseau-objectif.

L'entraînement est plus facile dans le cas d'un lissage temporel important. Les comportements topographiques d'un réseau correctement entraîné et du réseau-objectif peuvent être très différents, même pour de petits réseaux avec un large ensemble d'entraînement. L'analyse de l'erreur en fonction des paramètres synaptiques illustre la difficulté d'optimiser de tels réseaux. Avec cette méthode, on pourrait entraîner un réseau à modéliser une colonne corticale type et juxtaposer ces réseaux pour construire un tissu artificiel en optimisant les connexions inter-colonnes. Mais cela suppose de connaître les influences d'une colonne sur ses voisines au niveau des trains de pulses, ce qui n'est pas le cas pour l'instant.

La deuxième méthode réunit deux concepts complémentaires pour construire un modèle satisfaisant aux objectifs fixés :

- les **cartes topographiques auto-organisatrices** (Kohonen), i.e. un réseau effectuant une spécialisation non supervisée en fonction des signaux en entrée, à la manière de l'auto-adaptation du cortex visuel;

- les machines à états liquides (Maass) sont des systèmes dynamiques astables dont on peut exploiter la puissance computationnelle en entraînant un perceptron ou un neurone en sortie grâce à une règle d'apprentissage dite « p-delta »; sous certaines conditions sur les connexions synaptiques dynamiques, un réseau neuronal récurrent à pulses de grande taille en est une bonne approximation.

On propose d'intégrer ces deux concepts dans un réseau d'architecture biologiquement réaliste et d'entraîner des neurones de sortie à reconnaître certains motifs de stimulation particuliers. Ainsi, on obtient un modèle auto-adaptatif de la couche corticale 4 avec des neurones se projetant hors de cette couche, à l'instar de neurones pyramidaux.

Pour valider notre approche, on simule un influx optique en connectant le réseau à un modèle simpliste de rétine avec des récepteurs réagissant à des barres lumineuses mobiles orientées. Les résultats montrent que ces deux concepts peuvent être complémentaires et on parvient notamment à discriminer les déplacements verticaux de ceux horizontaux, ou encore des déplacements selon des directions opposées. Notre réseau constitue une base souple pour construire un modèle de tissu cortical et il pourra être raffiné pour rendre compte de l'organisation en colonnes corticales lorsque l'on déplace des barres d'orientation aléatoire ou encore inclure d'autres composantes visuelles (rotations, etc.).

La première méthode (algorithme génétique) reste dépendante de la connaissance du codage de l'information visuelle pour définir une métrique et pouvoir évaluer les sorties du réseau conformément aux scènes représentées dans leurs activités. Le second modèle s'affranchit de la connaissance de ce codage et il peut être amélioré avec des neurones et des connexions synaptiques de plus en plus réalistes (modèle de Hodgkin-Huxley), et on peut aussi l'entraîner afin d'intégrer progressivement d'autres fonctionnalités du cortex visuel et tester des protocoles de stimulation. Il reste également à définir un modèle d'électrode pour simuler les réponses du tissu biologique aux stimulations de la matrice d'électrodes.

ABSTRACT

Recent breakthroughs in microelectronics allow for the design of micro-stimulators aimed at recovering diseased body functions, by stimulating the nervous system via small implants within the human body. Particularly, partial vision may be restored for the blind according to Foerster, Dobelle and Hambrecht, who showed that cortical electrical stimulation triggers phosphenes (visual sensation of a luminous dot), and who studied the impact of stimulation parameters on the visual abilities of the created phosphenes (intensity, stability, etc.). Such experiments were run with people who were born with, but who later lost the ability to see, i.e. whose visual system had been organized and functional. Born blind people don't have the same organization of the visual cortex, which is a problem for recovering vision. The PolySTIM microelectronics laboratory is developing a visual prosthesis to be implanted in the 4th layer of the primary visual cortex, which will allow stimulation of the neural tissue and the recording of its activity at many locations, thanks to an electrode array. It will be versatile enough to send a huge diversity of electrical signals towards the cortical cells.

If visual information cannot be reproduced at the cortical level, we instead look to exploit the existing paths structured in the visual cortex. Until now, inducing other geometric shapes than dots, such as motion and color, has not been mastered. Such visual features are necessary to recover a « real » partial vision, which reproduces more than a simply pixilated version of the visual scene in front of the subject. Thus, we shall define stimulation protocols suitable for allowing the superior visual areas to interpret scenes according to what we want to represent. A stimulation protocol describes for instance the kinds of electrical signals to use and the ways to combine them with the pool of electrodes. First, we consider using pulse trains as a basis of the electrical signals to be sent to the neural cells, and we want to optimize the parameters of such pulse trains (frequency and amplitude of the pulses, duration of the trains and delays between trains).

The number of such parameters to be set is enormous even to represent a simple visual scene, so we shall determine which ones are significant, something which cannot be determined through experiments alone. That is why we have to design a model of the cortical tissue to forecast its behavior. We choose to restrict the modeling to the neurons (assuming that other cells don't take part in the information processing) and to build a « biological-like » neural network that deals with pulses (symbolizing the action potentials). Such a model must respect 4 objectives :

- its **architecture** is to be inspired by cortical **physiology** (massively recurrent, with realistic neuron and synapse densities, etc.),
- its **behavior** must be **globally realistic**, for example the increase of activity between resting and during a stimulation (the synaptic parameters are set in some precise intervals),
- it has to **structure itself** to the inputs, according to the column organization due to the lateral connectivity with the 4th layer as described by Hubel and Wiesel,
- finally, it must be **functional**, i.e. some neurons have to accomplish specific tasks, such as pattern recognition, in addition to testing stimulation protocols related to visual scenes.

The two last points require the optimization of the synaptic parameters of the network by means of training.

We study in this thesis the possibility of building such a neural network and optimizing it, which is a non-classical learning problem due to the constraints linked to physiology. For instance, classical training of a neural network for approximating mathematical functions using back-propagation goes hand in hand with seeking the minimal architecture of the network, in order to avoiding over-fitting. In our case, we have to deal with the huge number of functions generated within the network (due to the number of cells taken into account), and we have to discover within all these signals pertinent information related to the representation of a visual scene.

First, we study a method based of the knowledge of the network activity at defined output neurons, in response to stimulations in some other sites (we know the

parameters of the signals sent to the network). We train a network to copy the activities at the outputs (supposedly realistic) for a training pool of stimuli (the sent signals). A metric is defined on the pulse trains, which enables to compare the outputs of the trained network and the objective activity, by means of an error function. Ideally, this measure should be consistent with the information encoded at the cortical level. Since this encoding is still hardly known, we study two distinct cases *a priori* : one measure corresponding to the activity density of the neurons and one that respects the individual positions of the pulses within the temporal sequences. A **genetic algorithm** is used to determine synaptic parameters of the trained network suitable for minimizing the error function. We restrain our study to small networks and the objective activity consists of the outputs for the same architecture of the network with particular synaptic parameters, which assures the existence of a solution. We wanted especially to force the global behavior (including the activity of all its nodes and not only the outputs) of the trained network to converge towards the global activity of the objective network.

The training is easier in the case of an important temporal averaging. Even for a well-trained small network and many stimuli, the global behavior can be very different from the objective one. The analysis of the error function shows that the training of our networks is a difficult problem. Using this method, we could train small networks in order to copy a cortical column and then rebuild the neural tissue with a network of such columns, while adjusting the connections between the columns. Yet, we would have to know the influence of each column on its neighbors, what is still only partially known.

The second method combines two complementary concepts to build a model able to satisfy our four objectives :

- the **self-organizing maps** of Kohonen, which consists of a network specializing in a non supervised way depending on the inputs, and which was inspired by auto-adaptation in the biological cortex,
- the **liquid-state machines** (Maass), which are non-stable dynamic systems the power of which can be exploited by training outputs (a perceptron or a neuron) thanks to the

« p-delta » rule; under certain conditions on its synaptic connections, a large enough recurrent pulsed neural network is a good approximation of a liquid-state machine.

We propose to combine these two concepts in a biological-like neural network, and to train output neurons to recognize particular stimulation patterns. Thus we obtain a self-adapting model of the cortical 4th layer, with lateral connectivity and output neurons representing the pyramidal cells.

We simulate an optical influx while connecting our network to a receptor field (a mere model of the biological retina), and we present moving luminous bars in the visual field. Our results show that the two concepts can be complementary and we achieved discrimination between vertical and horizontal motions, or between opposite motions. Our network is a suitable basis for building a model of the cortical tissue, and it can be improved to show the organization in columns, such as its biological counterpart when presenting randomly oriented bars or to take into account other visual features (rotations, etc.).

The first method (genetic algorithm) is dependent on the knowledge of the encoding of the visual information at the cortical level, in order to define a suitable metric to gauge the outputs of a network regarding the representation of a visual scene. The second model doesn't need such knowledge and it can be improved in terms of biological realism by using refined neurons (Hodgkin-Huxley model) and synapses, and its training enables it to take many other functionalities of the real cortex, so that we can test stimulation protocols with it. For our global problem, we have to develop an electrode model able to simulate both the implant and the neural tissue.

TABLE DES MATIÈRES

Remerciements.....	iv
Résumé.....	v
Abstract.....	ix
Table des matières.....	xiii
Plan.....	xvi
Liste des figures.....	xix
Liste des tableaux.....	xxiii
Liste des équations.....	xxiv
Liste des abréviations.....	xxv
1 Introduction.....	1
1.1 Les prothèses visuelles.....	1
1.2 Le projet d'implant visuel cortical du laboratoire PolySTIM.....	6
1.3 Simuler le tissu neuronal pour trouver des paramètres de stimulation adéquats.....	11

2 Modélisation du tissu neuronal de la couche 4 du cortex visuel primaire.....	14
2.1 Description physiologique du tissu biologique.....	14
2.1.1 Physiologie et fonctionnement du neurone.....	14
2.1.2 Connexions des neurones.....	20
2.1.3 Stimulation et lecture de l'activité dans un tissu neuronal.....	22
2.1.4 organisation du cortex visuel primaire.....	25
2.1.5 Codage de l'information visuelle par les potentiels d'action....	30
2.2 Les modèles neuronaux HH, I&E et sigmoïdaux.....	32
2.2.1 Le modèle de Hodgkin-Huxley.....	32
2.2.2 Les neurones à pulses I&E.....	35
2.2.3 Les neurones « sigmoïdaux ».....	37
2.3 Les réseaux neuronaux récurrents à pulses (RNRP).....	38
2.4 Récapitulatif des objectifs du modèle.....	40
2.5 Le logiciel NEURON.....	41
3 Algorithmes d'apprentissage appliqués aux RNRP.....	44
3.1 Algorithmes classiques d'entraînement de réseaux neuronaux.....	44
3.1.1 La rétro-propagation pour les RNPA.....	44
3.1.2 Les réseaux de Hopfield et l'apprentissage de Hebb.....	47
3.1.3 Spécificités de l'entraînement des RNRP « biologiques ».....	49
3.2 Entraîner des RNRP grâce à des méthodes générales d'optimisation.....	50
3.3 Apprentissage non supervisé : les cartes topographiques auto-organisatrices.....	53

4 Entraînement de RNRP par algorithmes génétiques.....	57
4.1 Description du réseau à entraîner.....	57
4.1.1 But de l'entraînement et architecture du réseau.....	57
4.1.2 Calcul de la fonction d'erreur utilisée pour l'optimisation.....	58
4.2 Description de l'algorithme génétique utilisé.....	62
4.3 Protocole de simulation.....	67
4.4 Résultats.....	68
4.4.1 Entraînement des nœuds de sortie.....	68
4.4.2 Comportement global du réseau.....	73
4.4.3 Analyse des fonctions d'erreur.....	75
4.5 Discussion.....	78
5 Entraînement de cartes topographiques à états liquides.....	81
5.1 Les cartes topographiques auto-organisatrices relatives aux signaux visuels.....	81
5.2 Les machines à états liquides.....	85
5.2.1 Définition du concept d'états liquides.....	85
5.2.2 La règle p-delta.....	89
5.3 Protocole de simulation.....	93
5.4 Résultats.....	98
5.4.1 Comportement général du réseau.....	98
5.4.2 Puissance du RNRP en tant que MEL.....	99
5.4.3 Impact de l'auto-organisation.....	101
5.5 Discussion.....	105
6 Conclusion et travaux futurs.....	108
Bibliographie.....	110

PLAN

Ce mémoire étudie des méthodes d'ajustement de paramètres synaptiques dans des réseaux neuronaux à pulses au moyen d'algorithmes d'apprentissage, dans le but de construire un modèle de tissu neuronal biologique. Ce problème d'optimisation n'est pas classique dans la mesure où entre autres contraintes l'architecture de tels réseaux doit s'inspirer de la réalité physiologique, tandis que généralement la philosophie des problèmes d'optimisation utilisant des réseaux de neurones consiste à construire une architecture adaptée au problème. Notamment, on va devoir tirer parti du grand nombre de neurones dans le réseau et autant de fonctions générées (trains de pulses). De plus, le modèle ne doit pas seulement respecter des critères physiologiques (activité au repos, extensions temporelles et spatiales de l'activité moyenne suite à des stimulations électriques, etc.), il doit également rendre compte de l'auto-organisation corticale décrite par Hubel et Wiesel et être fonctionnel, i.e. comporter des neurones capables d'effectuer des tâches spécifiques (par exemple reconnaître certains motifs de stimulation en vue de tester n'importe quel protocole de stimulation).

On présente en introduction les raisons qui ont motivé ces travaux : on donne d'abord un aperçu général des prothèses visuelles dont celle développée par l'équipe de Dobelle qui est la seule réalisée à ce jour (1.1), puis on décrit le projet de l'équipe PolySTIM de stimulateur implantable dans la couche 4 du cortex visuel (1.2). On montre ensuite la nécessité de modéliser le tissu cortical afin de déterminer des protocoles de stimulation du système visuel adéquats pour recréer une « véritable » vision partielle (1.3).

On veut construire un tel modèle avec des réseaux neuronaux « biologiquement réalistes ». Pour définir ce que l'on entend par « réalisme biologique » et justifier nos choix de modélisation, on commence par décrire la physiologie du tissu cortical visuel (2.1), depuis le niveau du neurone (2.1.1) avec les connexions synaptiques (2.1.2) et les

manières de le stimuler ou d'enregistrer son activité (2.1.3), jusqu'à l'organisation spécifique du cortex visuel en colonnes découverte par les prix Nobel de médecine de 1981 Hubel et Wiesel (2.1.4), et le codage des informations visuelles dans les trains de pulses. On passe en revue différents modèles de neurones artificiels susceptibles de convenir pour construire le modèle : celui de Hodgkin-Huxley (2.2.1), les neurones Intégrer-&-Émettre (2.2.2) et les neurones sigmoïdaux (2.2.3); puis on présente quelques caractéristiques du type de réseau que l'on va utiliser et entraîner : les réseaux neuronaux récurrents à pulses (2.3); ainsi que les caractéristiques physiologiques que l'on va prendre en compte et les limites de raffinement du modèle que l'on se fixe à priori (2.4). Enfin, on présente sommairement le logiciel libre NEURON utilisé pour les simulations (2.5).

Une fois construite une architecture générale de réseau inspirée de la biologie, on va optimiser les paramètres des connexions synaptiques afin de donner un comportement « réaliste » à ces réseaux (par exemple conformes à des expériences sur des cobayes). D'abord, on présente les algorithmes d'apprentissage les plus classiques qui sont spécifiques à des réseaux de neurones artificiels particuliers, puis on expose deux méthodes que l'on va appliquer pour notre problème d'optimisation :

- la rétro-propagation pour les réseaux neuronaux sigmoïdaux à propagation avant permettant d'approximer n'importe quelle fonction mathématique (3.1.1),
- les réseaux associatifs développés pour certaines tâches particulières comme les ceux de Hopfield ou de Hebb (3.1.2),
- la méthode génétique en tant que méthode d'optimisation générale pour ajuster individuellement les paramètres synaptiques dans un réseau de neurones où l'on définit des neurones d'entrée et de sortie (3.2),
- les cartes topographiques auto-organisatrices définies par Kohonen comme modèle de l'organisation du cortex visuel en colonnes et une application aux réseaux de neurones à pulses (3.3).

Le chapitre 4 étudie l'application de la méthode génétique pour entraîner des réseaux neuronaux récurrents à pulses dans lequel sont définis des nœuds d'entrée et de sortie. Le but d'une telle méthode est de faire converger le comportement global du

réseau à partir de l'optimisation en quelques points seulement. On décrit l'architecture des réseaux utilisés et on définit une mesure de l'activité au niveau des noeuds de sortie qui vont intervenir dans l'optimisation (4.1), on présente l'application de l'algorithme génétique à l'ajustement des poids (4.2) et le protocole de simulation (4.3). On évalue ensuite les résultats (4.4) et on discute enfin de la pertinence de cette méthode lorsque le réseau modélisant le tissu biologique comporte un très grand nombre de neurones (4.5).

Dans le chapitre 5, on présente deux concepts : les cartes topographiques auto-organisatrices (5.1) et les machines à état liquide (5.2). La combinaison de ces deux concepts dans un réseau neuronal récurrent à pulses vise à entraîner certains neurones à effectuer des tâches spécifiques grâce au second concept dans un réseau auto-adaptatif en fonction de ses entrées. Mieux, on peut évaluer dans quelle mesure ils peuvent être complémentaires. On décrit le protocole de simulation (5.3), on analyse les résultats (5.4) et enfin on discute cette méthode et d'autres applications éventuelles de ce réseau puissant du point de vue computationnel (5.5).

Le chapitre 6 conclue le mémoire et synthétise les avancées dans la modélisation grâce à chaque méthode et propose plusieurs méthodes pour raffiner le modèle de tissu neuronal biologique. On y aborde également les prochaines étapes pour construire un modèle du tissu de la couche corticale qui nous intéresse avec la matrice d'électrodes implantée pour permettre de tester les protocoles de stimulation.

Les deux premières parties du chapitre 2 sont assez longues avec la présentation détaillée de la physiologie du cortex visuel et du modèle de Hodgkin-Huxley qui n'est pas utilisé dans ce mémoire, mais elles servent principalement à montrer la pertinence de nos choix dans la construction du modèle ainsi qu'à entrevoir des possibles raffinements du modèle. On peut se référer pour ce qui concerne la « réalité biologique » à la partie 2.4 qui synthétise les caractéristiques utiles que l'on va prendre en compte.

LISTE DES FIGURES

Figure 1.1.A : Vue de dessus du système visuel et détail du corps genouillé.....	2
Figure 1.1.B : Schéma du système visuel de l'œil au cortex visuel primaire.....	3
Figure 1.2 : Schéma du dispositif de l'équipe de Dobelle.....	5
Figure 1.3 : Schéma de la disposition des électrodes de l'implant cortical.....	8
Figure 1.4 : Reconstitution d'une scène en mode fil de fer.....	9
Figure 1.5 : Exemple de paramètres intervenant dans la stimulation.....	10
Figure 1.6 : Exemple de mesures biologiques de l'activité neuronale d'un tissu suite à une stimulation électrique.....	12
Figure 2.1 : Structure générale d'un neurone.....	15
Figure 2.2 : Différentes géométries de neurones.....	16
Figure 2.3.A : Schéma d'un canal ionique de la membrane d'un neurone.....	17
Figure 2.3.B : Canal ionique de la membrane d'un neurone.....	17
Figure 2.4.A : Potentiel d'action - description des phases des canaux ioniques.....	18
Figure 2.4.B : Potentiel d'action - propagation.....	19
Figure 2.5 : Fonctionnement d'une synapse biologique.....	20
Figure 2.6 : Exemples de comportement d'un modèle de synapse dynamique.....	21
Figure 2.7 : Simulation en courant par une électrode monopolaire.....	23
Figure 2.8 : Schéma des neurones pyramidaux et stellaires.....	26
Figure 2.9 : Organisation du cortex visuel primaire.....	28
Figure 2.10 : Courbe physiologique de la réponse d'un neurone en fonction de l'orientation présentée.....	29
Figure 2.11 : Carte topographique des colonnes corticales.....	30
Figure 2.12 : Circuit équivalent et formule principale du modèle de Hodgkin-Huxley.....	33

Figure 2.13 : Évolution du potentiel transmembranaire d'une section de câble selon le modèle HH lors de stimulations intracellulaires en courant.....	34
Figure 2.14 : Schéma fonctionnel d'un neurone I&E.....	35
Figure 2.15 : Exemple de fonctionnement d'un neurone I&E.....	36
Figure 2.16 : Description d'un neurone « sigmoïdal ».....	37
Figure 2.17 : Exemple de reconstruction d'une structure 3-D pour un neurone pyramidal du cortex visuel primaire chez l'homme.....	42
Figure 3.1 : Architecture d'un RNPA à une seule couche cachée.....	45
Figure 3.2 : Architecture d'un réseau de Hopfield.....	47
Figures 3.3 : Entraînement d'un RNRP en définissant des neurones de sortie.....	50
Figure 3.4 : Modèle du cortex visuel par juxtaposition de colonnes corticales.....	53
Figure 3.5 : Exemple de CTAO.....	54
Figure 3.6 : Fonction-chapeau de pondération des mises à jour dans la CTAO.....	56
Figure 4.1 : Architecture d'un RNRP avec des entrées et sorties.....	58
Figure 4.2 : Calcul de l'erreur entre deux trains de pulses.....	60
Figure 4.3 : Variations de l'erreur lors d'une translation temporelle.....	61
Figure 4.4 : Niveaux d'organisation des paramètres pour l'algorithme génétique...	63
Figure 4.5 : Évaluation d'un individu par rapport à un ensemble d'entraînement...	64
Figure 4.6.A : Détail d'une mutation.....	65
Figure 4.6.B : Détail de la création par reproduction.....	65
Figure 4.6.C : Évolution de la population durant une génération.....	66
Figure 4.7 : Architecture du réseau R4.....	67
Figure 4.8 : Influence de la taille de l'ensemble d'entraînement dans le cas du lissage grossier.....	70
Figures 4.9 : Influence de la taille de l'ensemble d'entraînement dans le cas du lissage grossier.....	71
Figure 4.10 : Répartition des erreurs d'entraînement et de validation pour R10 entraîné avec 400 stimuli dans le cas du lissage grossier.....	72

Figure 4.11 : Comparaison des activités des deux nœuds internes dans R4 entre le réseau entraîné et le réseau-objectif.....	73
Figure 4.12 : Comparaison topographique des activités résultat et objectif.....	74
Figures 4.13 : Variations locales de l'erreur en fonction des paramètres synaptique d'une connexion de R4 avec le lissage grossier.....	75-77
Figures 4.14 : Variations locales de l'erreur cumulée pour 1 stimulus.....	78
Figure 5.1 : Architecture du modèle RF-LISSOM.....	82
Figure 5.2 : Carte topographique corticale obtenue par RF-LISSOM.....	83
Figure 5.3 : Fonction-noyau intervenant dans la formule d'auto-organisation.....	84
Figure 5.4 : Schéma général d'une MEL à une entrée et une sortie.....	86
Figure 5.5 : Illustration des états liquides d'un RNRP.....	87
Figure 5.6 : Influence des méta-paramètres d'entraînement de la règle p-delta.....	90
Figure 5.7 : Influence des entrées sur l'entraînement d'un neurone à pulses.....	91
Figure 5.8 : Classes des fonctions-objectifs que l'on peut approcher en sortie d'un neurone à pulses.....	92
Figure 5.9 : Graphes des distributions des probabilités de connexion.....	94
Figure 5.10 : Exemple d'architecture du RNRP.....	94
Figure 5.11 : Génération des entrées du RNRP par des cellules réceptrices.....	97
Figure 5.12 : Variation de l'activité globale du RNRP lors d'une stimulation.....	98
Figure 5.13.A : Comparaison des activités générées par des stimuli.....	99
Figure 5.13.B : Évolution de la distance calculée sur le RNRP en fonction de la distance entre les stimuli.....	100
Figure 5.14 : Entraînement d'un neurone d'extraction pour discriminer des barres horizontales et verticales.....	101
Figure 5.15 : Effet de la répétition d'un stimulus sur les connexions efférentes d'un neurone.....	102
Figure 5.16 : Topographie de l'activité du RNRP suite à son auto-organisation....	103
Figures 5.17 : Évolution de la distance entre les activités générées par des stimuli après auto-organisation du RNRP.....	104

Figure 5.18 : Comparaison des performances d'un neurone d'extraction avant et après l'auto-organisation du RNRP	105
---	-----

LISTE DES TABLEAUX

Tableau 5.1.A : Coefficients relatifs aux probabilités de connexion de 2 neurones.....	88
Tableau 5.1.B : Distributions des paramètres synaptiques.....	88
Tableau 5.2 : Coefficients des probabilités des connexions latérales.....	93
Tableau 5.3 : Valeurs des moyennes des distributions normales régissant les poids et les temps caractéristiques de dépression synaptiques.....	95

LISTE DES ÉQUATIONS

Équation 2.1 : Formule de calcul de la sensibilité synaptique.....	22
Équations 2.2 : Formules du modèle HH.....	32
Équations 2.3 : Formules régissant le comportement d'un neurone I&E.....	35
Équations 2.4 : Formules régissant le comportement d'un neurone « sigmoidal »..	37
Équation 3.1 : Fonction d'activation d'un neurone.....	45
Équations 3.2 : Formules de descente du gradient avec une erreur quadratique.....	46
Équations 3.3 : Loi de Hebb et expression des poids du réseau de Hopfield.....	48
Équation 4.1 : Formule du calcul de l'erreur pour un nœud.....	59
Équation 5.1 : Formule d'ajustement des poids synaptiques des connexions latérales.....	83
Équation 5.2 : Formule de la probabilité de connexion de 2 neurones.....	88
Équation 5.3 : Formules de mise à jour des poids synaptiques de la règle p-delta..	89
Équations 5.4 : Formules des probabilités des connexions latérales.....	93

LISTE DES ABRÉVIATIONS

CTAO : carte topographique auto-organisatrice

CTAOEL : carte topographique auto-organisatrice à état liquide

CGL : corps genouillé latéral

HH : Hodgkin-Huxley

I&E : intégrer-&-émettre (traduction de l'anglais « integrate-&-fire »)

K⁺ : ion potassium

LSM : « liquid-state machine »

MEL : machine à état liquide (traduction de l'anglais « liquid-state machine »)

Na⁺ : ion sodium

NT : neurotransmetteur

PA : potentiel d'action

RNPA : réseau de neurones à propagation avant

RNRP : réseau de neurones récurrents à pulses

RS : récepteur synaptique

V1 : cortex visuel primaire

1 Introduction

1.1 Les prothèses visuelles

Depuis le succès d'appareils comme le stimulateur cardiaque, la restauration de fonctions du corps humain en excitant le système nerveux afin de pallier ses déficiences (suite à un accident ou dues à une maladie par exemple) est maintenant étendue à de nombreuses fonctions : métaboliques (comme la vessie pour la miction), fonctionnelles (muscles des membres inférieurs pour la marche) ou sensorielles (l'ouïe, la vision). Dans le cas de la vision, le choix de la partie du système visuel à stimuler (la rétine, le nerf optique ou le cortex visuel primaire, cf. figures 1.1) soulève différents problèmes pratiques :

- stimuler le cortex permettrait de soigner un plus grand nombre de cas cliniques (notamment quand la rétine ou le nerf optique est endommagé),
- la rétine est plus facile d'accès, donc nécessite une opération chirurgicale moins complexe, en plus des problèmes liés aux phénomènes de rejet et les infections au niveau de l'interface entre le tissu biologique et le dispositif implantable,
- organiser la stimulation au niveau de la rétine est plus facile : la correspondance entre un site sur la rétine et dans le champ visuel est connue, contrairement au cas du cortex ou du nerf optique où la cartographie du champ visuel est prétraitée par les cellules rétinianes et n'est que partiellement connu,
- les signaux à envoyer au niveau du nerf optique ou de la rétine sont supposés être moins complexes que ceux destinés aux neurones corticaux.

On va s'intéresser par la suite uniquement à la stimulation du cortex, option choisie par l'équipe de PolySTIM. Cela implique d'organiser de telles stimulations électriques au niveau de circuits neuronaux très complexes, de sorte que les aires visuelles supérieures « interprètent » une scène dans le champ visuel conformément à la scène visuelle que

l'on veut représenter. On s'attend donc à de nombreuses difficultés pour parvenir à interagir finement avec ces circuits corticaux.

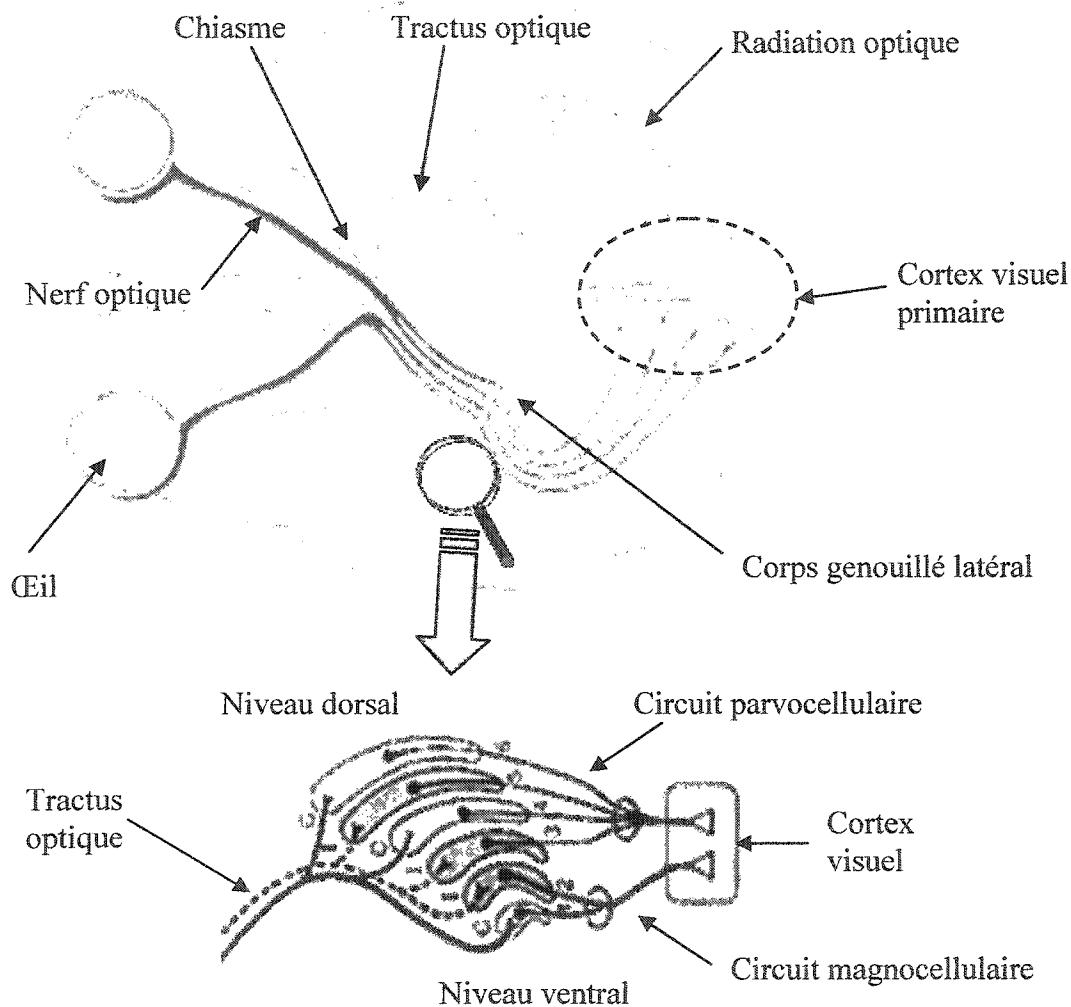


Figure 1.1.A : Vue de dessus du système visuel et détail du corps genouillé.

(lcavwww.epfl.ch/~alleyss0, www.psychology.psych.ndsu.nodak.edu)

L'information visuelle est prétraitée au niveau de la rétine puis acheminée par le nerf optique jusqu'au cortex visuel (dans le lobe occipital) via les corps genouillés latéraux (CGL). Les nerfs optiques se séparent au niveau du chiasme de manière que chaque CGL reçoive des informations binoculaires. À leur niveau, deux principaux circuits existent : la voie magnocellulaire et la voie parvocellulaire, qui sont spécialisées dans des tâches distinctes et se projettent vers la couche IV du cortex visuel primaire.

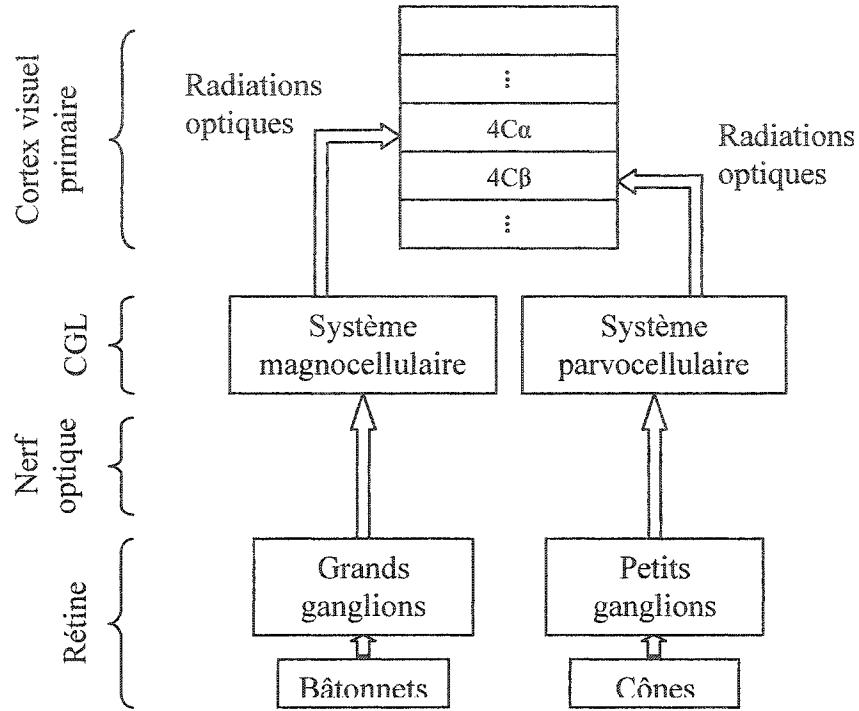


Figure 1.1.B : Schéma du système visuel de l'œil au cortex visuel primaire.
 En bas, au niveau de la rétine se trouvent les cellules réceptrices sensibles aux ondes électromagnétiques (cônes et bâtonnets) et les cellules ganglionnaires; au milieu, le nerf optique se projette dans les CGL (situés dans le thalamus); en haut, le cortex visuel primaire (VI) s'organise en 6 couches, et les radiations optiques innervent certaines sous-couches de la couche 4. VI se projette ensuite vers les aires « supérieures » de la vision et en retour vers le CGL. Seules quelques-unes des connexions intervenant dans le traitement de l'information visuelle sont représentées. Se référer à la partie 2.1.4 qui présente de manière plus détaillée le cortex visuel.

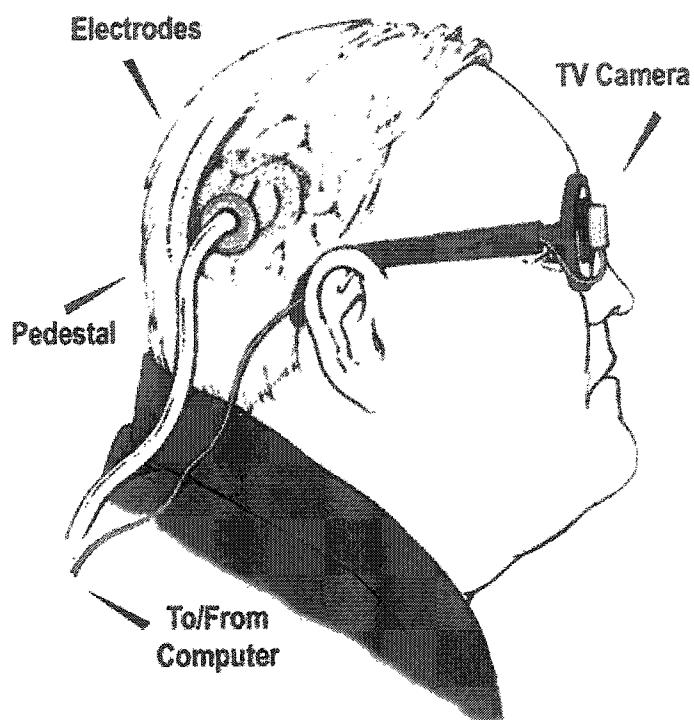
Dès les années 1930, des chercheurs tels que Foerster ont montré que stimuler le lobe occipital de l'encéphale de personnes devenues non-voyantes provoquait des **phosphènes**, sensations mentales d'un point lumineux dans le champ visuel [6]. Plus tard, en 1974, Mladéjovsky et Dobelle ont affiné les connaissances et la maîtrise de la création de tels phosphènes par **stimulation électrique du cortex visuel primaire** [5]. En 1996, Hambrecht et son équipe ont déterminé des paramètres de stimulation efficaces pour les différentes couches du cortex visuel primaire en utilisant des électrodes intracorticales avec des trains de pulses en courant de fréquence de l'ordre du kHz et une

amplitude d'environ $20\mu\text{A}$ on obtient des phosphènes stables d'une durée maximale de 1s environ. Ils ont également étudié l'impact de certains paramètres de stimulation (les durées et l'amplitude des trains de pulses, les fréquences utilisées, etc.) sur les qualités visuelles des phosphènes obtenus, ou encore l'induction de plusieurs phosphènes et les problèmes survenant lorsque deux électrodes de stimulation sont trop proches (moins de $400\mu\text{m}$ de distance). Ces études montrent la faisabilité de construire un microstimulateur intracortical implantable dans la couche 4 du cortex visuel (celle qui reçoit l'information visuelle via le nerf optique, cf. figure 1.1.B) [8] [26].

Au cours de la petite enfance, le cortex sensoriel s'organise en différentes aires lorsque celles-ci sont stimulées durant les premières semaines. Dans les cas de personnes devenues aveugles accidentellement, cette organisation s'est produite et l'on utilise les voies visuelles déjà créées pour restaurer une vision partielle. Le cas des personnes aveugles de naissance pose donc un autre problème : la partie corticale relative à l'aire visuelle a pu être « recrutée » pour servir d'autres fonctions sensorielles, notamment auditive. Il faudrait alors réentraîner cette partie du cortex afin de créer de nouvelles voies visuelles, ce qui s'avère être une tâche très ardue sachant qu'il faudrait proposer un certain codage de l'information visuelle au cortex et que le codage de l'information visuelle à ce niveau chez des sujets sains n'est encore que très partiellement compris. C'est pourquoi les études se limitent souvent aux sujets devenus aveugles suite à un accident ou une maladie (notamment toutes celles citées au paragraphe précédent), ce que l'on fera également au moins dans un premier temps.

En 2002, l'équipe de Dobelle a réussi grâce à des électrodes implantées en surface du cortex visuel à permettre à une personne devenue non-voyante de recouvrer une certaine autonomie de déplacement, notamment à lui permettre de conduire une voiture dans une zone de stationnement sans obstacle mobile [13], grâce à un champ visuel restauré de résolution 32 par 32 pixels qui est suffisante pour la majorité des déplacements selon cette équipe. Le dispositif comprend une caméra optique, des électrodes implantées à la surface du cortex et un système de traitement de l'information qui sélectionne les électrodes à activer pour donner des informations sur la position du

sujet par rapport à son environnement immédiat. Le lien entre les électrodes et le dispositif externe est un ensemble de fils traversant un orifice percé dans la boîte crânienne (cf. figure 1.2).



*Figure 1.2 : Schéma du dispositif de l'équipe de Dobelle. (www.dobelle.com)
Un ordinateur analyse la scène filmée devant le sujet par une caméra digitale pour envoyer des informations utiles à la stimulation aux électrodes via les fils de connexion.*

Cette prothèse présente plusieurs désavantages majeurs :

- la **faible résolution** de 32 par 32 pixels due au petit nombre d'électrodes utilisées (environ cent électrodes selon un article paru dans WIRED en septembre 2002),
- comme il s'agit d'électrodes de surface, l'**intensité des courants de stimulation** est élevée (de l'ordre du mA) ce qui peut provoquer des effets secondaires très indésirables (épilepsies notamment),
- son **encombrement** : cette prothèse ne bénéficie pas des derniers progrès en matière de miniaturisation (elle nécessite de porter un lourd appareillage à la ceinture) et les fils de connexion sont apparents et nombreux,

- enfin, selon les propos du chercheur D. Normann (responsable d'un autre projet de microstimulateur intracortical pour l'université d'Utah) interrogé par l'auteur d'un article sur ce sujet publié dans WIRED, on doit mieux comprendre **comment le patient peut exploiter les signaux qu'il reçoit** : est-ce qu'il utilise pour se mouvoir un certain codage en fonction des pixels allumés ou bien a-t-il une véritable scène représentée devant lui?

On voit donc que de tels résultats bien qu'encourageants sont encore très partiels en matière de maîtrise de la stimulation corticale et ne consistent pas encore en une « véritable » vision partielle, i.e. inclure des composantes visuelles plus fines. Les récents développements en micro-électronique laissent entrevoir un espoir d'y parvenir grâce à des **dispositifs miniaturisés versatiles implantables** permettant de gérer la micro-stimulation corticale, et donc de maîtriser plus finement la création de phosphènes et d'autres formes géométriques dans le champ visuel, ainsi que leur luminosité et leur couleur. Surtout, le passage du point à la ligne ou au segment constituera une avancée notable en matière de restauration de vision partielle. De plus, une telle micro-stimulation du cortex nécessite de plus amples connaissances sur le traitement de l'information visuelle (qui peut sembler être du bruit à priori) par les circuits neuronaux situés en amont vers l'œil, par exemple la correspondance entre la cartographie du champ visuel et celle du cortex [29]. De tels outils seraient donc très utiles pour étudier le système visuel *in vivo* dans la mesure où ils permettraient de collecter des données *in situ* sur des sujets en santé, et permettre d'améliorer la connaissance du système visuel.

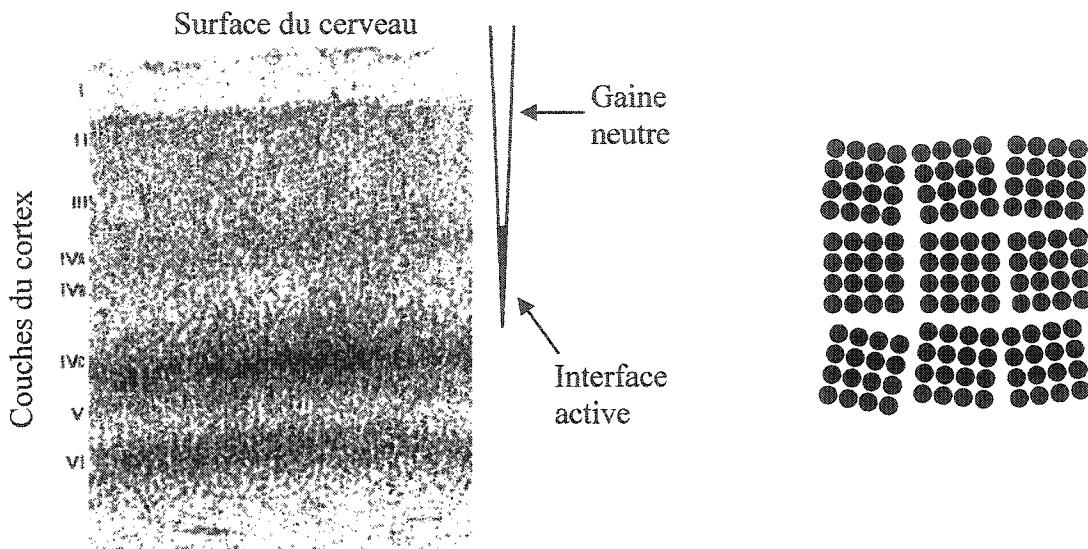
1.2 Le projet d'implant visuel cortical du laboratoire PolySTIM

Le laboratoire de microélectronique PolySTIM de l'École Polytechnique de Montréal développe actuellement un stimulateur électrique implantable dans la couche 4 du cortex visuel primaire [9]. La couche 4 a été retenue (comme pour d'autres projets de

prothèses visuelles corticales) car c'est elle qui reçoit l'information visuelle afférente via les radiations optiques et les études de micro-stimulation de cette partie du cortex ont montré que ce choix était judicieux [8] [26]. Comme la plupart des projets de prothèse visuelle, il se compose de trois modules : une partie interne implantée, une caméra vidéo et un module de traitement de l'information visuelle et d'alimentation électrique.

Une matrice d'électrodes monopolaires (de l'ordre de 1000 électrodes réparties sur une surface de l'ordre du cm^2 , cf. figure 1.3) est incrustée dans un module interne conçu pour pouvoir contrôler pour chaque électrode le courant émis ou reçu vers le milieu environnant au niveau de la partie active de l'électrode. A chaque cycle de fonctionnement du module interne (de l'ordre du kHz), un nombre limité d'électrodes de la matrice peuvent stimuler le tissu environnant en forçant un courant positif ou négatif dont l'intensité est paramétrable. Cela qui permet d'avoir une grande diversité de profil de courant pour stimuler les neurones (envoi de pulses, de signaux sinusoïdaux, etc.). Il est également possible de mesurer la différence de potentiel entre cette électrode et une référence, dans le but d'évaluer l'activité des neurones environnants, et cela pour un certain nombre d'électrodes durant chaque cycle. Les processus intervenant dans l'excitation des neurones par stimulation électrique en courant et dans la lecture de l'activité des neurones seront décrits dans le chapitre 2.

L'implant est relié au module externe par un lien haute-fréquence qui transmet l'énergie nécessaire ainsi que les informations utiles à l'« affichage » dans le champ visuel. La caméra vidéo collecte, sur l'environnement extérieur de la personne portant la prothèse, des données pertinentes qui sont traitées par le module externe pour être transformées en informations directement utilisables par l'implant (on peut également penser à d'autres appareils comme un sonar, etc.). Le module interne organise ensuite la stimulation pour la matrice d'électrodes de manière à ce que les aires cérébrales supérieures de la vision « interprètent » conformément à ce que l'on veut faire « apparaître » dans le champ visuel.



À gauche, les électrodes sont implantées de manière que leur interface active se trouve à la profondeur de la couche 4 du cortex visuel primaire (<http://webvision.med.utah.edu/>). À droite, vue de dessus de l'organisation de la matrice d'électrodes en sous-matrices 4 par 4, ce qui permet notamment de mieux « coller » à la surface corticale.

À la différence du dispositif de l'équipe Dobelle & al., un tel implant interne n'a pas de lien physique avec l'extérieur et présente normalement une durée de vie beaucoup plus longue : il y a moins de risques d'infections comme au niveau du contact vivant-électronique à l'air libre et l'implant encapsulé devrait s'intégrer convenablement dans l'organisme à long terme [23]. De plus, le fait d'implanter au sein du cortex et non en surface les électrodes abaissera d'un facteur mille (on parle de μA au lieu de mA) les seuils de stimulation efficace en courant et augmentera la finesse de la stimulation, de même que la résolution du champ visuel recréé grâce au nombre plus important d'électrodes. Enfin, l'électronique embarquée devrait permettre pour l'implant un fonctionnement souple et réactif ainsi qu'effectuer un certain nombre de tâches préprogrammées pour en faire un véritable **outil d'investigation du cortex** : on pourrait par exemple analyser l'activité de certains neurones au niveau des potentiels d'action ou encore effectuer des tests de synchronisation entre plusieurs aires du cortex grâce à la lecture de l'activité du tissu vivant en de nombreux points simultanément.

Pour recréer une vision partielle, on peut envisager dans un premier temps de reconstituer en mode « fil de fer » la scène devant le sujet (cf. figure 1.4) que l'on reproduit de manière pixellisée avec des phosphènes dans le champ visuel restauré. Mais on aimeraient à terme substituer à cet affichage d'une basique matrice de pixels une vision plus élaborée en exploitant mieux la micro-stimulation du cortex. On espère ainsi pouvoir gérer des composantes de l'information visuelle telles que la forme géométrique (pas seulement des phosphènes mais des lignes, etc.), la luminosité (pouvant être utile pour faire percevoir le relief), la couleur, le mouvement (afin d'organiser un flux d'images). Cela nécessite d'**optimiser les paramètres de la stimulation corticale** et de mieux comprendre le **codage de l'information visuelle** au niveau du cortex visuel primaire. Sur ce dernier point, on est en présence de plusieurs millions de neurones par mm^3 reliés par des milliards de connexions et dont l'activité *in vivo* est difficile à évaluer, on ne peut donc espérer pouvoir reproduire le fonctionnement d'un système visuel sain et il s'agit plutôt de trouver comment utiliser efficacement les circuits neuronaux du cortex.

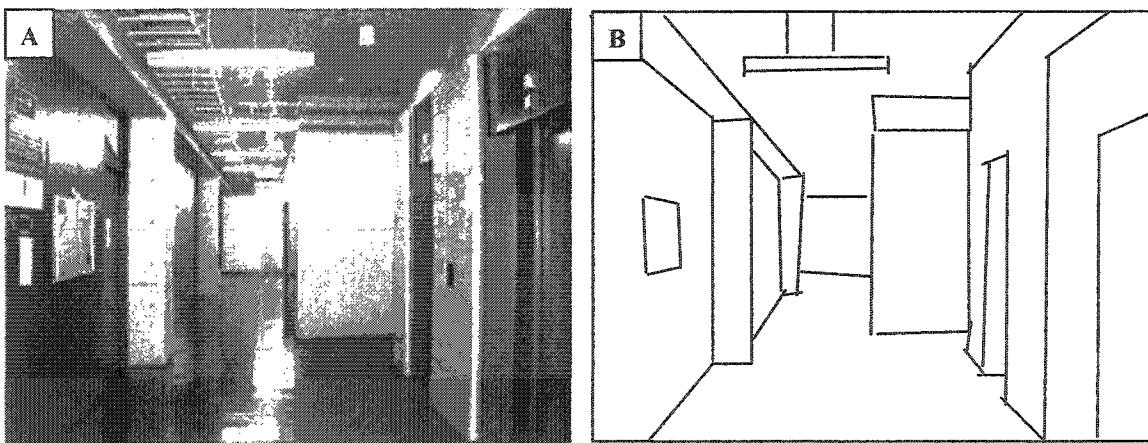


Figure 1.4 : Reconstitution d'une scène en mode fil de fer.
(http://www.humanoid.rise.waseda.ac.jp/booklet/booklet2000_narita.html pour A)
 On extrait de la scène réelle (A) les arêtes les plus significatives (B) pour la compréhension de la scène devant le sujet, et on affiche les pixels correspondant à ces arêtes dans le champ visuel reconstitué.

Dans un premier temps, on pense utiliser des trains de pulses biphasiques de courant comme base des profils de stimulation (cf. partie 2.1.3 pour un passage en revue de différents moyens de stimuler électriquement des neurones au moyen d'électrodes). De tels trains de pulses semblent être de bons candidats [24] selon plusieurs études comme celles de l'équipe d'Hambrecht [8]. Il s'agit donc d'une part de déterminer quels paramètres de ces trains d'onde sont importants et comment les ajuster pour obtenir des profils de stimulation efficaces. D'autre part, on sera amené à combiner les stimulations de plusieurs électrodes pour exciter le cortex visuel en différents sites, ce qui augmente encore le nombre déjà considérable de paramètres à optimiser (cf. figure 1.5).

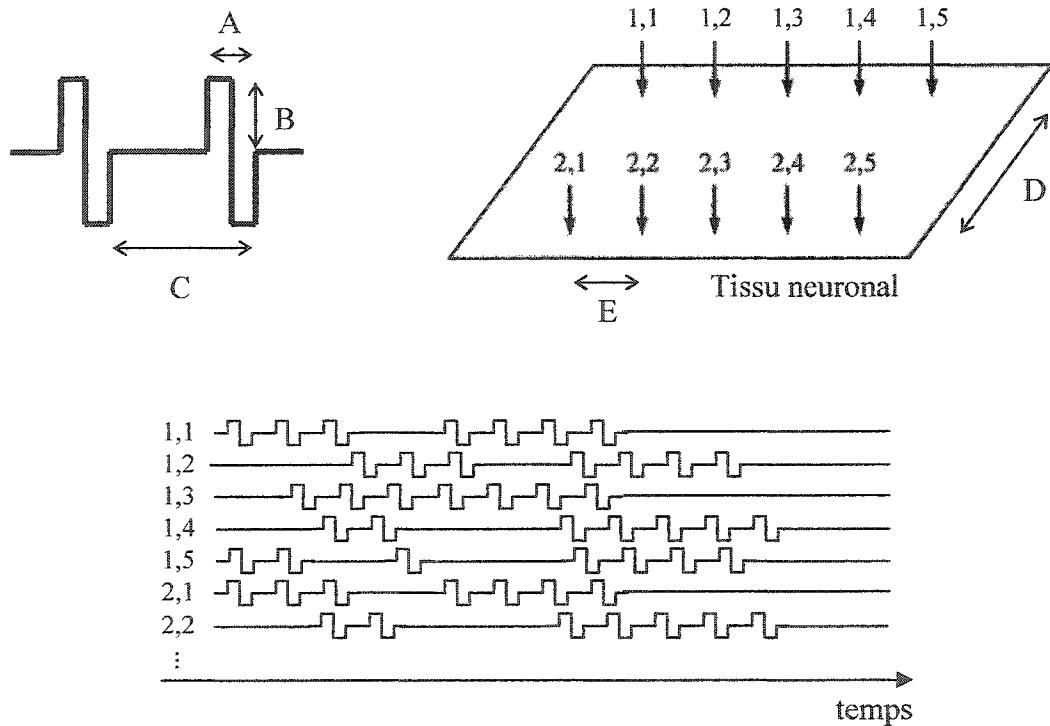


Figure 1.5 : Exemple de paramètres intervenant dans la stimulation.

En haut à gauche, un exemple de train de 2 pulses où interviennent comme paramètres l'amplitude B et la durée A d'une pulse, le délai inter-pulses C . En haut à droite, les sites de stimulation dans le tissu biologique : les flèches verticales en gras représentent les électrodes numérotées (les espacements inter-électrodes sont notés D et E). En dessous, un exemple de trains de pulses appliqués à plusieurs électrodes avec les instants de début et de fin de chaque train de pulses (du type de celles de la figure en haut à gauche), leur nombre de pulses, etc. qui constituent autant de paramètres à ajuster.

1.3 Simuler le tissu neuronal pour trouver des paramètres adéquats de stimulation

On a donc un grand nombre de paramètres de stimulation à optimiser, mais le faire sur des cobayes vivants serait coûteux et fastidieux, compte tenu par exemple de la durée légale maximale d'expérimentation sur un cobaye. On doit également dégager derrière le « bruit » apparent de l'activité du tissu des informations pertinentes relatives à la représentation d'une scène visuelle. Ainsi, des stimulations ayant les « mêmes » paramètres (autant qu'on puisse les contrôler) n'engendrent pas des activités semblables au niveau des trains de pulses de chaque neurone. Inversement, il serait intéressant de déterminer quels types de stimulations engendrent des réactions similaires du tissu. De plus, on désire anticiper et éviter divers phénomènes pouvant perturber le fonctionnement de l'implant, comme la saturation du tissu neuronal due à la combinaison de certains profils de stimulation (avec des dommages potentiels pour le sujet implanté) ou encore des « interférences » lorsque des électrodes voisines stimulent en même temps (ce qui ne produirait pas l'effet escompté au niveau du champ visuel).

Plus accessoirement, comme l'énergie transmise à l'implant interne par le lien haute-fréquence est limitée, on souhaiterait répartir dans le temps sa consommation électrique en optimisant l'organisation temporelle et spatiale des stimulations pour les électrodes (puisque un nombre limité d'électrodes peuvent être activées en même temps). Ou encore, dans l'hypothèse où une électrode deviendrait défectueuse, on aimerait pouvoir combiner des électrodes voisines pour la remplacer, en interpolant ainsi une électrode virtuelle.

Pour toutes ces raisons, on a besoin d'un modèle qui nous permette de simuler le comportement du tissu biologique lors du fonctionnement de l'implant : **un modèle de tissu neuronal « biologiquement réaliste »** et un modèle d'électrode notamment. On se limite dans notre modèle aux neurones, étant donné qu'à priori les autres cellules du cortex n'interviennent pas dans le traitement de l'information visuelle. Nous voulons donc utiliser des réseaux de neurones artificiels de style biologique, définir une

architecture inspirée de la physiologie corticale et les optimiser pour obtenir un comportement « réaliste ». Cela implique non seulement d'être conforme à certaines caractéristiques physiologiques (densité de neurones, de connexions, etc.), mais également d'être capable de réaliser certaines tâches (reconnaissance par des neurones spécifiques de certains profils de stimulation par exemple). La figure 1.6 donne un exemple d'expériences permettant d'enregistrer certains aspects du comportement du tissu biologique à reproduire dans le modèle, en l'occurrence les extensions temporelle et spatiale de l'activité nerveuse due à une stimulation.

Une des contraintes liées aux caractéristiques biologiques est l'immense nombre de neurones à prendre en compte, compte tenu des dimensions spatiales de la portion du tissu cortical visuel à modéliser et de la densité moyenne de neurones. Il faut déterminer jusqu'à quel niveau de précision, notamment si l'on représente chaque neurone ou des groupes de neurones.

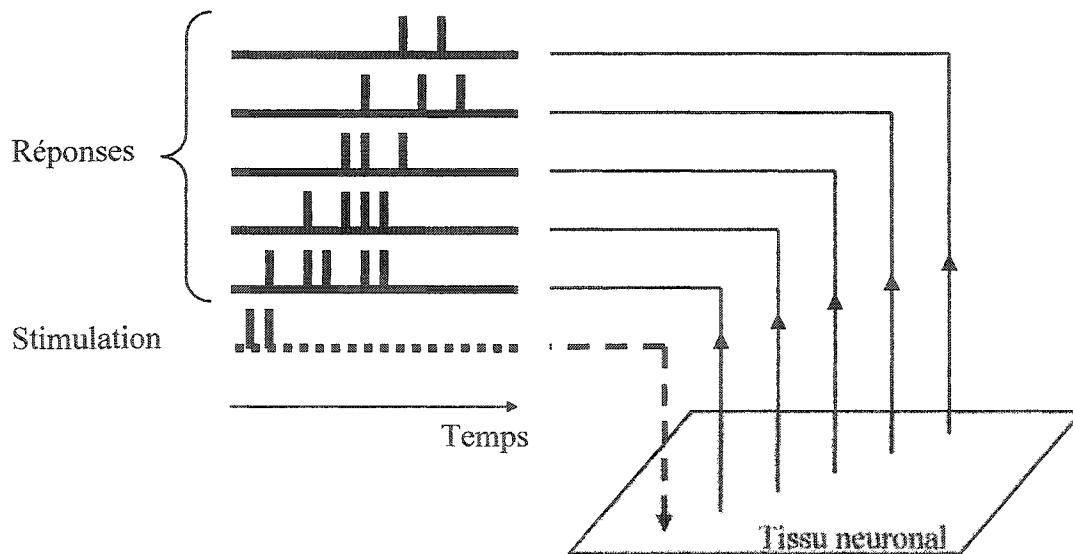


Figure 1.6 : Exemple de mesures biologiques de l'activité neuronale d'un tissu suite à une stimulation électrique.

Sur le tissu neuronal, les flèches représentent les électrodes (celle stimulatrice à gauche en gras et pointillés, celles de lecture en trait plein). En haut à gauche sont représentés les trains de pulses relatifs aux électrodes. L'idée serait de reproduire les extensions spatiales et temporelles dans une direction spécifique de l'activité du tissu dans un réseau artificiel, en testant aussi si ces propriétés sont isotropes ou non.

Lorsqu'un tel tissu artificiel sera entraîné, on aura la possibilité de tester son comportement pour d'autres protocoles de stimulation, afin de prédire certains phénomènes intéressants et de les confronter ensuite à des résultats expérimentaux (cela pourrait être par exemple la présence d'interférences ou non entre électrodes voisines pour certains profils de stimulation). On aura ainsi à terme un modèle artificiel du tissu biologique et un dispositif implantable qui seront complémentaires pour étudier le cortex visuel, et pourquoi pas ultérieurement d'autres régions corticales de l'encéphale.

On va donc étudier comment construire un tel modèle de tissu cortical au moyen de réseaux neuronaux artificiels de style biologique : on décrira d'abord quels aspects et quelles fonctionnalités du tissu biologique on veut prendre en compte, puis on présentera plusieurs candidats possibles de réseaux artificiels (type de neurones, architecture, etc.), et on étudiera les paramètres à optimiser dans les réseaux choisis et les algorithmes pertinents pour le faire. On proposera également des expériences susceptibles de servir de référence pour ces optimisations.

2 Modélisation du tissu neuronal de la couche 4 du cortex visuel primaire

2.1 Description physiologique du tissu biologique

La partie 2.1 présente la physiologie du cortex visuel, depuis la description d'un seul neurone et des connexions synaptiques jusqu'à l'organisation corticale fonctionnelle destinée à traiter l'information visuelle. Elle est relativement détaillée et longue, ceux qui connaissent déjà le sujet peuvent passer à la partie 2.2 et se contenter de la partie 2.4 où les caractéristiques physiologiques prises en compte pour la construction du modèle sont récapitulées.

2.1.1 Physiologie et fonctionnement du neurone

Les neurones sont les cellules de base du système nerveux et ils sont connectés entre eux pour former un réseau qui traite et transmet l'information nerveuse. Dans un neurone, on peut distinguer 3 « parties » (cf. figure 2.1) : des **dendrites** supportant des récepteurs synaptiques qui correspondent à des « entrées », un **corps cellulaire** où l'information nerveuse est intégrée et un **axone** qui se ramifie à son extrémité où se trouvent des boutons synaptiques (les « sorties »). Deux neurones sont connectés par une **synapse**, i.e. le site entre un bouton synaptique du neurone-source et des récepteurs situés sur les dendrites du neurone-cible (cf. partie 2.1.2).

La géométrie des neurones varie suivant leurs fonctionnalités (et donc selon la partie du système nerveux auquel ils appartiennent). On peut les différencier suivant :

- le nombre de neurites (arbres dendritique et axonique), ce qui distingue les neurones monopolaires, bipolaires et multipolaires (cf. figure 2.2),

- la taille des neurites (longs axones pour les nerfs) et leur arborescence plus ou moins touffue (quelques dizaines de ramifications pour les dendrites de neurones pyramidaux contre plusieurs centaines pour celles des cellules stellaires dans le cortex visuel par exemple),
- la présence d'« épines » sur les dendrites (sur les cellules pyramidales dans le cortex par exemple), ce qui augmente la surface supportant les récepteurs des synapses et que l'on suppose intervenir dans des phénomènes d'apprentissage et d'auto-adaptation (cf. partie 2.1.4),
- la myélinisation de l'axone : des cellules gliales de Schwann enroulées autour de l'axone forme une gaine multicouche qui accélère la conduction des PA (présent principalement sur les longs axones formant des nerfs).

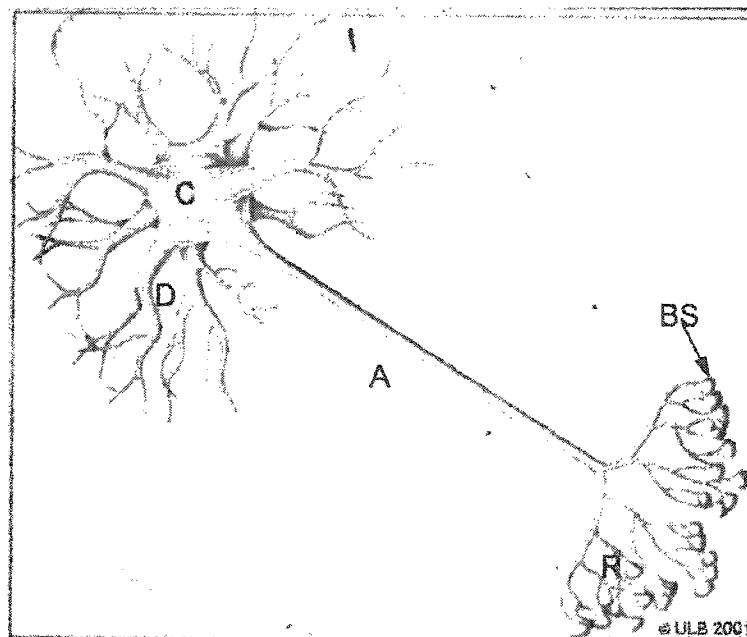


Figure 2.1 : Structure générale d'un neurone. (www.ulb.ac.be)

Il est constitué d'un soma (ou corps cellulaire, C) duquel partent des dendrites (D) et un axone (A) dont l'extrémité se ramifie (R) et porte des boutons synaptiques (BS). Les récepteurs synaptiques se trouvent essentiellement sur le soma (synapses inhibitrices) et les dendrites (synapses excitatrices). L'information nerveuse au sein d'un neurone est recueillie par les dendrites au niveau de connexions synaptiques et se propage ensuite vers le soma, où se forme dans certaines conditions un PA qui se propage le long de l'axone vers les terminaisons synaptiques

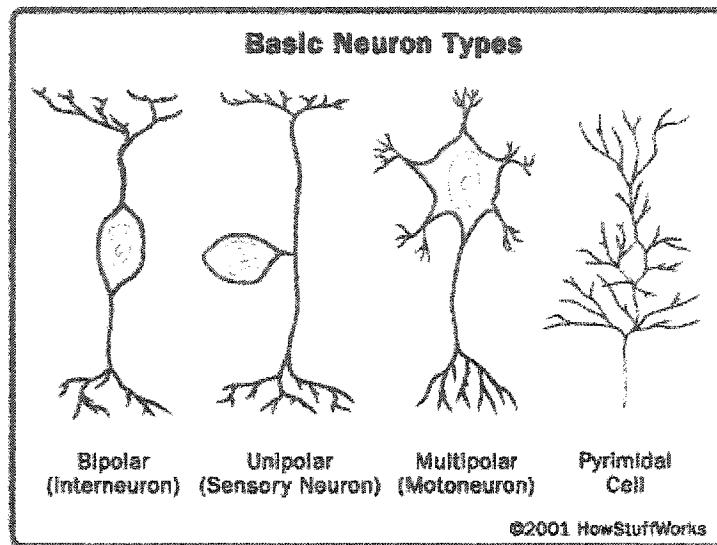


Figure 2.2 : Différentes géométries de neurones. (<http://static.howstuffworks.com>)
 Différents types de neurones classés selon la nature de leurs neurites. Ces différentes géométries correspondent à des fonctionnalités différentes : par exemple, la cellule sensorielle réagit à des stimulations de l'environnement extérieur et émet des pulses par son axone tandis qu'un inter-neurone intègre des informations nerveuses provenant d'autres neurones pour émettre une fonction de ces informations. Plus elle est touffue, plus l'intégration de l'information nerveuse peut être complexe.

L'activité d'un neurone se traduit dans les variations des propriétés électriques de sa membrane. L'excitation de récepteurs synaptiques situés sur les dendrites engendre des variations du **potentiel transmembranaire** qui convergent vers le corps cellulaire où elles sont intégrées. Lorsque le potentiel au niveau du corps cellulaire dépasse un certain seuil, il se produit une **dépolarisation** appelée **potentiel d'action** (PA) au niveau de la jonction du soma et de l'axone. Ce PA se propage ensuite le long de l'axone vers les boutons synaptiques où il entraîne l'excitation de récepteurs synaptiques sur d'autres neurones (cf. partie 2.1.2). Les variations du potentiel correspondent à des mouvements ioniques (principalement sodium Na^+ et potassium K^+ , mais aussi chlore Cl^- et calcium Ca^{++}) dans des canaux transmembranaires spécifiques à chaque type d'ions (cf. figures 2.3). Les mécanismes d'ouverture et de fermeture de ces pores sont commandés par le potentiel transmembranaire.

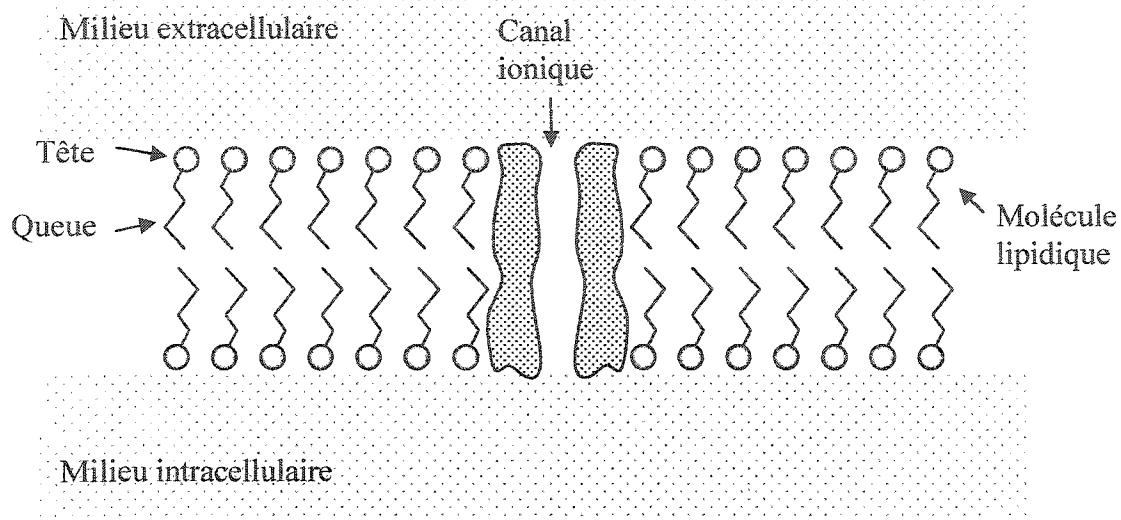


Figure 2.3.A : Schéma d'un canal ionique de la membrane d'un neurone.
La membrane se compose d'une double couche de molécules lipidiques (en majorité des phospholipides) possédant une tête hydrophile et une chaîne caudale hydrophobe. Les canaux ioniques sont de grosses protéines incrustées dans cette double couche.

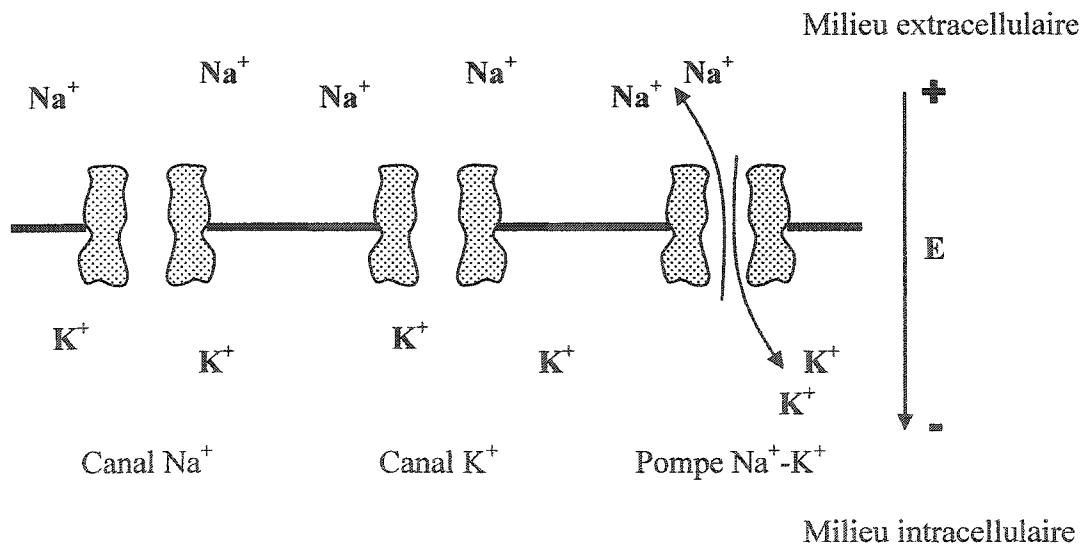


Figure 2.3.B : canal ionique de la membrane d'un neurone.
Les équilibres et variations du potentiel transmembranaire (E) sont régis en grande partie par les canaux spécifiques à Na^+ , ceux à K^+ et les pompes $\text{Na}^+ - \text{K}^+$. Les pompes fonctionnent en permanence pour entretenir un déséquilibre des concentrations qui correspond à un potentiel de repos négatif (les ions Na^+ et K^+ sont représentés du côté de leur concentration majoritaire au repos). L'ouverture des canaux est commandée par le potentiel et les déplacements des ions à travers eux sont spontanés du fait des différences de concentrations de part et d'autre de la membrane (en trait gras).

Les pompes $\text{Na}^+ \text{-K}^+$ maintiennent un déséquilibre entre les concentrations des ions de part et d'autre de la membrane au repos, et le potentiel est alors de -65mV. Une variation de potentiel vers 0mV entraîne l'ouverture de canaux Na^+ proportionnellement à son amplitude, et quand le potentiel dépasse un certain seuil (-40mV) l'ouverture s'emballe telle une réaction en chaîne, ce qui dépolarise la membrane avec un pic à 30mV (il s'agit d'un PA). NB : les valeurs données ici sont des ordres de grandeur. Les phases de montée et de descente du potentiel sont très rapides : respectivement 1ms et 2ms environ (cf. figure 2.4.A). L'hyperpolarisation constitue une période réfractaire de quelques ms durant laquelle le neurone est pratiquement insensible à toute nouvelle excitation. Un site dépolarisé affecte les canaux situés à son voisinage, ce qui entraîne la propagation du PA. Spontanément, celle-ci est unidirectionnelle le long d'un neurite du fait de la période d'inactivation des canaux Na^+ succédant à leur ouverture (cf. figure 2.4.B). La vitesse de propagation (de l'ordre de 5 à 25 ms^{-1}) le long d'un neurite dépend de la géométrie du neurite (diamètre, etc.) et de sa physiologie (résistivité radiale et longitudinale). Une variation sous-seuil (par exemple créée par une synapse) engendre une dépolarisation partielle, qui se propage de même qu'un PA. Hodgkin et Huxley ont établi un modèle électrique de la membrane rendant compte de ces phénomènes en 1952 (cf. partie 2.2.1).

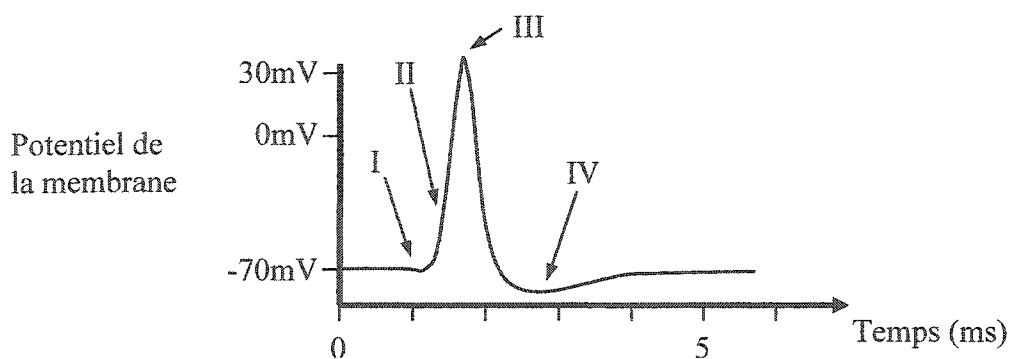


Figure 2.4.A : Potentiel d'action - description des phases des canaux ioniques.
Variations du potentiel transmembranaire lors du passage d'un PA avec les différentes phases des canaux ioniques : ouverture de quelques canaux Na^+ (activation, I), ouverture d'un grand nombre de canaux Na^+ (après dépassement du seuil d'activation, II), inactivation des canaux Na^+ et ouverture des canaux K^+ (pic de dépolarisation, III) et finalement fermeture des canaux K^+ (hyperpolarisation postérieure et relaxation, IV).

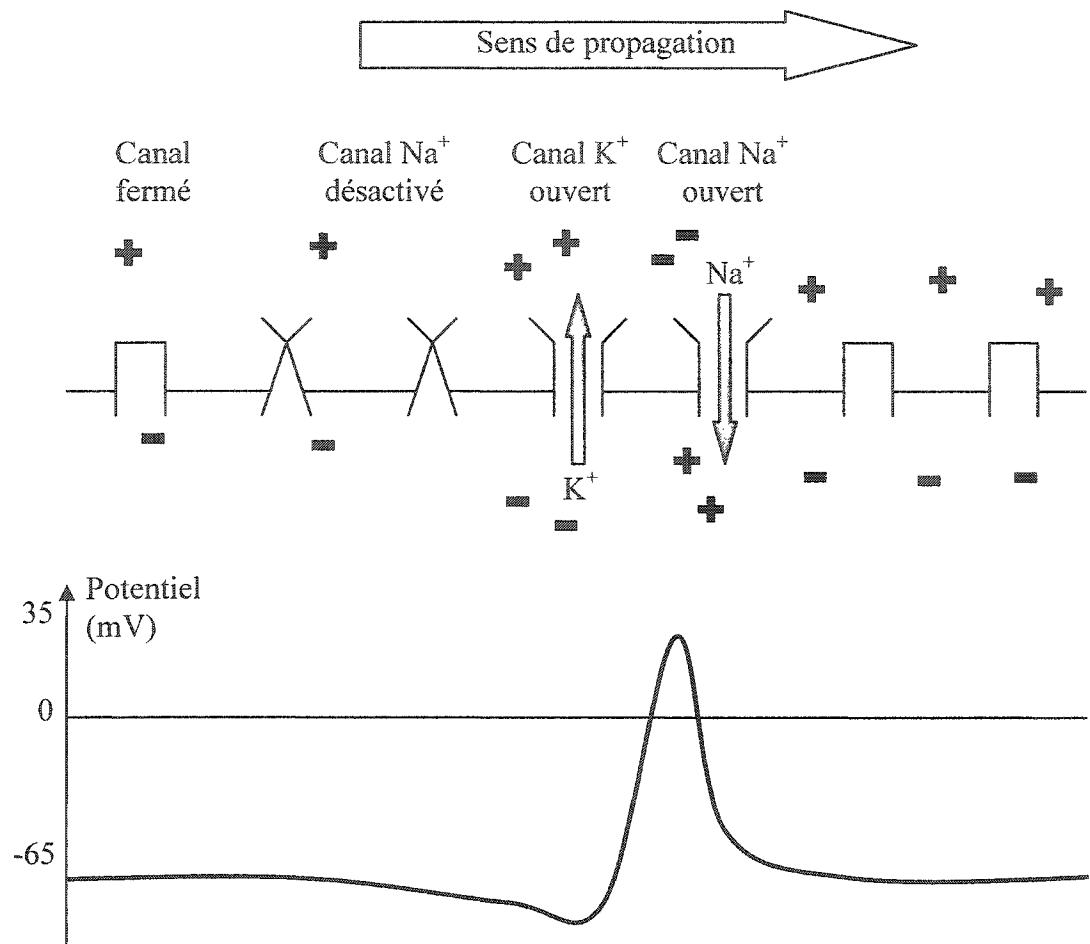


Figure 2.4.B : Potentiel d'action - propagation.

Schéma du déplacement du PA : la variation du potentiel transmembranaire entraîne l'ouverture des canaux Na^+ , phénomène qui augmente la dépolarisation, donc qui est auto-entretenue, jusqu'à inactivation de ceux-ci, puis ouverture des canaux K^+ (ici le schéma ne distingue pas les deux types de canaux). L'absence d'inactivation des canaux K^+ est responsable de l'hyperpolarisation. Les variations du potentiel le long de la membrane sont affichées dans le graphique en dessous. Le PA se déplace sans perte d'amplitude : la membrane est dite active. La propagation est unidirectionnelle du fait de l'inactivation des canaux Na^+ après ouverture.

Ainsi, les PA constituent des événements brefs dont les séquences temporelles émises par les neurones codent l'information nerveuse (cf. partie 2.1.5) et sont transmises d'un neurone à l'autre dans un réseau par les connexions synaptiques.

2.1.2 Connexions des neurones

Une connexion entre deux neurones se fait entre un bouton synaptique et des récepteurs synaptiques (RS) situés en général sur les dendrites dans le cas excitateur et sur le soma dans le cas inhibiteur (où la création de PA est alors bloquée). Le bouton synaptique contient des réservoirs à neurotransmetteurs (NT) dont l'ouverture dépend de l'excitation du bouton par un PA. Libérés dans l'espace synaptique, les NT se fixent sur des récepteurs spécifiques, ce qui provoque l'ouverture de canaux ioniques alentour sur le neurone-cible (cf. figure 2.5). Plus un bouton est excité par des PA, plus le nombre de réservoirs vidés est élevé et plus les RS sont activés par les NT. Parallèlement, les réservoirs nécessitent une période caractéristique pour reconstituer leur stock de NT. La sensibilité des synapses, qui correspondant à l'impact de l'activité du neurone-source sur l'excitation du neurone-cible, est dynamique sous l'influence de ces phénomènes, entre autres.

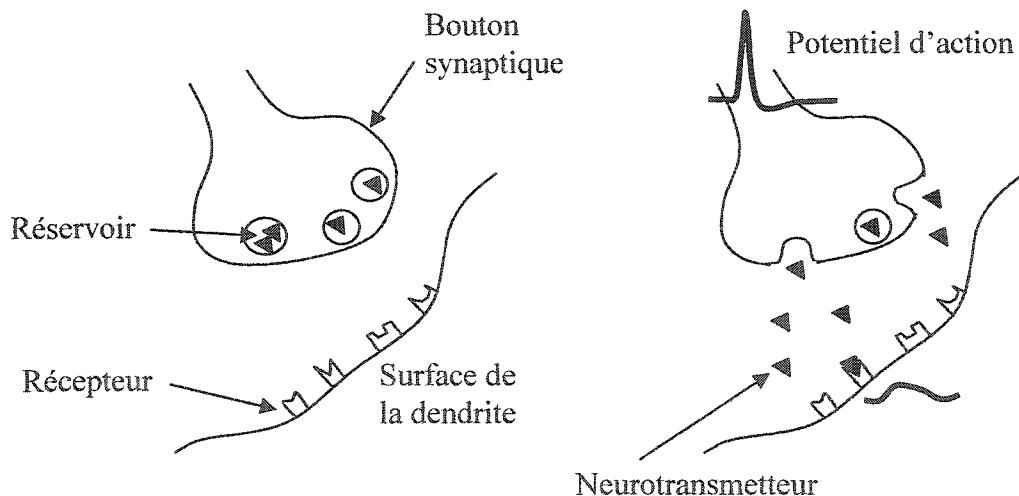


Figure 2.5 : Fonctionnement d'une synapse biologique.

L'arrivée d'un PA au niveau d'un bouton synaptique (à droite) provoque l'ouverture de réservoirs contenant des NT qui se répandent dans l'espace synaptique. Lorsqu'ils se fixent sur des RS leur correspondant, ils créent une dépolarisation locale (pas forcément sur-seuil) de la membrane portant ces RS.

En première approximation, on peut décrire la sensibilité synaptique dynamique au moyen de deux mécanismes simultanés : la facilitation (F) et la dépression (D) [19]. Le premier augmente la sensibilité chaque fois que la synapse est excitée tandis que le second la fait diminuer (cf. équation 2.1). La combinaison des mécanismes de facilitation et de dépression peut engendrer des comportements assez complexes rien qu'au niveau d'une seule synapse (cf. figure 2.6). La dynamique des synapses joue un rôle important dans la puissance des réseaux de neurones biologiques [20].

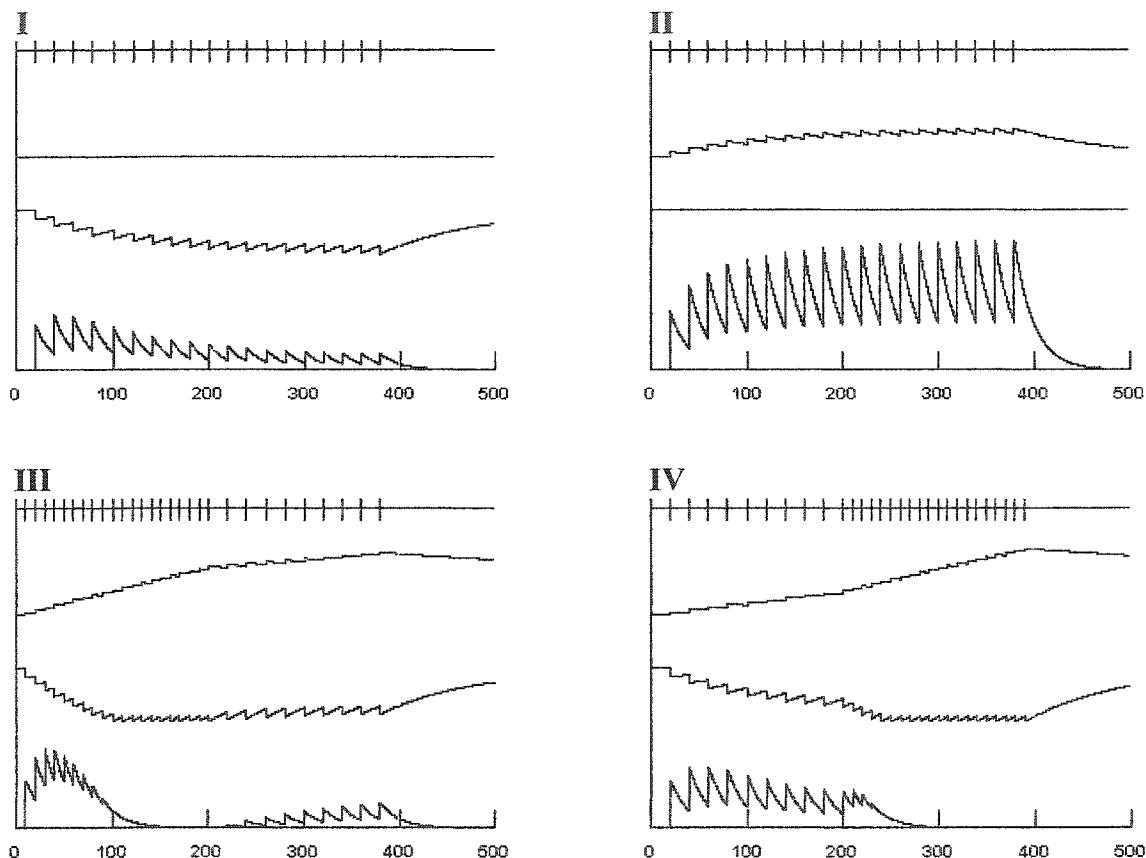


Figure 2.6 : Exemples de comportement d'un modèle de synapse dynamique.
Chaque graphique montre en haut le train de pulses en entrée de la synapse, les variations de F et D ensuite et en bas le potentiel post-synaptique créé, pour le modèle décrit par l'équation 2.1 (avec $\tau_D=10\text{ms}$; $A_D=0,15$; $\tau_F=100\text{ms}$; $A_F=0,1$; en abscisses le temps en ms). On voit l'effet de la dépression seule (I) et celui de la facilitation seule (II), puis la combinaison des deux lorsque la fréquence des trains est divisée par 2 (III) et enfin lorsqu'elle est multipliée par 2 (IV). Une synapse peut ainsi jouer un rôle de filtre en fréquence (passe-bas dans le cas de III et IV).

$$s = s \times D \times (1 + F)$$

$$D = D - \Delta_D \text{ si la synapse est activée, } D = 1 - (1 - D) \times \left(1 - \frac{1}{\tau_D}\right) \text{ sinon}$$

$$F = F + \Delta_F \text{ si la synapse est activée, } F = F \times \left(1 - \frac{1}{\tau_F}\right) \text{ sinon}$$

Équation 2.1 : Formule de calcul de la sensibilité synaptique.

Ce modèle de synapse dynamique simplifié comporte deux variables : la facilitation F et la dépression D , qui influent sur la sensibilité s . Chacune des ces variables correspond à un temps caractéristique τ au bout duquel la variable diminue son écart avec sa valeur d'équilibre de 64%, et une grandeur Δ qui est ajoutée/retranchée lorsque la synapse est activée par une pulse en entrée.

Le modèle proposé consiste en une version simplifiée de celle de Tsodyks, mais elle permet déjà de rendre compte de nombreux phénomènes dynamiques, a été utilisée dans d'autres études et est relativement rapide à calculer.

2.1.3 Stimulation et lecture de l'activité dans un tissu neuronal

Il existe plusieurs façons de stimuler des neurones, notamment magnétiquement ou électriquement. Dans le dernier cas, on peut injecter un courant ou imposer un potentiel pendant une durée donnée grâce à différents types d'électrodes (monopolaires ou bipolaires, extra- ou intra-cellulaires). De manière générale, les sections plus grosses (soma ou jonction soma-axone) sont les plus facilement excitables, on peut considérer en première approximation qu'une stimulation déclenche la création de PA à ces niveaux. À un niveau très raffiné, l'électrode crée des déplacements d'ions qui se propagent dans le milieu extracellulaire et qui créent des variations du potentiel transmembranaire ce qui peut conduire à un potentiel d'action lorsque l'intensité dépasse un certain seuil. L'effet de l'électrode peut ainsi être influencé par le milieu extérieur (anisotropies, etc.)

L'équipe PolySTIM a choisi d'utiliser des électrodes monopolaires pour stimuler les cellules corticales par injection de courant dans le milieu extracellulaires (cf. figure 2.7). L'implant sera capable d'envoyer un courant d'intensité contrôlée durant chaque cycle de fonctionnement, ce qui permet de reproduire une grande diversité de profils de stimulation (approximation d'une sinusoïde, pulses, etc.). Il serait intéressant d'exploiter cette versatilité pour effectuer des **stimulations fonctionnelles** : ajuster les protocoles de stimulations peut permettre d'exciter uniquement certains types de neurones parmi tous ceux alentours. Ce phénomène est utilisé notamment pour les nerfs, où l'on sélectionne les fibres excitées selon leur diamètre en adaptant les fréquences et les amplitudes des trains de pulses de courant. Dans le cas du cortex, on voudrait de même pouvoir stimuler préférentiellement certains types de cellules (les somas des cellules pyramidales sont au moins 3 fois plus gros en moyenne que ceux des cellules stellaires) en vue d'obtenir une plus grande finesse de stimulation.

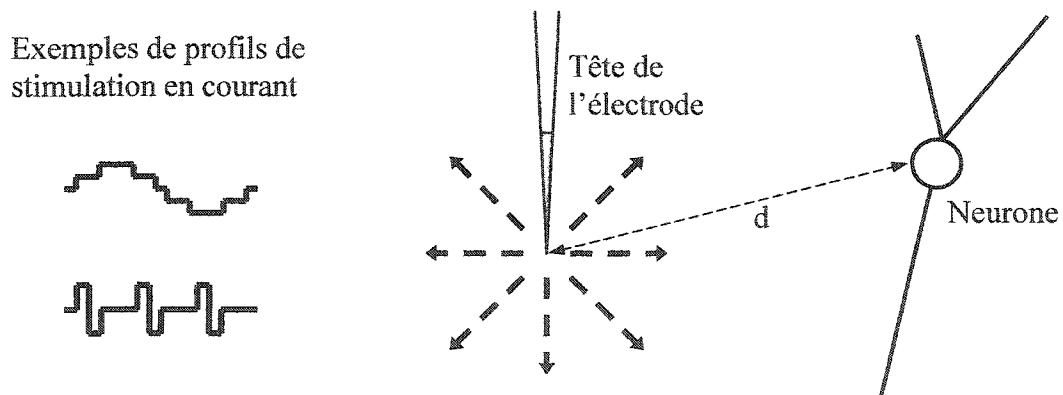


Figure 2.7 : Simulation en courant par une électrode monopolaire.
L'électrode provoque des mouvements d'ions par injection de courant (représentés par les flèches en gras et en pointillés), ce qui perturbe les équilibres au voisinage des cellules les plus proches. Les membranes réagissent par dépolarisation, et lorsque celle-ci dépasse le seuil d'activation, un PA est créé (on peut supposer que cela intervient au niveau du soma). Pour chaque neurone, il existe donc un seuil d'intensité (fonction notamment de la distance d) au-dessus duquel une stimulation engendre un PA. À gauche, on présente deux exemples de profils de stimulation en courant : approximation d'une sinusoïde en haut, un train de pulses biphasiques en bas.

Lorsque l'on utilise des électrodes monopolaires extracellulaires classiques pour enregistrer l'activité du milieu alentour, on obtient un mélange des activités de plusieurs neurones avec un bruit ambiant. On ne maîtrise généralement pas le nombre de neurones pris en compte (entre 0 et 4 en général), même si on peut l'évaluer après coup. Il est alors nécessaire d'effectuer un traitement du signal obtenu afin de séparer les PA du bruit et des artefacts, et de séparer les activités des différents neurones autant que possible. Récemment, une équipe de l'université d'Iowa a mis au point une électrode « de profondeur hybride » (i.e. hybrid depth electrode) qui permet d'enregistrer l'activité d'un seul neurone depuis le milieu extracellulaire. Pour l'instant, l'équipe PolySTIM n'envisage pas d'utiliser ce nouveau type d'électrodes, car les électrodes de la matrice doivent pouvoir effectuer alternativement des lectures et des stimulations, être robustes en vue de l'implantation chirurgicale, etc. On est donc contraint de traiter les signaux enregistrés, ce qui devra être inclus à terme dans le modèle global du tissu avec l'implant.

Il est donc clair que l'optimisation des protocoles de stimulation de l'implant nécessite un modèle d'électrode adapté au modèle du tissu neuronal. Par exemple, si l'on prend en compte les pulses individuellement pour analyser les séquences de pulses, le modèle d'électrode doit intégrer des délais dans le calcul de l'effet à distance. De même, il doit rendre compte des anisotropies de l'effet de la stimulation sur les neurones dans le cas d'une reconstruction fine en 3D du tissu. On pourrait ainsi entraîner un modèle pour interpoler l'impact sur un neurone produit par n'importe quel profil de stimulation, en tenant compte de la géométrie du système, des variations statistiques du milieu extracellulaire, etc. On peut se contenter pour l'instant de considérations de base : on suppose que tous les neurones d'un voisinage sphérique autour de la pointe de l'électrode sont affectés par une stimulation avec un impact proportionnel à l'intensité du courant émis et inversement proportionnel à la distance et que l'on a accès à l'activité d'un neurone (le voisin le plus proche) par électrode en lecture. Cela servira à faire le lien avec la réalité biologique mais on n'utilisera pas par la suite de modèle d'électrode stimulatrice en tant que tel, la présente étude est focalisée sur le seul tissu neuronal.

2.1.4 Organisation du cortex visuel primaire

Le cortex correspond à la partie superficielle de l'encéphale (la matière grise plissée qui s'étend de la surface jusqu'à une profondeur de 2-3mm). Il est divisé en plusieurs régions affectées chacune à une tâche spécifique : les sens, la fonction motrice, etc. Le cortex visuel primaire (V1) se situe au niveau occipital de l'encéphale et est très développé chez l'homme : la densité de neurones y dépasse $10^6/\text{mm}^3$. Chaque neurone est connecté avec des neurones voisins (de l'ordre de quelques milliers de connexions) de manière « éparse » : seule une fraction du nombre de neurones à l'intérieur d'un voisinage centré sur un neurone donné sont connectés avec ce neurone.

Deux types de cellules nerveuses prédominent dans V1 [12], comme dans le reste du cortex d'une manière générale (cf. figure 2.8) :

- les **cellules pyramidales**, avec un soma conique ($>30\ \mu\text{m}$), des dendrites regroupées en deux arborescences distinctes (basales et apicales) et un axone pouvant se projeter dans d'autres parties du cerveau (neurones de sortie pour le cortex),
- les **cellules stellaires**, avec un soma sphérique ($<10\mu\text{m}$) et qui sont majoritairement des inter-neurones (i.e. qui sont connectés uniquement avec des neurones proches), qui reçoivent les influx et sont connectés aux cellules pyramidales; environ 20% de ces cellules sont des cellules inhibitrices et sont associées au NT GABA (dont les connexions sont plus lointaines), tandis que celles excitatrices ont des dendrites apicales, un court axone et leur soma est situé principalement dans la couche 4.

Les neurones pyramidaux sont impliqués plutôt dans une organisation verticale tandis que les neurones stellaires ont des connexions à l'intérieur d'une même couche ou avec les voisines immédiates. Il existe un grand nombre d'autres types de cellules (au moins 50 ont été identifiés dans le cortex humain), mais elles sont de loin minoritaires par rapport aux deux espèces citées ci-dessus.

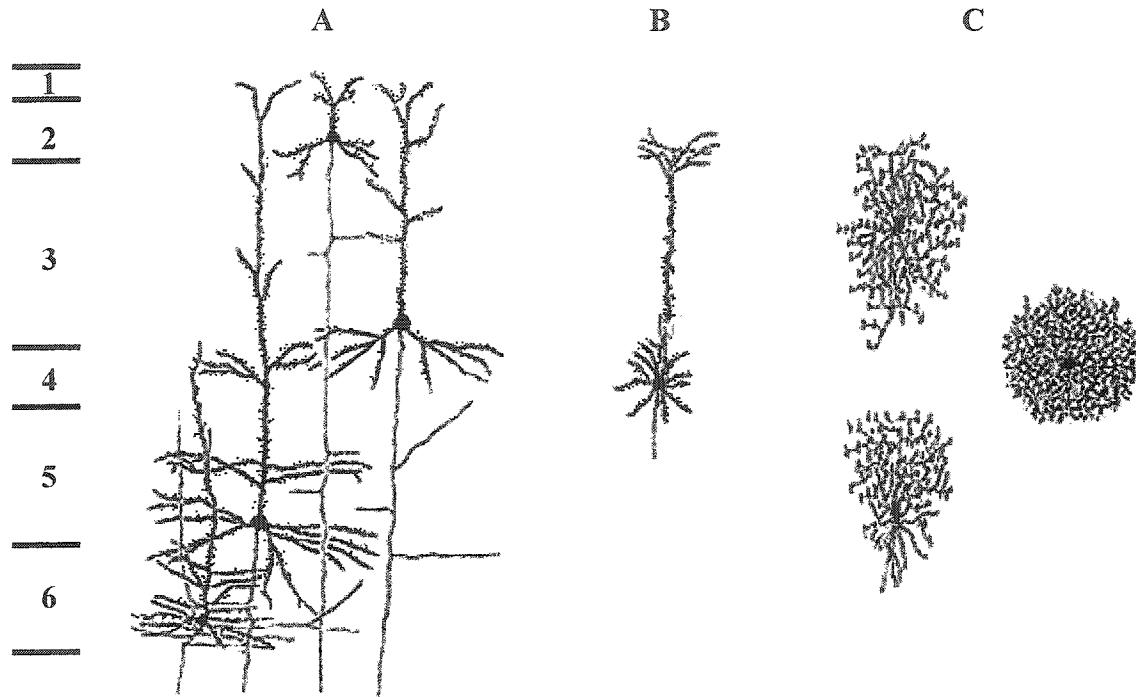


Figure 2.8 : Schéma des neurones pyramidaux et stellaires.

L'échelle à gauche figure les couches corticales (cf. figure 2.9). L'architecture des neurones pyramidaux (A) permet d'intégrer des informations provenant de plusieurs couches. Les cellules stellaires avec épines (B) ont un arbre dendritique apical qui se rend jusqu'à la surface corticale mais ses dendrites basales sont concentrées dans la couche 4. Enfin, les inter-neurones inhibiteurs de la couche 4 (C) sont connectés avec des cellules proches (couches 3, 4 et 5). À noter : les épines présentes sur les dendrites de neurones pyramidaux sont supposées intervenir dans des phénomènes d'apprentissage (la sensibilité synaptique serait capable de varier au niveau de ce type de connexions).

V1 se présente comme une succession de couches dans le sens de la profondeur qui peuvent être différenciées par imagerie médicale (cf. figure 2.9) :

- la 1^{ère} est la couche moléculaire (en surface), elle est relativement aneuronale, reçoit des influx non spécifiques à la vision et contient la partie supérieure des dendrites apicales des neurones pyramidaux,
- la 2^{ème} est la couche granulaire externe et contient des inter-neurones stellaires liés à des axones afférents non spécifiques,
- la 3^{ème} est la couche externe où se concentrent des cellules pyramidales dont les axones courts se projettent majoritairement dans les couches voisines, et la partie profonde de cette couche reçoit un influx important de la sous-couche 4C,
- la 4^{ème} est la couche granulaire interne, contenant majoritairement des inter-neurones stellaires; elle est très développée et peut se décomposer en plusieurs sous-couches :
 - 4A contient des cellules stellaires de petite taille,
 - 4B contient peu de neurones (cellules pyramidales se projetant en dehors de V1) et reçoit un influx important de 4C α (voie magnocellulaire),
 - 4C α et 4C β sont innervées par les radiations optiques émanant des CGL et correspondent respectivement aux voies parvocellulaire et magnocellulaire,
- la 5^{ème} est la couche interne où se concentrent des cellules pyramidales, avec des axones longs se projetant à l'extérieur de V1 et d'autres vers la 1^{ère} et la 3^{ème} couche (faisant ainsi des boucles récurrentes),
- la 6^{ème} est la couche multiforme et contient de nombreuses variétés de cellules (pas uniquement stellaires ou pyramidales) qui se projettent hors de V1, et ce niveau reçoit un influx des CGL et se projette en retour vers les CGL.

Ce système de couches est largement récurrent tant au sein d'une même couche (latéralement) qu'entre deux couches grâce aux cellules stellaires excitatrices à axones courts et aux dendrites apicales de neurones pyramidaux de la couche 5 par exemple. En particulier, les dendrites apicales pourraient jouer un rôle actif dans une forme supervisée ou renforcée d'apprentissage [28]. La puissance des circuits corticaux découle en grande partie de cette organisation massivement parallèle.

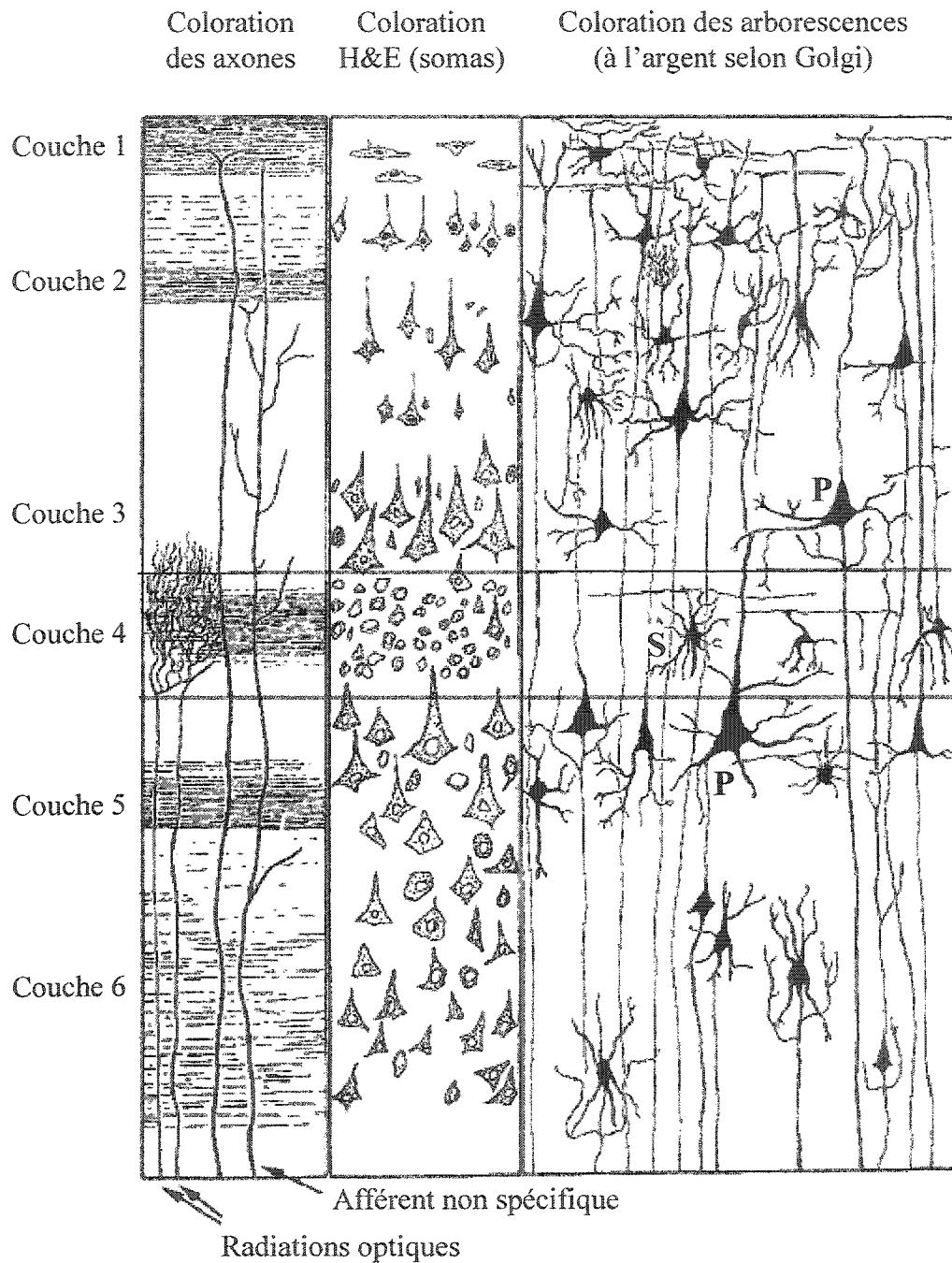


Figure 2.9 : Organisation du cortex visuel primaire. (vanat.cvm.umn.edu)
 Les 3 dessins correspondent à des techniques d'imagerie distinctes. On voit des neurones stellaires (S) au niveau de la couche 4, où se terminent les radiations optiques. Les neurones pyramidaux (P) ont des dendrites basales dans la même couche que leur soma, des dendrites apicales qui remontent jusqu'à la couche 1 et des axones qui peuvent se projeter dans d'autres régions du cerveau.

V1 possède une structure spéciale qui a été mise en évidence par les travaux de Hubel et Wiesel en 1962 (prix Nobel en 1981) [15] : des **colonnes** verticales traversant toutes les couches dont les neurones correspondent à une **portion du champ visuel**, une **orientation préférentielle** (cf. figure 2.10) et une **dominance oculaire**. Les variations de dominance oculaire se font par bandes de largeur $50\mu\text{m}$ maximum, tandis que celles d'orientation préférentielle correspondent à des largeurs de l'ordre de $400\mu\text{m}$. Environ un tiers des cellules de V1 présentent une orientation préférentielle très marquée. En vue de dessus, la surface corticale s'apparente ainsi à une carte topographique de colonnes juxtaposées (cf. figure 2.11). Les inter-neurones stellaires et notamment les connexions latérales inhibitrices au sein des couches granulaires 4 relatives au NT GABA jouent un rôle prépondérant dans cette auto-structuration du cortex [27].

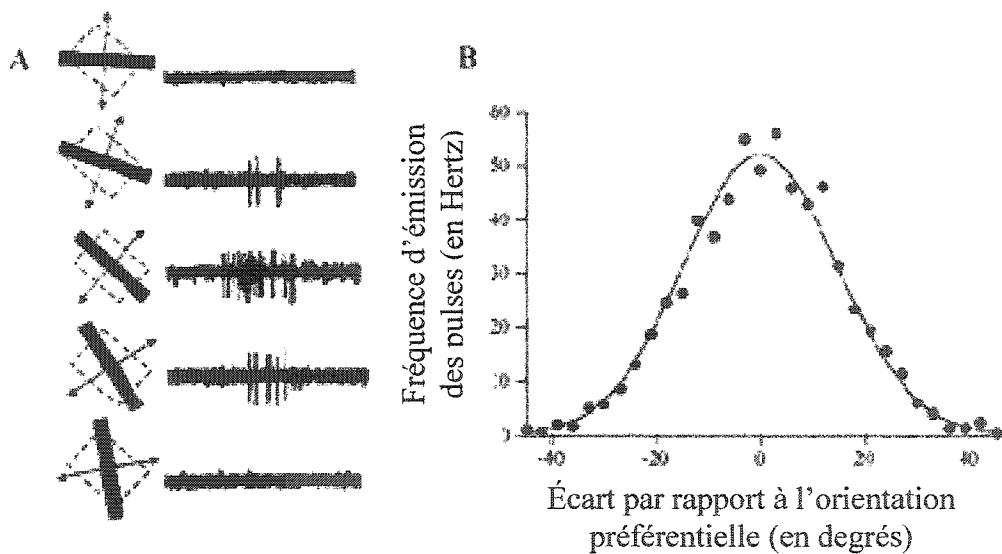


Figure 2.10 : Courbe physiologique de la réponse d'un neurone en fonction de l'orientation présentée.

(white.stanford.edu/~heeger/psych202/lecture-notes/visual-cortex/visual-cortex.html)

La variation de l'activité d'un neurone (B) dans une colonne corticale est de type gaussien autour d'une orientation préférentielle. À gauche (A) sont présentés des exemples de trains de pulses en réponse à diverses orientations (l'orientation préférentielle est donnée par la grande dimension du rectangle en pointillés).

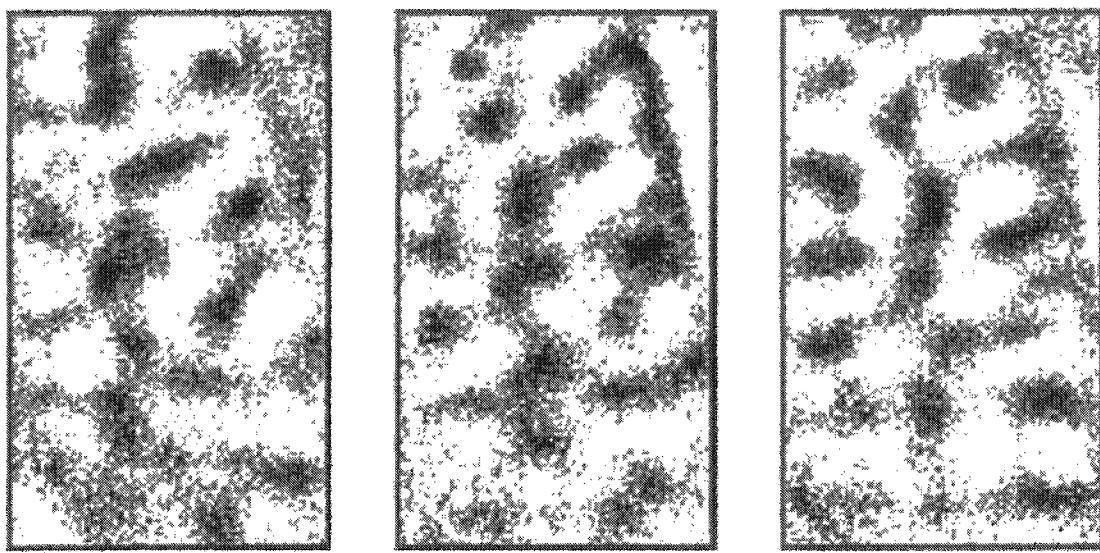


Figure 2.11 : Carte topographique des colonnes corticales. (ni.cs.tu-berlin.de)
 Ces cartes topographiques correspondent aux activités des colonnes corticales chez un macaque suite à une stimulation par un segment lumineux d'orientation variable (de gauche à droite, 0°, 45° et 90°). Les zones claires correspondent aux activités élevées.

2.1.5 Codage de l'information visuelle par les potentiels d'action

Du point de vue du traitement de l'information nerveuse, un neurone peut être considéré comme une unité dynamique traitant des événements brefs. L'information nerveuse est ainsi codée dans les séquences temporelles de PA et elle est répartie sur l'ensemble du réseau. En effet, un neurone seul ne semble pas coder une connaissance en particulier, même si son activité peut être fortement corrélée avec des caractéristiques d'un stimulus particulier. C'est plutôt la structure fine des **microcircuits neuronaux** ainsi que les types de neurones impliqués (multipolaires, pyramidal, stellaire, etc.) et de synapses dynamiques (les cinétiques relatives à chaque NT engendrent des dynamiques spécifiques) qui régissent et caractérisent le traitement de l'information. Par exemple, l'orientation préférentielle d'une colonne se traduit au niveau de tous ses neurones et met en jeu des phénomènes de synchronisation entre eux. De nombreux mécanismes entrent en jeu dans le traitement des trains de PA : intégration temporelle des potentiels post-

synaptiques au niveau du soma, période réfractaire, délai et sensibilité dynamique des synapses. Aussi est-il difficile de découvrir des informations pertinentes parmi l'activité d'un groupe de neurones, qui peut sembler à priori peu différent du bruit.

Dans le cas du cortex visuel, la structure fine s'organise durant les premiers jours après la naissance et dépend en grande partie de l'environnement visuel qui est vu : si par exemple on présente uniquement certaines orientations particulières à des chatons durant cette période, ils présentent des troubles pour reconnaître les autres orientations par la suite [27]. L'architecture corticale qui se met alors en place semble déterminante pour les types de signaux visuels analysables : on suppose que l'auto-organisation mise en évidence par Hubel et Wiesel relève de l'ajustement des paramètres synaptiques dans les voies visuelles préalablement structurées.

Le codage d'une représentation de la scène visuelle n'est que très partiellement connu pour l'instant. Globalement, les connaissances se limitent à des corrélations entre les densités d'activité de cellules pour des composantes visuelles comme l'orientation de barres mobiles comme montré dans la figure 2.10. De récentes études ont montré que les seules **densités de PA** émis par un neurone ne suffisaient pas à rendre compte d'une représentation convenable du contraste au niveau cortical : les variations des densités chez des neurones spécialisés dans le contraste ne sont pas compatibles avec un codage significatif. Par contre, les motifs formés par les PA (en particulier la structure des **périodes inter-pulses**) se révèlent beaucoup plus significatifs et la précision temporelle intervenant dans la synchronisation entre neurones est de l'ordre de la milliseconde [22].

On doit choisir le niveau de précision que l'on adopte pour analyser des trains de pulses : si l'on se restreint aux densités de pulses, le modèle est plus aisément gérable, mais il se peut que les résultats obtenus soient incohérents avec une représentation valable de l'information visuelle. Cependant, comme on ne connaît pas la nature du codage, on ne peut construire un estimateur de la similarité de deux trains de pulses dont on aurait besoin pour tester si deux protocoles de stimulation induisent des scènes visuelles semblables, par exemple. La comparaison des densités de PA émis par les neurones n'est qu'un estimateur partiel, aussi faut-il étudier si elle peut être utile dans notre cas.

2.2 Les modèles de neurones HH, I&E et sigmoïdaux

Il existe plusieurs modèles « mathématiques » de neurones, qui rendent compte plus ou moins finement des aspects des neurones biologiques décrits dans la partie 2.1 : le modèle de Hodgkin-Huxley (HH), les neurones à pulses Intégrer-&-Émettre (I&E) et les neurones artificiels « sigmoïdaux » (par ordre décroissant de réalisme).

2.2.1 Le modèle de Hodgkin-Huxley

Hodgkin et Huxley ont développé un modèle de la membrane neuronale à partir d'expériences sur des axones géants de calmars [10] et ils ont reçu avec Eccles le prix Nobel de médecine en 1963. Leur modèle se base sur les **courants ioniques** relatifs à K⁺ et Na⁺ (plus un courant de fuite) pour calculer le **potentiel transmembranaire** (cf. équations 2.2 et figure 2.12).

$$\begin{aligned}
 I_m &= I_{ionique} + C_m \frac{\partial E}{\partial t} \\
 I_{ionique} &= m^3 h G_{Na}(E - E_{Na}) + n^4 G_K(E - E_K) + G_F(E - E_F) \\
 \frac{\partial m}{\partial t} &= \frac{m_{inf} - m}{T_m} \quad \frac{\partial h}{\partial t} = \frac{h_{inf} - h}{T_h} \quad \frac{\partial n}{\partial t} = \frac{n_{inf} - n}{T_n}
 \end{aligned}$$

Équations 2.2 : Formules du modèle HH.

I_m est le courant total sortant à travers la membrane, $I_{ionique}$ le courant ionique total, E le potentiel transmembranaire et C_m est la capacité équivalente de la fibre nerveuse. Les trois courants ioniques sont : Na pour le sodium, K pour le potassium et F pour un courant de fuite. Les G sont les conductances maximales pour chaque courant, et n , m et h sont des variables sans dimension régies par des équations différentielles (les valeurs indiquées par _{inf} correspondent aux valeurs à l'équilibre et les T à des durées caractéristiques de variation).

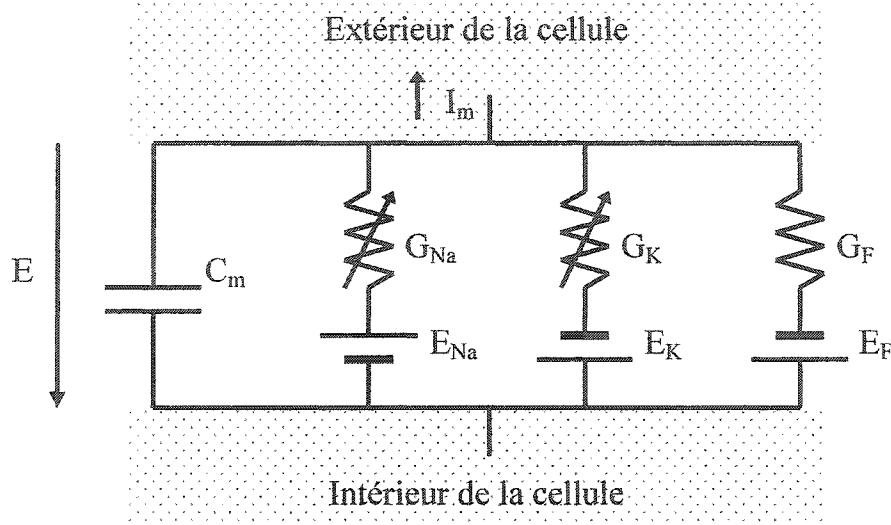


Figure 2.12 : Circuit équivalent et formule principale du modèle de Hodgkin-Huxley. Les notations sont les mêmes que dans l'équation 2.2. Cet élément peut être vu comme un élément de base d'un câble le long duquel on peut calculer le potentiel E , de proche en proche.

Le modèle HH rend compte de nombreux phénomènes biologiques couramment observés : seuil d'activation (cf. figure 2.13), période réfractaire successive à l'émission d'un PA, conduction spontanée unidirectionnelle d'un PA par exemple. Même s'il a été amélioré et dépassé depuis 1963, il constitue la base de quasiment tous les modèles de membranes neuronales. Il permet de construire un neurone avec une structure 3D fine représentant sa membrane. Les variations locales des propriétés géométriques et physiologiques (i.e. dimensions physiques, résistivité, etc.) et des paramètres entrant en compte dans les équations 2.2 influent alors sur la conduction du potentiel. Le logiciel NEURON permet une utilisation simple de ce modèle : à partir de sections cylindriques, on définit une architecture 3D ou symbolique d'un neurone et permet ensuite de simuler son comportement en appliquant la formule ci-dessus comme une équation de câble qui permet de calculer la propagation des variations du potentiel le long de la membrane modélisée (cf. partie 2.5).

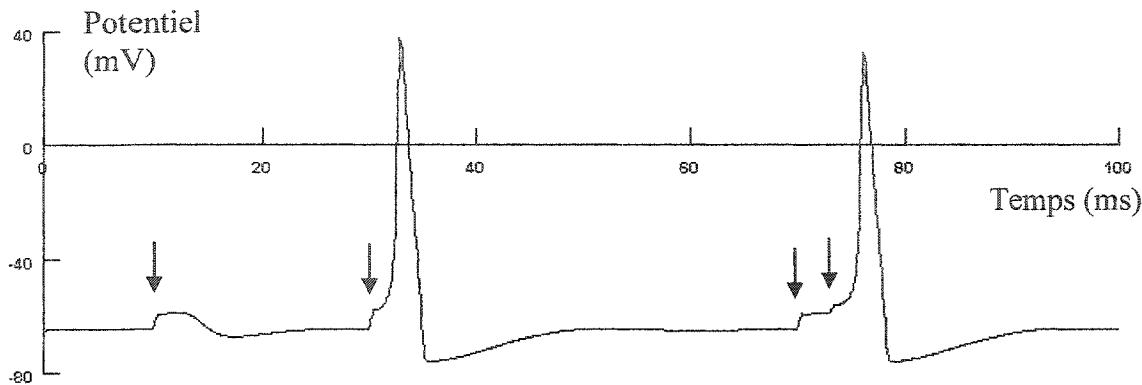


Figure 2.13 : Évolution du potentiel transmembranaire d'une section de câble selon le modèle HH lors de stimulations intracellulaires en courant.

On injecte du courant au moyen d'un modèle d'électrode intracellulaire de NEURON, avec des pulses de courant d'une durée de 0.5ms et d'intensité variable (indiquées par les flèches en trait plein). Le seuil pour cette section seule est entre 20 et 21nA. Le graphique montre une stimulation sous-seuil d'amplitude 20nA à $t=10\text{ms}$ qui ne déclenche rien, une stimulation sur-seuil de 25nA à $t=30\text{ms}$ qui déclenche un PA, et deux stimulations sous-seuil consécutives de 20nA à $t=70\text{ms}$ et de 11nA à $t=73\text{ms}$ qui déclenchent un PA. Ces phénomènes sont conformes à la réalité biologique.

Dans le cadre d'une reconstitution 3D fine d'un tissu, il est possible de modéliser les neurites individuellement de manière à obtenir des connexions synaptiques réalistes, que ce soit au niveau de la géométrie ou des mécanismes synaptiques liés aux NT. On peut aussi construire des modèles d'électrodes tenant compte des anisotropies du milieu, où l'effet sur les neurones voisins n'est pas simpliste. Ainsi, ce modèle est suffisamment souple pour intégrer pratiquement n'importe quel aspect du tissu biologique ou des électrodes. L'approximation faite par rapport à la réalité consiste en pratique à effectuer les calculs dans un espace unidimensionnel (à la manière d'équations de câbles). On est alors limité uniquement par la puissance de calcul nécessaire à faire tourner ce modèle, qui est assez gourmand de ce point de vue.

2.2.2 Les neurones à pulses I&E

Un neurone I&E (integrate-&-fire en anglais) consiste en une unité simplifiée qui gère des pulses modélisant les PA. On peut les représenter avec les mêmes neurites que des neurones biologiques (cf. figure 2.14). La figure 2.15 donne une illustration du fonctionnement en temps discret d'un tel neurone avec une seule entrée sur une synapse.

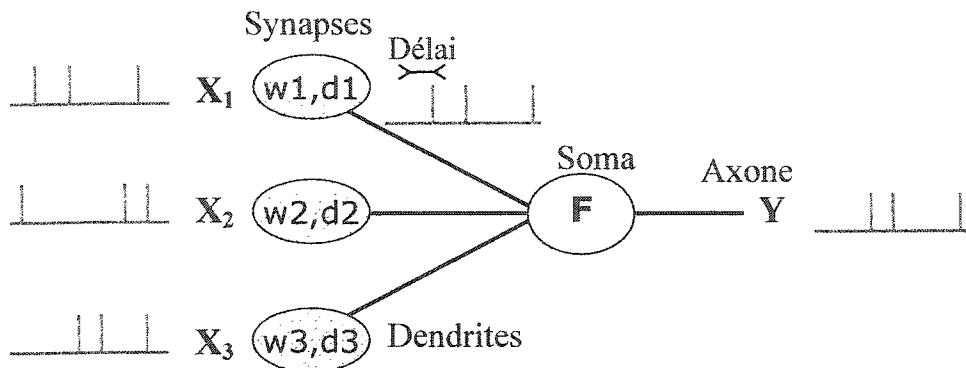


Figure 2.14 : Schéma fonctionnel d'un neurone I&E.

Les X_i sont les entrées aux synapses situées sur les dendrites et Y est la sortie à l'axone, ce sont des trains de pulses (vecteurs binaires indexés par le temps). F est la fonction interne et les grandeurs w_i et d_i sont respectivement les poids et délais relatifs à chaque synapse. La sortie est commandée par la fonction F (cf. équations 2.3).

si $F(t) > 1$, alors $Y(t) = 1$ et $\forall u \in (t, t + T_r], Y(u) = 0$ et $F(u) = 0$

si $F(t) \leq 1$, alors $F(t + \delta t) = \mu F(t) + \sum_i \omega_i X_i(t - d_i)$

Équations 2.3 : Formules régissant le comportement d'un neurone I&E.

Dans ces formules, δt correspond au pas temporel de la simulation et μ est relié au comportement de fuite. La fonction interne effectue l'intégration temporelle des événements (chaque pulse a pour valeur 1) pondérés par les poids synaptiques en tenant compte des délais synaptiques. Lorsqu'elle dépasse un certain seuil, le neurone décharge et reste muet pendant une période réfractaire. Lorsque aucun événement n'est reçu, la fonction interne décroît vers 0 (comportement de fuite), ce qui modélise un courant post-synaptique exponentiellement décroissant.

Les entrées et sorties de ces neurones sont donc des séries temporelles binaires et les réseaux sont construits en connectant les neurones entre un axone et une synapse. La version de neurone I&E présentée ici est relativement simple : des raffinements sont possibles pour inclure d'autres formules du courant post-synaptique; on peut aussi donner un caractère aléatoire au neurone en émettant des pulses en sortie suivant une probabilité fonction de la valeur de F . Bien que plus simple que le modèle HH, les neurones I&E sont très puissants du point de vue computationnel [20].

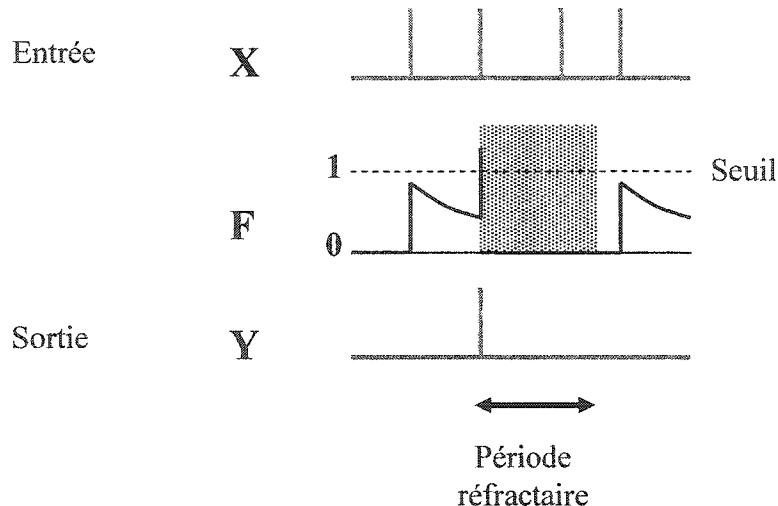


Figure 2.15 : Exemple de fonctionnement d'un neurone I&E.

Ce neurone possède une seule synapse. On remarque que la première pulse fait augmenter la fonction interne F en deçà du seuil, puis la deuxième déclenche l'émission d'un pulse sur l'axone. La troisième pulse intervient durant la période réfractaire et est ignorée. Avec la quatrième, on reprend le fonctionnement normal comme au début.

Ce modèle rend compte des aspects importants du comportement d'un neurone biologique : gestion individuelle des pulses, période réfractaire après émission d'une pulse, retour au « potentiel » de repos après une stimulation sous-seuil (comportement de fuite) et un modèle simple de courant post-synaptique.

2.2.3 Les neurones « sigmoïdaux »

Ces neurones peuvent être décomposés en 3 parties correspondant aux neurites des neurones biologiques (cf. figure 2.16) : le noyau (ou soma) d'un côté relié à un arbre dendritique et de l'autre à l'axone, avec des synapses en bout des dendrites du côté opposé à la connexion avec le soma.

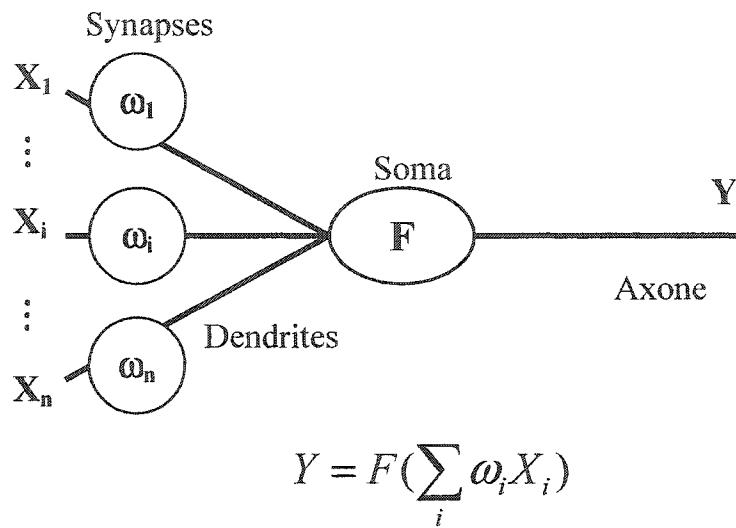


Figure 2.16 : Description d'un neurone « sigmoïdal ».

Équations 2.4 : Formules régissant le comportement d'un neurone « sigmoïdal ».
Les X_i sont les n entrées, ω les poids synaptiques, Y la sortie et F une fonction réelle.

Un neurone « sigmoïdal » traite des nombres réels : la sortie est une fonction de la somme des entrées pondérées par les poids synaptiques. Cette fonction est en général une fonction réelle, continue, dérivable, monotone et bornée (souvent une sigmoïde, d'où le nom donné ici). Les connexions entre de tels neurones se font de l'axone d'une cellule-source avec une « synapse » sur une dendrite d'une cellule-cible. Ce modèle de neurone est avant tout un outil mathématique et n'est pas destiné à la modélisation biologique à priori. Même si rien n'empêche à priori d'utiliser des séries temporelles de réels avec des délais synaptiques, ces neurones sont généralement utilisés dans les réseaux de neurones à propagation avant (RNPA) avec des entrées statiques. Ces réseaux possèdent une

architecture unidirectionnelle et ils sont associés à un algorithme d'entraînement des poids synaptiques appelé rétro-propagation de l'erreur (RPE) [25] qui permet à la sortie du réseau d'approximer n'importe quelle fonction mathématique réelle (cf. partie 3.1.1).

Pour faire l'analogie biologique, un tel modèle de neurone traite des densités de pulses (et non des pulses individuellement), et l'on a exclu de se restreindre aux densités de pulses compte tenu que l'on ne veut pas représenter une scène visuelle. Si l'on voulait les utiliser à des fins de modélisation, on devrait donc utiliser d'autres architectures que les RNPA qui ne peuvent rendre compte de dynamiques complexes tels que celles de boucles de rétroactions. Des modifications de leur architecture ont été proposées pour inclure le temps, mais le réseau n'est que localement récurrent pour que la RPE reste applicable [21]. Il s'ensuit que ce modèle ne semble pas adapté à la modélisation biologique, car il correspond à une version trop simpliste d'un neurone biologique.

2.3 Les réseaux neuronaux récurrents à pulses (RNRP)

Les informations physiologiques énoncées dans la partie 2.1 nous incitent à utiliser des réseaux de neurones pour modéliser le traitement de l'information visuelle dans le tissu cortical. En effet, le modèle doit être capable de traiter des trains de pulses en tant que tels pour permettre à terme de tester des protocoles de stimulation. Il servira également d'outil d'investigation du fonctionnement du cortex, en parallèle à des expériences *in vivo* lorsque l'on testera l'implant sur des sujets.

On va se restreindre au moins dans ce mémoire au modèle de neurones à pulses I&É : il rend compte des principaux aspects du fonctionnement neuronal et on espère qu'il pourra permettre de construire des réseaux suffisamment puissants pour modéliser des phénomènes complexes tels que l'auto-organisation ou la reconnaissance de motifs de stimulation. Il pourra par contre être intéressant à terme d'utiliser des neurones suivant le modèle HH pour raffiner le modèle avec un degré de réalisme supplémentaire. Tant que l'on s'intéresse à l'activité du réseau, on peut considérer ces modèles comme

quasiment équivalents pour tester des algorithmes d'apprentissage, le modèle I&É étant cependant plus rapide. Les neurones « sigmoïdaux » sont écartés car leur intérêt réside essentiellement dans leur utilisation mathématique au sein de RNPA et qu'ils constituent de piètres modèles de neurones biologiques.

On va utiliser dans le chapitre 5 des synapses dynamiques telles que définies dans la partie 2.1.2, avec un délai et un poids (sensibilité) synaptiques, et des temps caractéristiques de facilitation et de dépression (cf. équation 2.1). Une synapse inhibitrice correspond à un poids négatif dans le cas des neurones I&É. Les paramètres gérant les caractéristiques des neurones et des synapses, ainsi que les densités de connexion des neurones et les types des synapses (inhibitrices ou excitatrices) s'inspirent de **données physiologiques** dans la mesure du possible pour définir une architecture aussi réaliste que possible du réseau (ce qui constitue un premier objectif à remplir pour les RNRP « biologiques », cf. partie 2.4). Dans le cas où l'on ne trouve pas de référence physiologique, on essaie de trouver des ordres de grandeur cohérents qui induisent des comportements plausibles.

La question reste en suspens de savoir jusqu'où pousser la complexité et le détail du modèle du neurone qui sert de base au réseau : on ne connaît pas à priori dans quelle mesure cela influe sur le comportement et la puissance du réseau construit. On va définir les objectifs sur le comportement et l'architecture du modèle dans la partie suivante et on étudiera comment **optimiser les paramètres synaptiques** pour tenter de les réaliser, sachant que le caractère réaliste que l'on veut donner à l'architecture constitue une contrainte forte sur les réseaux construits. À noter que tous les mécanismes impliqués dans les réseaux construits ici sont déterministes.

2.4 Récapitulatif des objectifs du modèle

On définit ici 4 objectifs que le réseau doit valider afin qu'on puisse le considérer comme biologiquement réaliste. Ces objectifs (qui n'incluent pas les considérations sur

les types de neurones et de synapses utilisés) sont énoncés du plus « simple » au plus « complexe ».

1) Son architecture s'inspire de la physiologie corticale, c'est-à-dire que l'on prend en compte un grand nombre de neurones (avec des caractéristiques internes réalistes), de nombreuses connexions « éparses » (avec une partie des voisins seulement) pour obtenir une densité plausible (le réseau est donc massivement récurrent). Dans un premier temps, on se contente de modéliser uniquement les deux types de neurones prédominants (stellaires et pyramidaux) en considérant les premiers comme responsables de l'organisation des connexions latérales dans le cortex et les seconds comme des sorties. On peut représenter le tissu comme une fine couche (i.e. la couche 4) dans laquelle on peut déterminer des neurones excitateurs et inhibiteurs.

2) Le comportement doit être globalement réaliste dans le sens où il ne doit pas être en saturation lors de la stimulation et le changement entre l'activité au repos et durant la stimulation doit être significative. On peut définir globalement pour les paramètres synaptiques des distributions gaussiennes adéquates dont l'importance relative des moyennes doit être ajustée pour obtenir le résultat escompté.

3) Conformément à l'organisation du cortex en colonnes décrite dans les travaux de Hubel et Wiesel, il doit s'auto-structurer de manière non supervisée en fonction des signaux reçus et rendre compte de colonnes d'orientation (pour l'instant, on peut se focaliser sur ce point et on écarte la dominance oculaire; la correspondance entre la carte corticale et la topographie du champ visuel dépend en plus d'un modèle de rétine et des liaisons entre le réseau et cette rétine). Cela se traduit lors d'une stimulation par des densités d'activité élevées dans les zones correspondant au type de stimulus (orientation, etc.). Ce phénomène doit se baser sur l'adaptation des connexions synaptiques latérales entre des inter-neurones stellaires à l'intérieur de la couche 4 en particulier.

4) Enfin, le RNP doit être fonctionnel, i.e. pour certains neurones accomplir des tâches spécifiques, tel que reconnaître des motifs parmi les signaux reçus en entrée, dans l'optique de tester des protocoles de stimulation associés à des scènes visuelles à

terme. On peut prendre ces neurones (pyramidaux) comme des sorties et les entraîner de manière supervisée.

Les deux derniers points consistent en des tâches complexes que l'on veut intégrer dans notre modèle en utilisant des algorithmes d'apprentissage pour ajuster les paramètres de certaines connexions synaptiques.

2.5 Le logiciel NEURON

NEURON est un logiciel libre développé par M. Hines, J.W. Moore et T. Carnevale des universités Yale et Duke aux États-Unis qui permet de simuler des réseaux de neurones [14]. Il existe sous plusieurs versions (Windows, Linux et MacOS) et est encore en évolution actuellement (dernière version 5.4 disponible sur le serveur). Il est dérivé du C et optimisé pour effectuer les calculs durant la simulation, il possède une interface graphique permettant de gérer une grande partie de ses fonctionnalités et exécute également des scripts (.hoc).

NEURON inclut notamment les modèles I&E et HH dont les paramètres sont configurables et modifiables durant la simulation, ainsi que différents modèles de synapses correspondant à chaque type de neurones utilisés. Un neurone selon le modèle de HH se « construit » de la manière suivante : on définit des sections de câble dont les membranes obéissent aux lois du modèle HH par exemple, on ajuste les dimensions de ces sections et on les connecte pour former une structure géométrique en 3-D (cf. figure 2.17). Cette structure géométrique influe sur le calcul du potentiel au travers des dimensions physiques, de la résistivité des sections, etc. En revanche, les angles n'interviennent pas dans le calcul, et on peut ramener toute géométrie 3D à une structure plane équivalente. On dispose des « synapses » (correspondant en fait à des récepteurs synaptiques) sur la structure 3-D et on peut connecter n'importe quelle section d'un neurone à une synapse d'un autre neurone. Habituellement, on connecte des sections à l'extrémité de l'arbre axonique (figurant les boutons synaptiques) de la cellule-source

avec ces synapses situées sur le soma ou l'arbre dendritique de la cellule-cible. Les synapses utilisées pour les neurones HH modélisent un potentiel post-synaptique lorsque le potentiel de la section-source dépasse un seuil fixé.

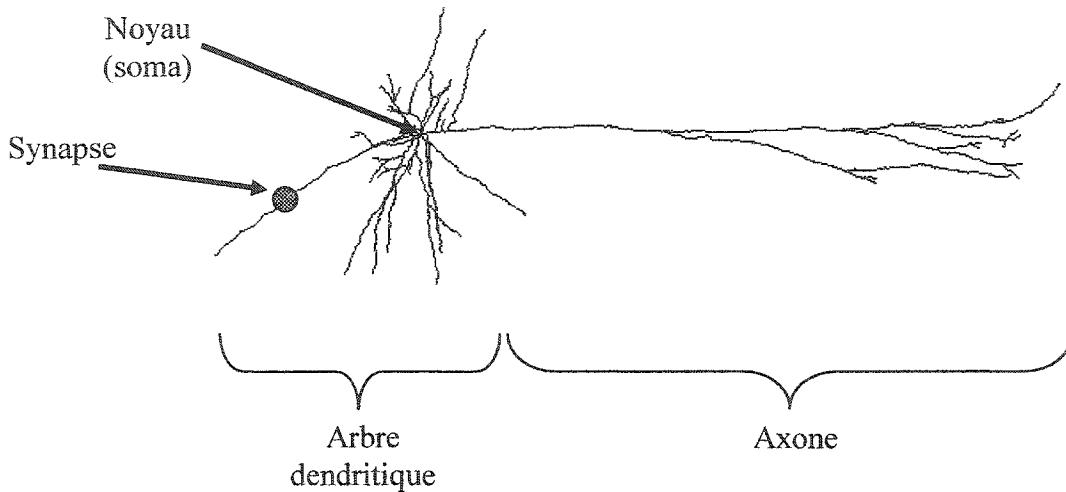


Figure 2.17 : Exemple de reconstruction d'une structure 3-D pour un neurone pyramidal du cortex visuel primaire chez l'homme. [14]
La géométrie du neurone est reconstruite avec des sections de câbles mises bout à bout.

Cette approche est intéressante car les calculs sont ramenés à une dimension du fait que l'on considère les équations du modèle HH sur des **sections de câble**. Les calculs se font en temps discret et chaque section dans le cas du modèle HH est divisée en segments sur lesquels on linéarise les formules de HH. Cette approximation a été validée par de nombreuses études sur la reconstruction fine de neurones réels. D'autres travaux ont également raffiné le modèle en incluant d'autres ions comme ceux du calcium ou d'autres mécanismes tels que la présence de myéline dur un axone.

Tous les paramètres des modèles sont accessibles et modifiables durant la simulation et de nombreux outils sont disponibles pour l'analyse du comportement des réseaux, en particulier au niveau affichage : graphiques du potentiel pour voir apparaître les PA (cf. figure 2.6), cartes d'activité singeant un électro-encéphalogramme, etc.

On a donc vu dans ce chapitre les types de réseaux que l'on va utiliser afin de construire un modèle réalisant 4 objectifs de « réalisme biologique », et brièvement le logiciel utilisé pour les simulations. Dans le chapitre suivant, on va s'intéresser à l'ajustement des paramètres synaptiques qui déterminent leur comportement grâce à des algorithmes d'entraînement.

3 Algorithmes d'apprentissage appliqués aux RNRP

Une fois l'architecture d'un RNRP définie, on va devoir ajuster ses paramètres synaptiques pour obtenir un comportement « biologiquement » réaliste. En effet, on ne peut espérer définir les valeurs de ces paramètres par des mesures expérimentales, étant donné le nombre de connections dans le tissu réel d'une part, et qu'elles sont variables dans le temps et difficiles à mesurer d'autre part. Une solution est d'entraîner les RNRP grâce à des algorithmes d'apprentissage, sauf qu'il n'existe pas de méthode classique spécifique aux réseaux globalement récurrents qui permette d'ajuster de manière supervisée les activités de ses neurones. On présente donc d'abord deux algorithmes classiques et les architectures qui correspondent : la rétro-propagation pour les réseaux neuronaux à propagation avant (RNPA), puis les réseaux de Hopfield entraînés avec une formule dérivée de celle de Hebb; et on analyse pourquoi ces algorithmes sont inadaptés dans notre cas et en quoi notre problème d'optimisation n'est pas classique. On expose ensuite deux méthodes originales distinctes dans leurs philosophies et comment elles peuvent conduire à la construction du modèle de tissu biologique.

3.1 Algorithmes classiques d'entraînement de réseaux neuronaux

3.1.1 La rétro-propagation pour les RNPA

Les RNPA sont des réseaux orientés, c'est-à-dire qu'ils sont formés de plusieurs couches de neurones où chacune n'a de connections qu'avec la suivante immédiate. Originellement, un RNPA comporte 3 couches : une d'entrées, une intermédiaire (ou

cachée) et une de sorties (cf. figure 3.1). Un neurone calcule la somme pondérée de ses entrées et renvoie en sortie la valeur de sa fonction d'activation correspondant à cette somme (cf. équation 3.1).

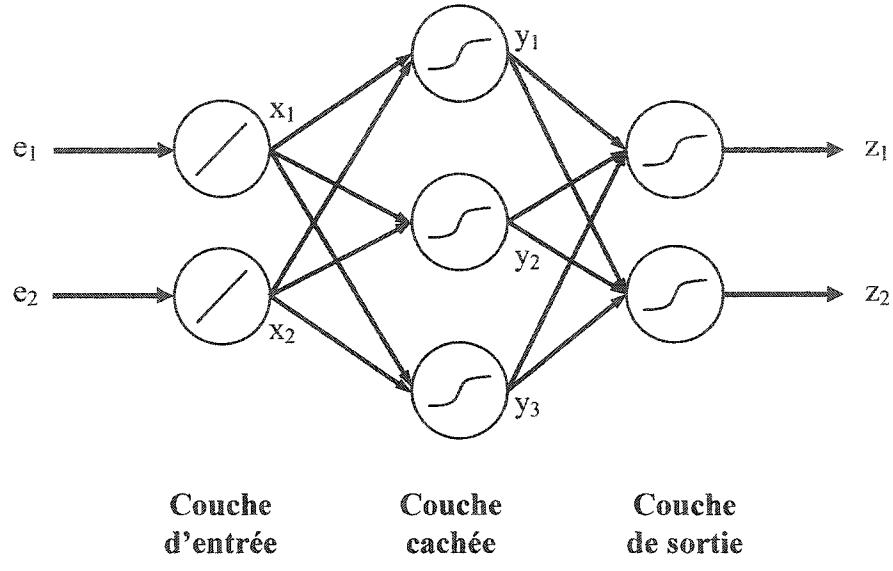


Figure 3.1 : Architecture d'un RNPA à une seule couche cachée.

Ce RNPA possède deux neurones d'entrée, trois dans la couche cachée et deux de sortie. Ici, les entrées du réseau sont e_1 et e_2 ; x , y et z correspondent respectivement aux sorties des neurones de la couche d'entrée, cachée et de sortie. Chaque connexion entre deux neurones possède un poids (non représenté ici) intervenant dans la formule du calcul de la sortie du neurone cible (cf. équation 3.1). Les fonctions d'activation de chaque neurone (représentées dans les « corps » des neurones) ne sont pas forcément les mêmes, mais elles sont généralement identiques à l'intérieur d'une couche.

$$y = f \left(\sum_i \omega_i \times x_i \right)$$

Équation 3.1 : Fonction d'activation d'un neurone.

Les x_i correspondent aux entrées du neurone, ω_i aux poids synaptiques relatifs aux entrées et y est sa sortie (nombres réels). Dans la figure 3.1, f est une sigmoïde pour les neurones intermédiaires et de sortie, et la fonction-identité pour la couche d'entrées.

La rétro-propagation consiste en une descente de gradient sur une fonction d'erreur dérivable calculée sur les nœuds de sortie afin d'approximer une fonction mathématique sur les entrées. On calcule les modifications pour chaque poids du réseau

grâce à la dérivée de la fonction d'erreur entre le vecteur de sortie obtenu lorsqu'on présente un vecteur d'entrées (vecteur-résultat) et celui correspondant à la fonction à approximer (vecteur-objectif) (cf. équations 3.2). Le calcul de cette dérivée nécessite que le réseau soit orienté. On applique ainsi à chaque époque un vecteur d'entrée pris dans un ensemble d'entraînement et on modifie les poids de plus en plus faiblement afin qu'ils se stabilisent dans le réseau et que la fonction en sortie du RNPA converge vers la fonction-objectif (lorsque l'entraînement réussit).

$$\varepsilon = \frac{1}{2} \sum_k \xi_k^2 = \frac{1}{2} \sum_k (t_k - z_k)^2$$

$$\partial \omega_{jk} = \eta \times \xi_k \times y_j \times f'(net_k)$$

$$\partial \omega_{ij} = \eta \times x_i \times f'(net_j) \times \sum_k (\xi_k \times \omega_{jk} \times f'(net_k))$$

$$\text{avec } net_k = \sum_j \omega_{jk} y_j \text{ et } net_j = \sum_i \omega_{ij} x_i$$

Équations 3.2 : Formules de descente du gradient avec une erreur quadratique. [25]
En haut, la fonction d'erreur quadratique ε est égale à la somme des termes ξ_k calculés à partir des vecteurs résultat z et objectif t . La première formule de mise à jour des poids s'applique à la couche de sortie et la seconde à la couche cachée (cas d'un réseau à une seule couche cachée avec la même fonction d'activation f). Les indices i , j et k et les variables x , y et z , correspondent respectivement aux couches d'entrée, cachée et de sortie; les ω sont les poids « synaptiques » d'une couche à l'autre (cf. figure 3.1). Le paramètre η est le facteur d'apprentissage qui diminue durant l'entraînement.

On peut rajouter des couches intermédiaires pour réduire le nombre total de neurones et conserver une bonne capacité d'approximation (les formules des équations 3.2 se généralisent). À la fin de l'entraînement, on calcule une **erreur d'entraînement** comme la moyenne de l'erreur sur l'ensemble d'entraînement pour voir si l'optimisation a été efficace, et également une **erreur de validation** avec de nouvelles entrées pour évaluer la capacité de généralisation du RNPA. Si l'ensemble d'entraînement est trop petit ou si l'optimisation est trop longue, le RNPA peut être surentraîné : il est alors très bien adapté à l'ensemble d'entraînement mais son erreur de validation est mauvaise. Il a

été montré que la taille minimale de l'ensemble d'entraînement nécessaire pour obtenir habituellement une bonne généralisation du réseau était égale à 30 fois le nombre de poids à optimiser, et qu'augmenter la taille n'apportait pas d'amélioration notable [1].

Ces réseaux de neurones sont destinés à la base à l'approximation de fonctions mathématiques avec comme entrées et sorties des nombres réels. On peut remplacer les grandeurs scalaires en entrée et sortie par des vecteurs indexés par le temps, et effectuer éventuellement un prétraitement sur les entrées pour gagner en efficacité, tout en gardant la capacité d'appliquer la rétro-propagation [21]. Du point de vue de la modélisation biologique, on peut alors concevoir le réseau comme une arborescence orientée de neurones transmettant des densités d'activité.

3.1.2 Les réseaux de Hopfield et l'apprentissage de Hebb

Les réseaux de Hopfield sont complètement récurrents : chaque neurone est connecté à tous les autres, reçoit une entrée et constitue une sortie (cf. figure 3.2). La fonction d'activation de chaque neurone est une fonction-seuil; les entrées sont des vecteurs binaires (autant d'élément qu'il y a de neurones dans le réseau) et les sorties du réseau sont des séries temporelles binaires.

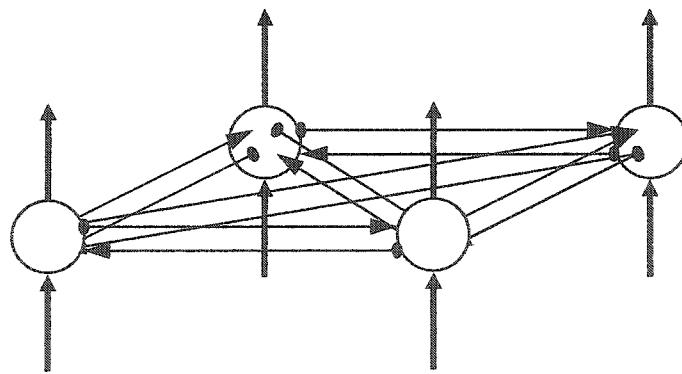


Figure 3.2 : Architecture d'un réseau de Hopfield.

Tous les neurones sont interconnectés (ici, ils sont représentés sur un plan horizontal en 3D et les flèches à bout arrondi dans ce plan symbolisent ces connexions) et sont des entrées et des sorties (les flèches en gras).

On applique à chaque neurone l'élément correspondant du vecteur d'entrée et on laisse le réseau se stabiliser. Ces réseaux sont spécialisés dans la reconnaissance de motifs (vecteurs binaires) : des motifs de référence sont associés à des états stables du système, et la présentation d'un motif doit conduire à un des états stables, ce qui permet de classifier le motif présenté. Cette dynamique est due à la structure spéciale du réseau, et équivaut à trouver le minimum d'une fonction d'énergie. En pratique, les états sont stables si le nombre de motifs à mémoriser est petit par rapport au nombre de neurones et si les motifs de référence ne sont pas trop corrélés entre eux [11].

Les poids synaptiques du réseau déterminent les états stables : ils peuvent être ajustés durant une phase d'apprentissage des motifs grâce à la formule de Hebb qui conduit à une formule exprimant les poids en fonction des motifs reconnus (cf. équations 3.3). Le fonctionnement de ce type de réseau requiert des états stables, ce qui ne semble pas être le cas dans le cerveau : les pulses sont des évènements précis émis successivement par les neurones, et la stabilité du système correspondrait à la saturation ou au silence des neurones, ce qui n'est pas riche en information. Cependant, la formule de Hebb est très générale et a de nombreuses applications avec d'autres architectures. Il s'agit d'une forme d'auto-organisation dépendant des entrées d'un réseau en renforçant les connexions des neurones synchrones, on en verra une autre application dans la partie 3.3.

$$\partial \omega_{ij} = \alpha \times x_i \times x_j$$

$$\omega_{ij} = \frac{1}{N} \sum_k m_i^k \times m_j^k \text{ pour } i \neq j \text{ et } \omega_{ii} = 0$$

Équations 3.3 : Loi de Hebb et expression des poids du réseau de Hopfield.
La formule en haut est la loi de Hebb, avec x_i et x_j les activités des neurones i et j , et α un coefficient. L'apprentissage hebbien renforce les connexions des neurones synchrones en augmentant les poids proportionnellement aux activités des neurones connectés.

Habituellement, les poids sont positifs et ne peuvent qu'augmenter, et on peut aussi effectuer une renormalisation du vecteur poids après sa mise à jour pour éviter qu'il ne diverge. En bas, les formules des poids en fonction des motifs reconnus par le système avec N le nombre de neurone du réseau, et m_j^k est la composante j du vecteur-motif appris numéro k .

3.1.3 Spécificités de l'entraînement des RNRP « biologiques »

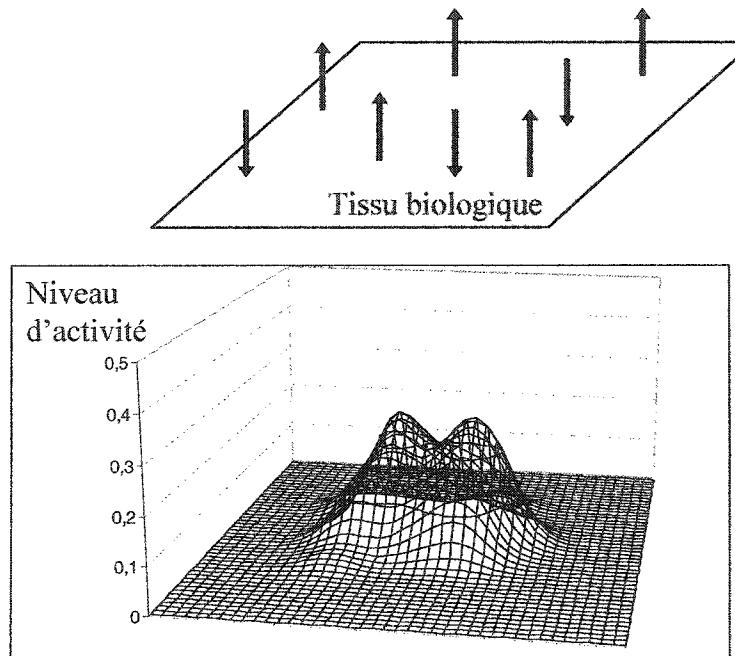
À l'instar des exemples précédents, de nombreux types de réseaux neuronaux ont été développés spécifiquement pour résoudre des problèmes de nature algorithmique tels que l'approximation de fonctions mathématiques, la classification ou la reconnaissance de motifs. Leur architecture est dédiée à ces tâches et dans un problème d'optimisation d'un de ces réseaux, l'entraînement va de paire avec la recherche de l'architecture minimale, c'est-à-dire qui génère le moins de fonctions possibles. La philosophie de l'optimisation des paramètres synaptiques d'un RNRP pour obtenir un modèle réaliste de tissu biologique est différente :

- L'**architecture** de RNRP biologiquement réalistes contient un **nombre élevé de neurones** et elle est **globalement récurrente**, on doit donc trouver un moyen de tirer parti de la profusion de fonctions générées.
- Du point de vue algorithmique, les fonctions internes des neurones utilisés (I&E ou HH) sont **non dérivables**, notamment à cause du seuil d'activation et des délais synaptiques, ce qui exclue des méthodes d'optimisation classiques comme la descente de gradient.
- Les **signaux** en œuvre sont **dynamiques**, et le codage de l'information visuelle est basé sur les potentiels d'actions qui sont des évènements précis, ce qui fait que le système biologique ne passe pas forcément par des états stables.
- L'**information visuelle** est de plus **répartie** sur l'ensemble des neurones, ce qui pose le problème de définir un objectif, nécessaire dans le cas d'un apprentissage supervisé, d'autant que l'on ne connaît pas suffisamment le codage en œuvre (par exemple comment est codée une ligne lumineuse? Au niveau des neurones d'orientation uniquement? Etc.).

Comme on ne dispose pas de méthode d'optimisation spécifique aux réseaux neuronaux récurrents, au sens où l'on pourrait entraîner un tel réseau en fonction d'un objectif en des sorties définies dans l'architecture lorsque l'on applique des entrées connues, on va devoir recourir à d'autres algorithmes d'optimisation et les adapter à notre problème.

3.2 Entrainer des RNRP grâce à des méthodes générales d'optimisation

On peut s'inspirer de la méthode d'entraînement des RNPA en se basant sur la « correspondance entrées-sorties ». Pour reproduire le comportement du tissu biologique compte tenu de la topographie, on entraîne un RNRP à reproduire les activités en un certain nombre de sites de lecture (sorties) lorsque l'on lui applique des stimuli connus au niveau d'électrodes (entrées). On interpole ainsi l'activité globale du tissu grâce à la connaissance partielle en certains points (cf. figure 3.3). À terme, on désire appliquer cette méthode pour reproduire les comportements relatifs aux objectifs 3 et 4.



Figures 3.3 : Entrainement d'un RNRP en définissant des neurones de sortie.
On définit dans le réseau des sites de lecture (sorties, les flèches ascendantes sur le dessin en haut) qui peuvent être des neurones dont on enregistre l'activité. Les entrées (flèches descendantes) correspondent aux électrodes dont on connaît des trains de pulses envoyés dans le tissu. On a ainsi une fonction des trains de pulses en entrée vers des trains de pulses en sortie à approximer avec un RNRP pour reproduire l'activité globale du réseau en respectant la topographie (représentée de manière lissée pour un tissu bidimensionnel dans le graphique du bas).

On définit comme entrées certains neurones (noeuds) du réseau qui représentent les électrodes et on va forcer leur activité avec les trains de pulses des stimuli. Les paramètres synaptiques des connexions partant de ces noeuds d'entrée modélisent l'effet de l'électrode sur les neurones voisins (on connecte à priori avec tous les voisins dans un voisinage de taille fixée). Pour les sorties, on considère que l'on a accès à l'activité d'un seul neurone (à la manière d'une électrode intracellulaire). On doit alors approximer la fonction (très complexe) de trains de pulses en entrée vers des trains de pulses en sortie qui est « calculée » par le circuit neuronal biologique. Les entrées et sorties constituent des échantillons d'entraînement, obtenus par des expériences sur des cobayes ou que l'on peut construire pour modéliser un phénomène physiologique connu (on pense aux objectifs 3 et 4 en particulier).

On doit d'abord déterminer une architecture adéquate de sorte que le réseau soit suffisamment puissant pour approximer la fonction sus-décrise. On peut prendre à priori un réseau complètement connecté et supprimer les connexions non significatives en fin d'optimisation, pour n'avoir que sa taille à ajuster. Approximer la fonction nécessite de comparer les sorties du RNP (résultats) avec des trains de pulses correspondant à l'activité que l'on veut reproduire (objectifs), pour un certain nombre d'échantillons (ensemble d'entraînement). Il faut donc définir une métrique sur les trains de pulses, idéalement cohérente avec le codage de l'information visuelle dans ces trains (cf. partie 2.1.5). Ainsi, un réseau entraîné reproduirait l'information visuelle codée dans le tissu modélisé avec une erreur mesurable par cette métrique. On ne connaît pas assez le codage pour définir une telle métrique, mais on sait que certaines composantes visuelles sont codées dans les positions individuelles des pulses à l'intérieur des trains et non par les densités de pulses (cf. partie 2.1.5). On va donc tester cette méthode avec deux métriques : une qui correspond à évaluer les densités de pulses émises et l'autre qui rend compte des positions des pulses prises individuellement.

Dans un premier temps, on va tester la capacité d'un RNP à reproduire le comportement de son architecture avec des paramètres synaptiques fixés (le réseau objectif). On construit une fonction d'erreur entre les sorties résultats et objectifs avec la

métrique définie et on utilise un algorithme d'optimisation général pour rechercher des paramètres synaptiques qui correspondent à une erreur suffisamment faible. On a choisi l'algorithme génétique qui est bien adapté pour tous les problèmes d'optimisation de fonctions d'un grand nombre de paramètres de manière générale [7]. Même si on peut raffiner la version proposée dans la partie 4.2, il s'agit avant tout d'étudier si l'on peut contraindre le comportement global du réseau à « converger » vers celui du réseau objectif, i.e. en regardant tous ses neurones et pas uniquement ceux de sortie. En effet, lorsque l'on entraîne un RNPA avec des entrées et des sorties, les neurones de la couche cachée sont adaptés mais leur comportement n'est pas « maîtrisé » (ce qui pourrait donner des comportements irréalistes dans notre cas comme la saturation de neurones). On désire au contraire obtenir un comportement réaliste de la part du réseau entier. L'application de cette méthode au cas de RNP de petite taille avec des synapses non dynamiques est étudiée dans le chapitre 4.

En tirant parti de l'organisation du cortex en colonnes, on peut imaginer une autre manière de reconstruire le tissu neuronal. On entraîne des petits circuits neuronaux à modéliser des colonnes corticales, à l'instar de boîtes noires dont on fixe un certain nombre d'entrées et de sorties, et on les juxtapose pour obtenir un pavage modélisant le tissu cortical (cf. figure 3.4). Il reste ensuite à ajuster les connexions inter-colonnes entre les entrées et sorties de colonnes voisines pour optimiser le comportement du réseau total. Ceci nous affranchit du problème lié au comportement des nœuds internes mais on doit alors définir une fonction (sur les trains de pulses) des neurones d'entrée vers ceux de sortie pour le patron de colonne corticale qui soit cohérente avec le comportement d'une colonne réelle. De plus, l'architecture des RNP doit alors être optimisée car le surentraînement peut se produire (le patron de colonne n'est pas « topographiquement » biologique). En résumé, cette méthode n'apparaît que théoriquement possible, du fait de la fonction sur les trains de pulses très complexe à déterminer.

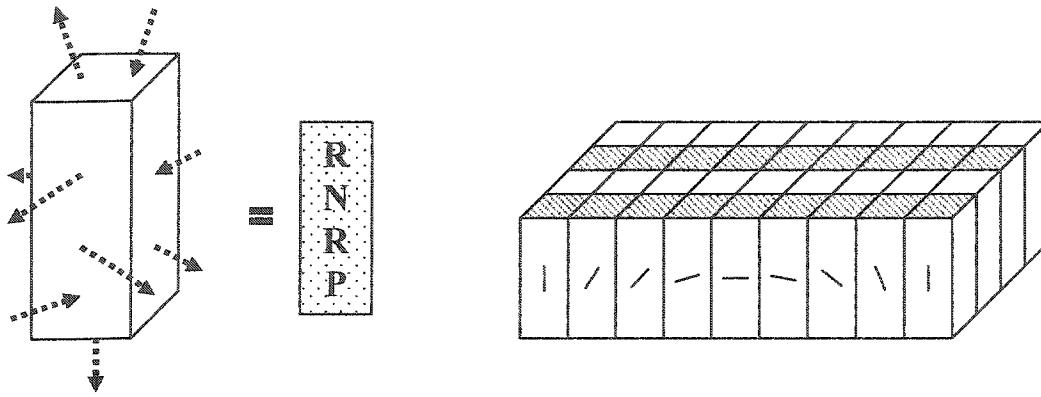


Figure 3.4 : Modèle du cortex visuel par juxtaposition de colonnes corticales.
 On modélise chaque colonne corticale par un RNRP que l'on entraîne selon une méthode générale comme une unité-type, avec des entrées et sorties définies (flèches en pointillés et en gras). On juxtapose ensuite ces RNRP et on les connecte entre eux via certains neurones (ceux définis comme entrées ou sorties dans l'entraînement précédent par exemple : on peut voir ces RNRP comme des boîtes noires). Enfin, il s'agit d'ajuster les paramètres de ces liaisons grâce à une nouvelle optimisation pour obtenir un comportement global tel que décrit par Hubel et Wiesel : chaque colonne a une dominance oculaire (hachuré et non hachuré), une orientation préférentielle (cf. les barres en façade de gauche à droite) et correspond à une portion du champ visuel (non représenté ici, d'avant en arrière).

3.3 Apprentissage non supervisé : les cartes topographiques auto-organisatrices

Plutôt que d'essayer de construire la fonction sur les trains de pulses énoncée à la fin de la partie précédente, ce qui s'annonce fastidieux, on peut plutôt utiliser des règles locales du style de la loi de Hebb pour auto organiser un RNRP (3^{ème} objectif). En s'inspirant des travaux de Hubel et Wiesel sur le fonctionnement du cortex visuel, Kohonen [16] a développé un modèle mathématique qui auto-adapte ses paramètres « synaptiques » en fonction des entrées que l'on présente dans un réseau neuronal particulier selon une loi dérivée de la formule de Hebb (cf. figure 3.5). On traduira ici le terme « self-organizing map » par carte topographique auto-organisatrice (CTAO).

On obtient à la fin de l'optimisation une correspondance entre les entrées et la topographie de l'activité du réseau : des entrées « proches » stimulent des zones voisines. On peut considérer que le réseau effectue une extraction des composantes principales de manière non supervisée : les entrées appartiennent souvent à un espace de grande dimension et sont projetées dans un espace de petite dimension (une CTAO est bidimensionnelle ou tridimensionnelle en général).

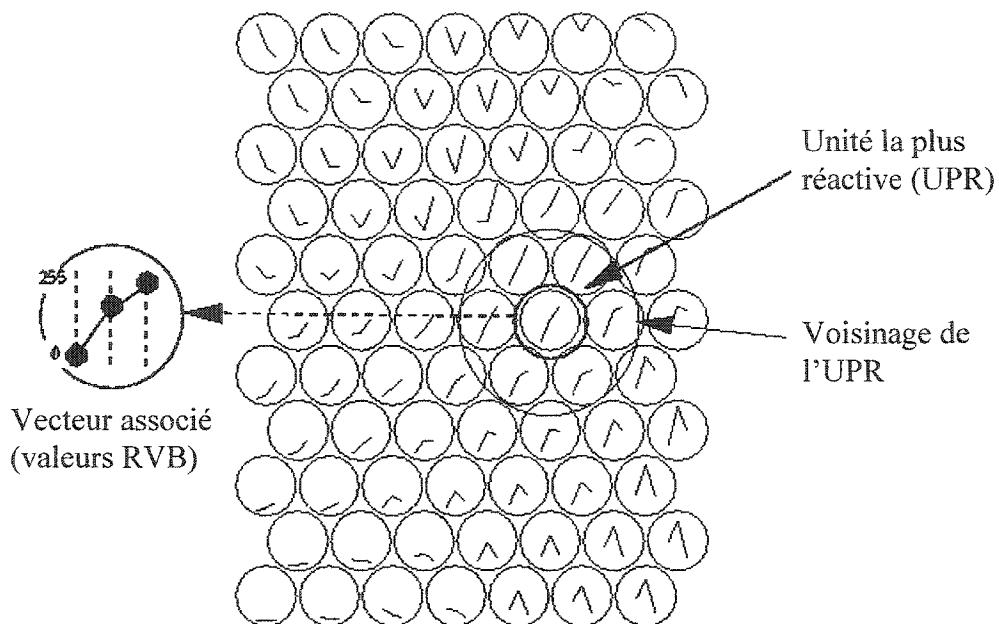


Figure 3.5 : Exemple de CTAO.

(*Thèse de Timo Honkela de l'université de technologies de Helsinki, Finlande*)
Les stimuli sont représentés par des vecteurs à trois dimensions, chaque unité représente un modèle de stimulus et on applique le vecteur-entrée à tous les neurones en effectuant un produit scalaire du vecteur représenté et de l'entrée pour trouver l'unité la plus réactive (UPR). Ici, la loi d'apprentissage modifie les voisins de l'UPR pour rapprocher leurs modèles correspondants de celui de l'UPR. À la fin de l'apprentissage, des unités voisines représentent des vecteurs de modèles proches.

Une des applications originelles des CTAO est la classification (cf. figure 3.4) :

- on construit un réseau où chaque neurone est une représentation d'un vecteur dans le domaine de variation des entrées,

- ce réseau s'auto-adapte en fonction d'un ensemble d'entraînement selon une règle locale : le meilleur neurone-modèle pour une entrée (le plus ressemblant) modifie les poids de ses voisins pour que leurs modèles se rapprochent du sien,
- on réduit progressivement le voisinage à l'intérieur duquel on effectue les mises à jour, afin que le système se stabilise,
- pour une nouvelle entrée inconnue, on regarde quelle unité du réseau répond le mieux à cette entrée et on extrapole afin de rapprocher l'entrée inconnue d'une donnée connue, en l'occurrence celle qui correspond au neurone le plus réactif.

Cet exemple illustre le concept d'entraînement non supervisé dérivé de la loi de Hebb pour les CTAO. On va s'inspirer de cet algorithme et le transposer dans une autre architecture (un RNRP de style biologique) afin de reproduire l'auto-structuration du tissu cortical correspondant au 3^{ème} objectif fixé pour notre modèle. Cela suppose de modifier les règles de mise à jour des poids, et on se contente de laisser des délais initialisés aléatoirement inchangés. Notamment, on ne va pas rechercher le neurone le plus actif (en terme de densité de pulses émises par exemple) et modifier ses liaisons efférentes, mais on va plutôt déterminer un seuil d'activité au-dessus duquel tout neurone est « mis à jour ». La recherche de l'UPR a en effet peu de chance d'être réaliste biologiquement car elle demande trop de calculs.

Une dernière divergence concerne la stabilisation qui est obtenue dans le modèle de Kohonen en réduisant le voisinage à l'intérieur duquel les modifications sont effectuées. Des études ont montré que ce phénomène pouvait être reproduit en utilisant une pondération des mises à jour au moyen de poids représentant une fonction-chapeau spéciale (cf. figure 3.6) autour de l'UPR dans un certain voisinage de taille fixe [17]. De plus, les paramètres des synapses ne sont pas fixes dans le temps (au contraire de l'architecture qui devient définitive quelques semaines après la naissance, cf. partie 2.1.5), ce qui signifie que notre modèle devrait stabiliser si on lui présente toujours les mêmes types de stimuli, mais sans recourir à une réduction de la zone de mises à jour. Ainsi, l'algorithme devient plus « biologique ».

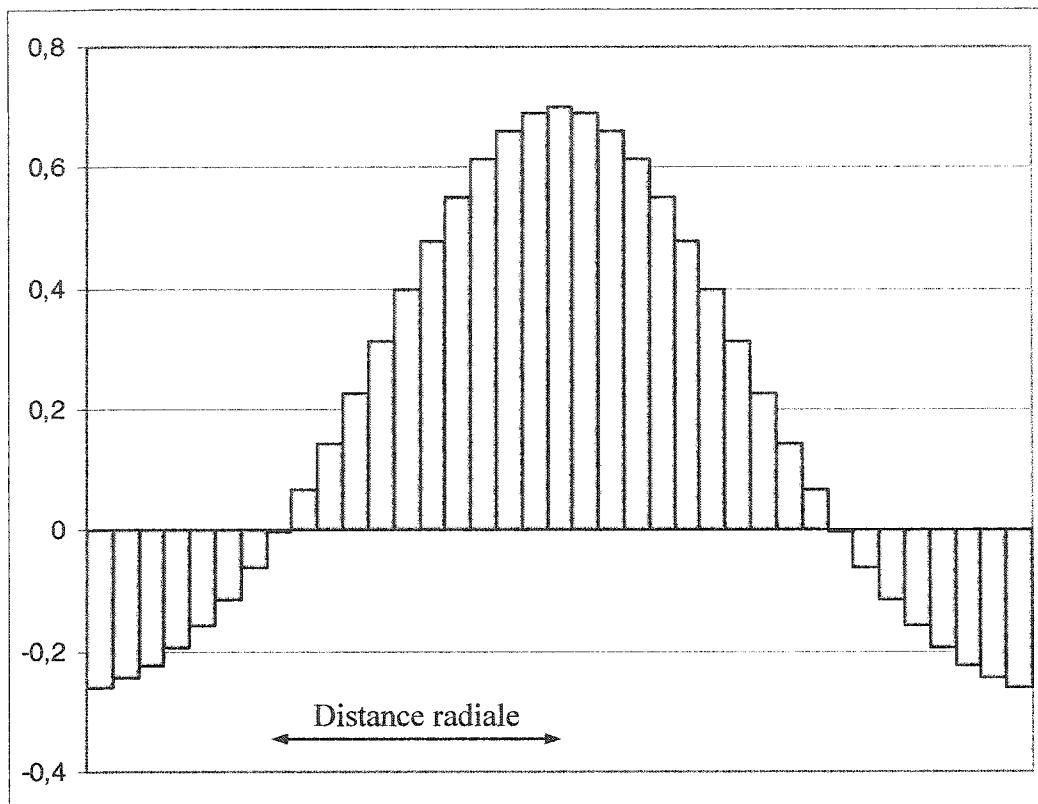


Figure 3.6 : Fonction-chapeau de pondération des mises à jour dans la CTAO. Cette fonction est radiale et doit comporter des poids positifs au centre et décroître jusqu'à des poids négatifs à la limite du voisinage fixe considéré.

On étudiera dans le chapitre 5 comment entraîner un RNNP de manière qu'il ait un comportement auto-adaptatif comme une CTAO et modélise les connexions latérales entre les inter-neurones de la couche 4 du cortex visuel. De plus, il faudra coupler cette auto-organisation avec un apprentissage supervisé afin d'entraîner certains neurones à effectuer des tâches spécifiques, et déterminer dans quelle mesure ces deux méthodes sont compatibles.

4 Entraînement de RNP par algorithmes génétiques

On considère dans ce chapitre l'ajustement des paramètres synaptiques d'un RNP avec un algorithme génétique. On présente d'abord l'architecture du réseau et une métrique définie sur les trains de pulses, l'algorithme d'apprentissage puis le protocole expérimental, et enfin on discute les résultats. On cherche à reproduire chez un RNP le comportement d'un réseau de même architecture avec des paramètres synaptiques fixés. On étudie la convergence du comportement en certains points du RNP entraîné vers celui du réseau-objectif, puis celle de son comportement global. Cette étude constitue un préliminaire à l'entraînement de tels RNP pour modéliser des microcircuits neuronaux biologiques, où d'autres problèmes se posent comme déterminer une architecture adéquate, ce qui sera discuté dans la conclusion. Il est indépendant du chapitre 5.

4.1 Description du type de réseau à entraîner

4.1.1 Architecture du réseau

Conformément à ce qui a été dit dans la partie 3.2, on définit dans le RNP des neurones (ou noeuds) de sortie, les autres sont les noeuds internes et tous ces neurones sont interconnectés (cf. figure 4.1). On ajoute des noeuds d'entrée connectés à tous les neurones du RNP (sauf les autres noeuds d'entrée), ce qui équivaut à ignorer des effets indirects de la stimulation d'une électrode dans son voisinage. L'effet de l'électrode sur les neurones du tissu est paramétré par les poids et délais synaptiques des connexions qui partent du nœud d'entrée correspondant.

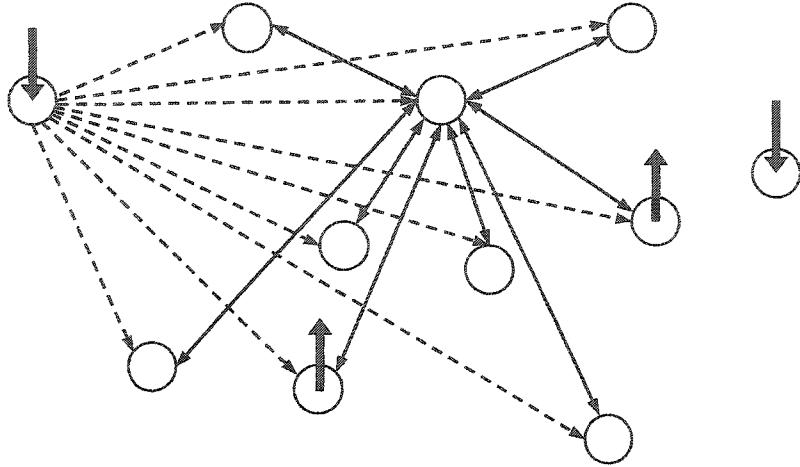


Figure 4.1 : Architecture d'un RNN avec des entrées et sorties.

Les neurones d'entrées sont représentés avec une flèche descendante en gras qui représente la stimulation, de même que ceux de sortie avec des flèches ascendantes. Seules les connexions d'un noeud d'entrée et d'un noeud interne sont représentées : les doubles flèches en trait fin symbolisent deux connexions distinctes (une dans chaque sens) entre deux neurones; les flèches simples en trait fin et en pointillés représentent une connexion d'un nœud d'entrée vers un autre nœud.

On enregistre l'activité des nœuds de sortie et on applique au niveau de chaque noeud d'entrée un train de pulses (ou stimulus). Une entrée du réseau est l'ensemble des stimuli pour chaque noeud d'entrée, et une sortie du réseau est l'ensemble des trains de pulses enregistrés à chaque neurone de sortie. Durant l'entraînement, on appelle résultat la sortie du réseau entraîné et objectif la sortie du réseau objectif.

4.1.2 Calcul de la fonction d'erreur utilisée pour l'optimisation

Le but de l'entraînement du réseau est d'obtenir pour un ensemble de stimuli des résultats « équivalents » aux objectifs, ce qui implique de comparer ces trains de pulses. Comme dit dans la partie 3.2, le codage de l'information visuelle au niveau des neurones du cortex n'est que partiellement connu et on ne sait pas si se restreindre aux densités d'activité permet d'extraire des informations suffisamment pertinentes afin d'évaluer des

protocoles de stimulation. On va donc étudier deux versions d'une même métrique basée sur un lissage des trains de pulses avec une fenêtre temporelle « gaussienne » (cf. figure 4.2) :

- avec une taille de la fenêtre est du même ordre que la durée minimale inter-pulses (i.e. la période de relaxation des neurones I&E), soit une demi-largeur de fenêtre de 5ms,
- avec une demi-largeur de fenêtre de 50ms, ce qui correspond à un lissage temporel très important (i.e. calcul de densité d'activité des neurones de sortie).

En utilisant cette métrique, on calcule l'erreur entre deux trains de pulses (cf. équation 4.1), puis l'erreur pour un stimulus entre deux réseaux (ayant des paramètres synaptiques distincts) comme la somme des erreurs au niveau de chaque noeud de sortie (on aurait également pu choisir le maximum, ce qui sera discuté dans la partie 4.4). Grâce à la présence du dénominateur, les contributions de chaque train de pulses sont équilibrées dans ce calcul. Par extension, on désigne par erreur d'un réseau entraîné l'erreur entre le réseau-objectif et lui.

$$\text{erreur} = \frac{\int (S_{res}(t) - S_{obj}(t))^2}{\int (S_{obj}(t))^2}$$

Équation 4.1 : Formule du calcul de l'erreur pour un noeud.

S_{res} et S_{obj} sont respectivement les séries lissées résultat et objectif. Le symbole d'intégration correspond une intégration de style Riemann sur le temps, que l'on approxime en sommant les valeurs successives des séries temporelles, à un coefficient près. Le dénominateur a été choisi pour que l'erreur pour tout train-objectif avec un train vide de pulse soit 1. Cela permet d'équilibrer les contributions de chaque train dans le calcul de l'erreur sur un réseau, et aussi les erreurs du réseau entraîné pour chaque stimulus de l'ensemble d'entraînement. De ce fait, les deux séries ne jouent pas des rôles symétriques.

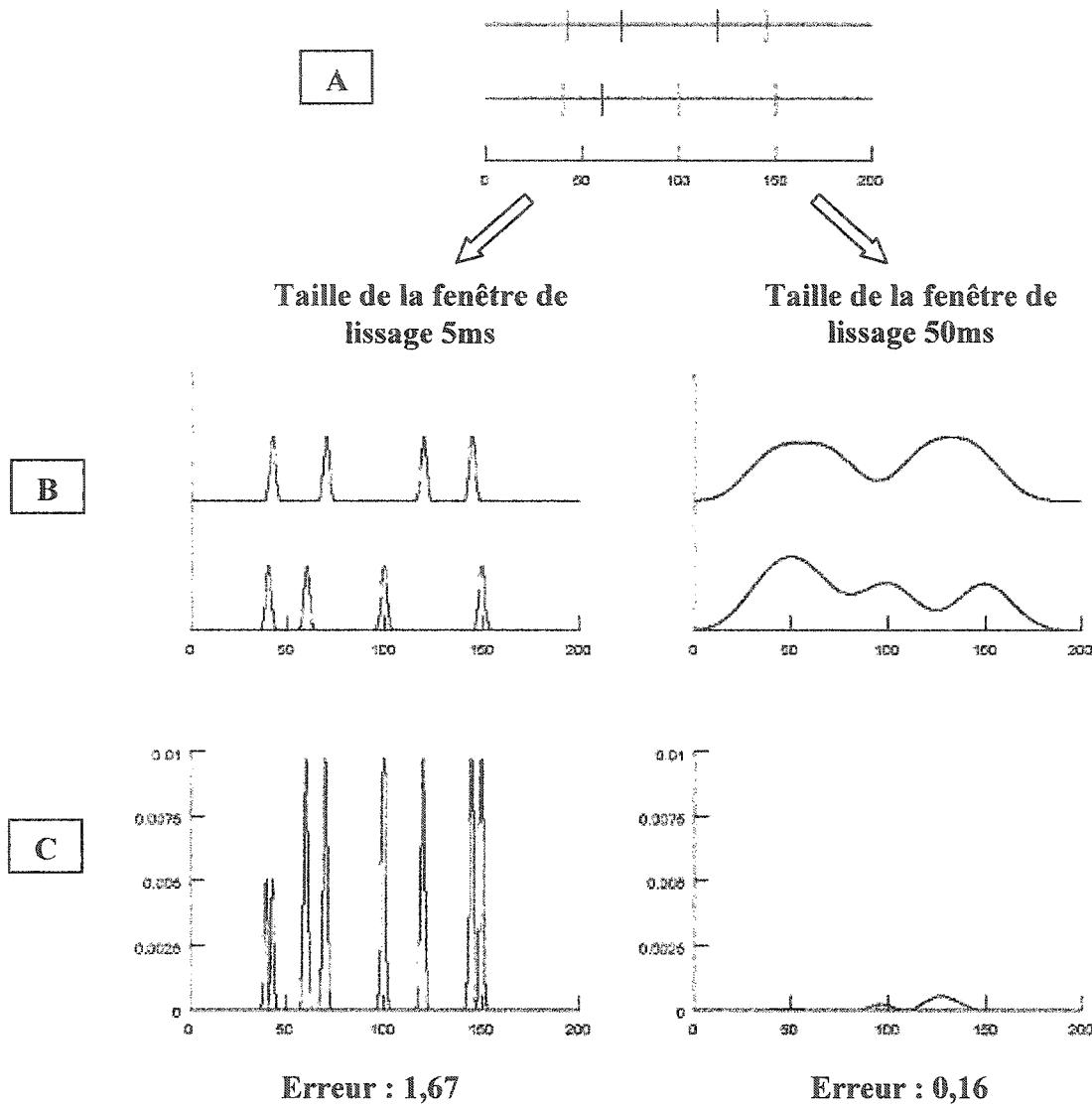


Figure 4.2 : Calcul de l'erreur entre deux trains de pulses.

À partir de 2 trains de pulses (A, avec l'objectif en haut et le résultat en bas), on construit des séries temporelles lissées avec comme noyau des gaussiennes normalisées et tronquées (la surface sous chaque courbe excède 0,99) de différentes largeurs caractéristiques : à gauche avec une fenêtre de lissage de taille 5ms, et 50ms à droite (B, les échelles sont différentes). Puis on calcule la série-différence élevée au carré de ces deux séries lissées (C, les échelles des deux graphes de droite et gauche sont ici les mêmes pour permettre la comparaison). L'erreur se calcule par « intégration temporelle » de la série-différence au carré pondérée par l'inverse de l'intégrale de la série-objectif au carré (cette intégration est proportionnelle à la sommation des termes de la série temporelle étant donné que l'on travaille en temps discret).

On obtient ainsi un lissage fin qui respecte les positionnements individuels des pulses dans les trains de pulses, tandis que l'autre lissage plus grossier correspond plutôt à évaluer les densités d'activité des neurones. Pour chaque cas, on doit définir une valeur de l'erreur en dessous de laquelle on considère des trains de pulses comme proches ou équivalents. Dans le cas fin, on choisit 0,2 comme erreur limite, ce qui correspond par exemple à deux trains de pulses identiques décalés de 1ms dans le temps environ. Dans le cas grossier, on prend aussi 0,1 qui correspond à un décalage de 20ms ou encore à des trains de même fréquence mais de nombres de pulses différents (10% de pulses en plus ou en moins). La figure 4.3 donne une illustration des variations de l'erreur entre deux trains de pulses, l'un obtenu à partir de l'autre par translation temporelle.

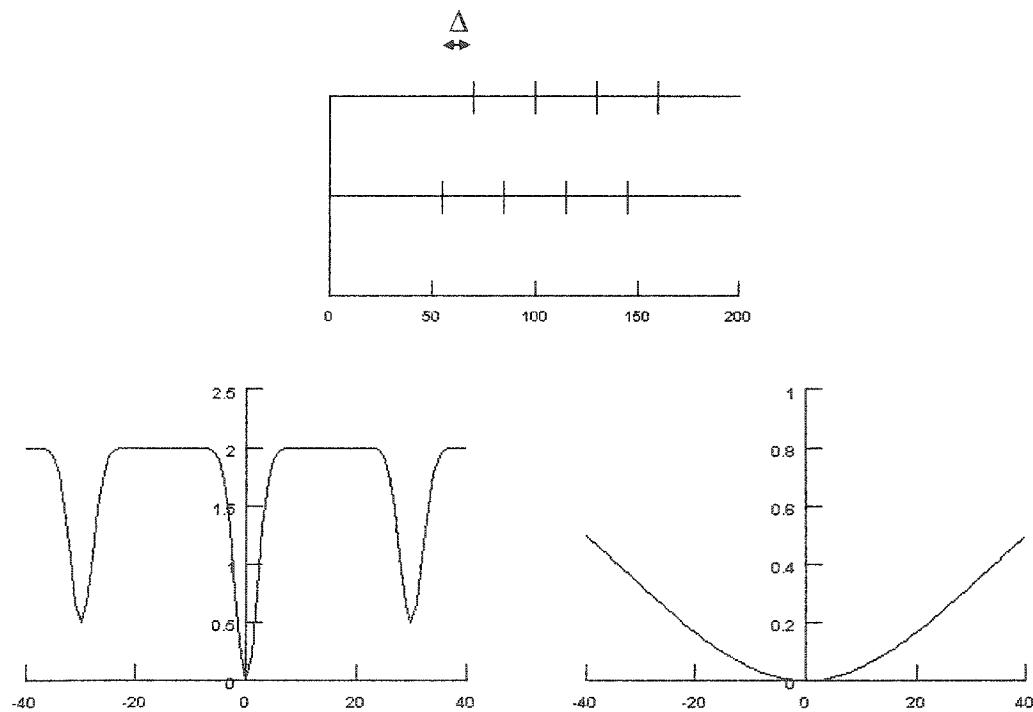


Figure 4.3 : Variations de l'erreur lors d'une translation temporelle.

En haut, les trains de pulses de 200ms de durée sont identiques à une translation temporelle près de Δ ms. Les courbes des graphes en bas correspondent aux tailles de fenêtre de la figure 4.2 (5ms à gauche et 50ms à droite, les échelles sont différentes) et représentent l'erreur entre les trains de pulses en haut en fonction de Δ (en ms). On voit que la courbe de gauche a plusieurs minima locaux tandis que la courbe de droite n'en a qu'un. De plus, la courbe de gauche a une pente élevée et atteint un palier à ± 5 ms autour du minimum global.

Ce lissage se justifie car on désire que l'erreur respecte une certaine continuité : elle doit diminuer graduellement au fur et à mesure que les trains de pulses deviennent « semblables » pour faciliter l'optimisation qui va utiliser cette erreur pour calculer la fonction-objectif. C'est pourquoi l'optimisation doit être à priori plus facile avec le lissage grossier. Enfin, le lissage fin induit une fonction d'erreur suffisamment contraignante : il rend compte de décalages temporels de l'ordre de la milliseconde, et une translation temporelle de 2ms donne la même erreur qu'une différence du nombre de pulses de 20% entre deux trains de même fréquence et synchronisés. On peut donc supposer transposer cette méthode à pratiquement n'importe quelle autre métrique, si l'on parvient à optimiser un réseau avec celle correspondant au lissage fin.

4.2 Recherche des paramètres synaptiques optimaux par algorithme génétique

L'algorithme génétique s'inspire de l'évolution et de la sélection naturelles : il s'agit de faire évoluer une population d'individus pour l'adapter à des « contraintes » définies par un environnement, de manière que ses individus deviennent en moyenne le mieux adaptés possible en même temps que son meilleur individu devienne de mieux en mieux adapté. Dans le cas de la recherche du minimum d'une fonction, un individu est un jeu de variables (ou paramètres) et plus la valeur de la fonction pour ses variables est basse, mieux il est adapté. Cette méthode s'apparente à une exploration du domaine de variation des paramètres par une population de points dans ce domaine de variation qui « collaborent » pour rechercher un minimum. Cette méthode est particulièrement désignée pour les fonctions de nombreux paramètres ou non régulières.

On va donc rechercher un minimum de l'erreur comme fonction des paramètres synaptiques d'un réseau d'architecture fixe. On définit plusieurs niveaux d'organisation des paramètres (cf. figure 4.4) :

- un **gène** représente un paramètre et une valeur d'un paramètre est l'équivalent d'un allèle pour un gène en biologie,
- un **chromosome** regroupe plusieurs gènes qui se retrouvent liés (comme dans la réalité biologique, cette propriété influe sur leurs probabilités d'être transmis de manière conjointe),
- un **individu** correspond à un génome (ensemble de tous les gènes sans redondance, où les gènes étant regroupés en chromosomes).

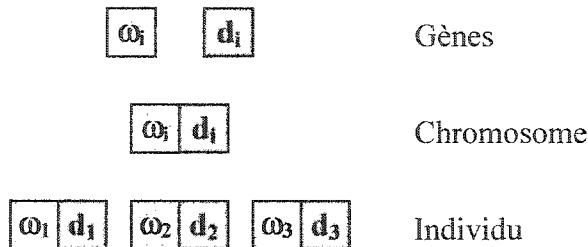


Figure 4.4 : Niveaux d'organisation des paramètres pour l'algorithme génétique. L'individu en exemple possède 3 chromosomes de 2 gènes chacun, où chaque gène est un paramètre synaptique : soit un poids (ω), soit un délai (d). Les gènes sont eux-mêmes codés au moyen de bits dans la mémoire de l'ordinateur.

Ici, un chromosome représente une synapse et possède deux gènes qui sont le poids et le délai de cette synapse; un individu représente un réseau et son comportement est régi par ses gènes (paramètres synaptiques). On peut aussi rajouter un niveau dans l'organisation des paramètres : le codage des gènes en bits, mais cela influe uniquement sur les opérateurs de mutation et de reproduction décrits plus loin.

La **fonction-objectif** dont l'algorithme génétique va chercher un minimum est construite à partir de l'erreur définie dans la partie 4.1.2 : on définit une « contrainte » de l'environnement extérieur comme l'erreur pour un seul stimulus (i.e. l'erreur du réseau entraîné définie dans la partie 4.1.2) et l'**évaluation** d'un individu correspond à l'erreur moyenne calculée sur l'ensemble d'entraînement (c'est la valeur de la fonction-objectif pour cet individu).

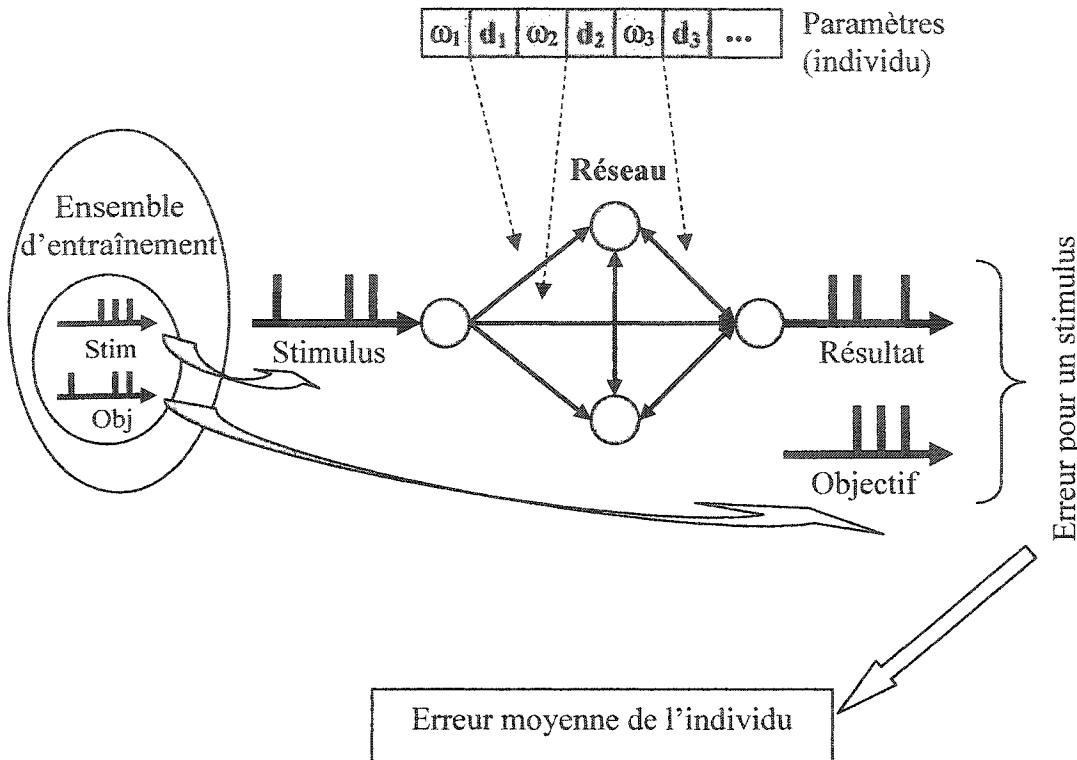


Figure 4.5 : Évaluation d'un individu par rapport à un ensemble d'entraînement.
Un individu correspond aux paramètres synaptiques d'un réseau, et son comportement consiste en les séries temporelles en sortie du réseau lorsque l'on le simule avec en entrées des trains de pulses pris dans un ensemble d'entraînement. On prend la moyenne des erreurs relatives à chaque stimulus pour calculer la valeur fonction-objectif correspondant à l'individu.

Une population de départ constituera la première génération et on va combiner plusieurs processus inspirés de l'évolution naturelle pour passer d'une génération à la suivante. À noter que la taille de la population est constante au fil des générations. Trois étapes distinctes se suivent dans l'évolution d'une génération (cf. figure 4.6.C) :

- La **sélection par tournoi** élimine les individus les « moins bien » adaptés : on évalue tous les individus de la population et on effectue des combats deux à deux entre individus tirés au hasard, durant lesquels le meilleur gagne. On garde un pourcentage de la population défini par un méta-paramètre.

- Des **mutations** affectent les individus survivants avec une certaine probabilité et pour les individus touchés, chacun de leurs gènes a une probabilité de muter, c'est-à-dire de voir sa valeur changée pour un tirage aléatoire dans le domaine de variation (cf. figure 4.6.A).
- La création de nouveaux individus se fait de deux façons distinctes :
 - la **reproduction** qui est inspirée de la méiose biologique et qui fait intervenir deux individus survivants et qui croise leurs gènes pour former le génome du nouvel individu (cf. figure 4.6.B),
 - la **création aléatoire** consiste à tirer aléatoirement les gènes du nouvel individu dans les domaines de variation des paramètres correspondants.

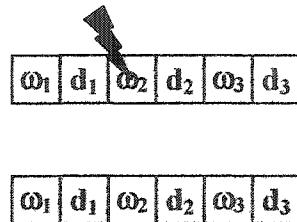


Figure 4.6.A : Détail d'une mutation.

Chaque individu a une probabilité d'être affecté par une mutation et dans le cas positif, chacun de ses gènes a une probabilité de muter, c'est-à-dire que sa valeur est remplacée par un nombre tiré aléatoirement dans le domaine de variation correspondant.

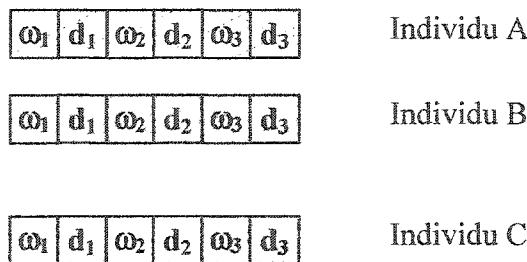
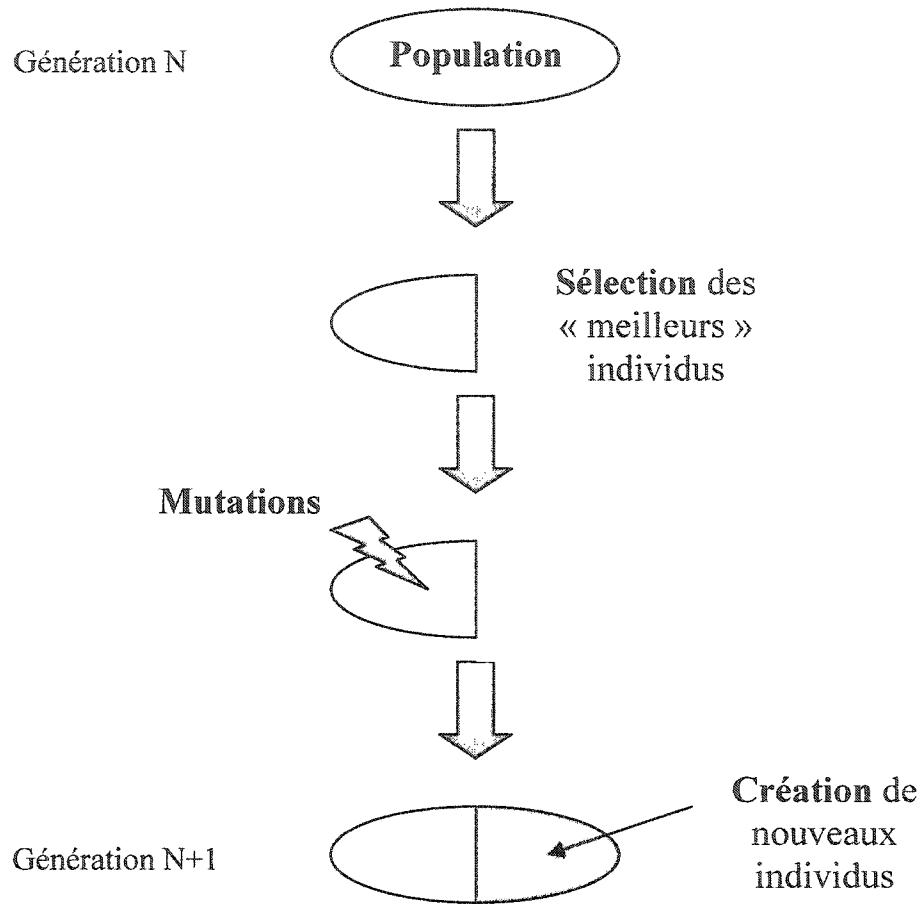


Figure 4.6.B : Détail de la création par reproduction.

Le nouvel individu C est créé à partir des individus A et B en tirant les chromosomes aléatoirement chez l'un des « parents ».



*Figure 4.6.C : Évolution de la population durant une génération.
On retrouve les 3 étapes qui permettent de passer d'une génération à la suivante.*

Bien que sa convergence n'ait été vraiment prouvée formellement, l'algorithme génétique a montré son efficacité en pratique [7]. Il se révèle efficace en général dans les cas où l'on ne connaît pas d'algorithme performant spécifique au problème (comme le nôtre qui n'est pas un problème d'entraînement classique de réseaux de neurones), et en particulier pour rechercher un extremum d'une fonction de nombreux paramètres ou bien irrégulière. L'efficacité des opérateurs de reproduction et de mutation dépend du codage des paramètres; comme on omet le niveau des bits, on n'utilise pas les classiques « cross-over ».

4.3 Protocole de simulation

On va tester deux architectures :

- un réseau R4 à 4 neurones avec 1 nœud d'entrée et 1 de sortie, soit 9 connexions et un total de 18 paramètres à optimiser (cf. figure 4.7),
- un réseau R10 à 10 neurones avec 2 nœuds d'entrée et 1 nœud de sortie, soit 72 connexions pour un total de 144 paramètres.

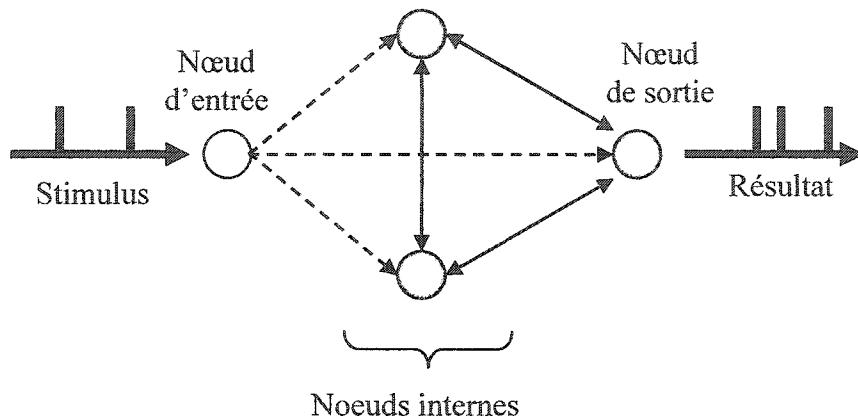


Figure 4.7 : Architecture du réseau R4.

R4 comprend un neurone d'entrée, deux neurones internes et une sortie. Il possède 9 connexions en tout (cf. figure 4.1 pour les significations des flèches), soit 18 paramètres.

Les paramètres internes des neurones sont fixés à 3ms pour la période réfractaire, 10ms comme durée caractéristique de fuite (cf. partie 2.2.2). Le domaine de variation est de [-1;1] pour les poids et de [0;20]ms pour les délais et on utilise des synapses statiques dans un premier temps. On tire les paramètres du réseau-objectif aléatoirement dans ces intervalles (il existe une solution exacte au problème). La durée de simulation est de 200ms et le pas temporel de 0,1ms. Dans le cas du lissage fin, un réseau bien entraîné correspond à une erreur de 0,22, et dans le cas du lissage grossier, il s'agit de 0,12.

La taille de la population est fixée à 200 et l'algorithme génétique a été implanté avec des méta-paramètres fixes relativement standards [7] : 50% pour le taux de survie (après sélection), la probabilité d'être affecté par une ou plusieurs mutations est de 5%

pour un individu et de 5% pour qu'un gène d'un individu affecté subisse une mutation, on crée 90% des nouveaux individus par « reproduction » et les 10% restant sont générés aléatoirement.

Les ensembles d'entraînement sont composés de paires stimuli-objectifs tirés où les stimuli sont des trains de pulses disposées aléatoirement entre 0 et 200ms, mais où deux pulses ne peuvent être séparées de moins de 3ms (comme la période réfractaire des neurones). À chaque génération, on évalue les nouveaux individus avec la totalité de l'ensemble d'entraînement. En fin d'optimisation, on calcule l'**erreur d'entraînement** (en prenant le meilleur individu de la dernière génération) sur l'ensemble d'entraînement et on évalue la capacité de généralisation de cet individu en calculant l'**erreur de validation** avec un même nombre de nouvelles paires stimulus-objectifs.

On se contente ici de réseaux de taille modeste pour tester notre méthode car le nombre de paramètres à ajuster est proportionnel au carré du nombre de cellules du réseau et le domaine de variations de ces paramètres devient rapidement très vaste et de grande dimension (sous-ensemble de \mathbb{R}^n , avec n le nombre de paramètres), ce qui rend l'optimisation très complexe et longue. D'autre part, leur étude permet déjà d'observer des phénomènes intéressants concernant la convergence de l'activité globale du réseau et d'évaluer la pertinence de la présente méthode.

4.4 Résultats

4.4.1 Entraînement des noeuds de sortie

D'une manière générale, l'optimisation ne réussit pas toujours et peut stagner au niveau de minima locaux. Cela s'explique par la nature de la fonction-objectif qui est étudiée dans la partie 4.4.3. L'algorithme ne nécessite pas forcément un nombre de générations trop élevé pour aboutir à une erreur d'entraînement conforme à la limite impaire (0,1) pour le lissage grossier. Ainsi, dans le cas de R4 avec 40 stimuli et une

population de 200 individus, 100 générations peuvent suffire pour réussir l'optimisation avec ce lissage. On arrête en général l'entraînement lorsque la moyenne de l'erreur du réseau entraîné se situe en dessous des limites correspondant à un réseau bien entraîné (0,2 pour le lissage fin et 0,1 pour le lissage grossier).

L'erreur d'entraînement se répartit assez bien dans le cas du lissage grossier : son étalement est homogène et relativement faible, ce qui augure une bonne généralisation (cf. figure 4.8). La métrique influe sur les résultats l'efficacité de l'entraînement : il est plus long d'obtenir une erreur d'entraînement dans les limites fixées avec le lissage fin. De plus, la répartition de l'erreur d'entraînement est différente par rapport au cas du lissage grossier et on peut distinguer deux groupes (cf. figure 4.9.A) : des stimuli pour lesquels l'erreur est quasiment nulle et d'autres pour lesquels l'erreur se situe vers 0,5 (cela correspond à une translation temporelle de 2ms environ). Cette répartition est obtenue de manière générale après un entraînement suffisant et réduire le deuxième groupe est difficile (stagnation de la diminution de l'erreur parfois).

Pour contrer ce phénomène, on pourrait substituer le maximum des erreurs des stimuli à leur moyenne pour définir l'erreur d'entraînement. Cependant, cette option ne permet pas en général de réussir l'entraînement pour le lissage fin. Dans le cas du lissage grossier, cette option n'a qu'un faible impact sur l'entraînement. Il faudrait améliorer l'algorithme pour espérer une bonne optimisation. Cela traduit la difficulté d'entraîner les RNRN : les plages de paramètres conduisant à une erreur convenable sont localisées et obtenir en sortie des trains en gardant la disposition des pulses prises individuellement nécessite d'approcher très finement les paramètres du réseau-objectif. L'analyse de la fonction d'erreur dans la partie 4.4.3 illustre cette difficulté et également à quel point le lissage fin induit une mesure contraignante.

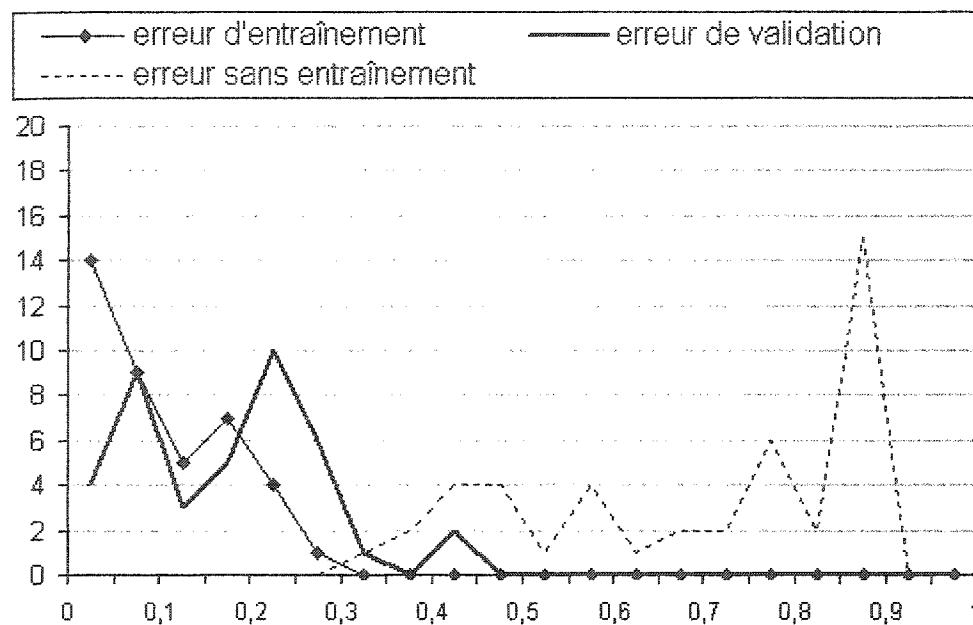
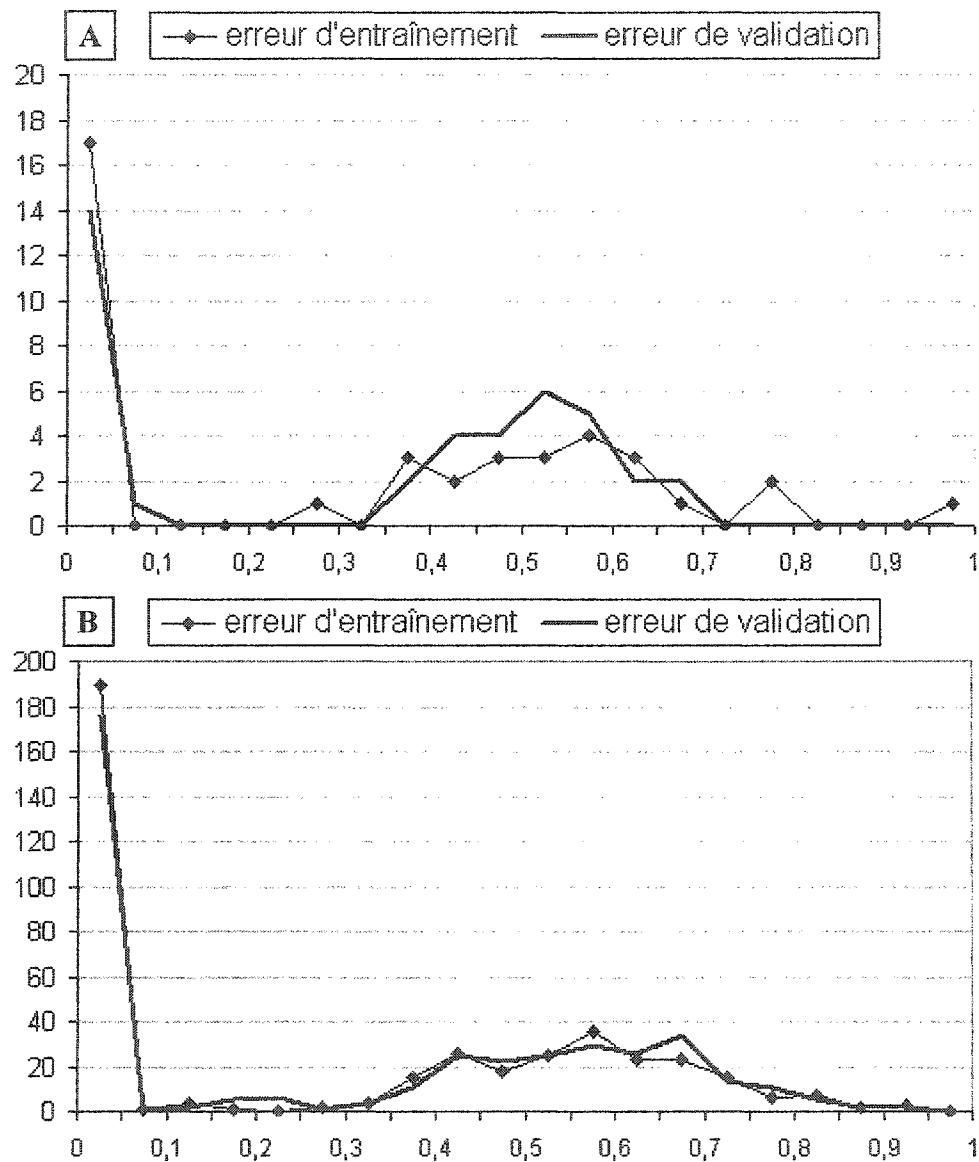


Figure 4.8 : Répartition des erreurs d'entraînement et de validation pour R4 dans le cas du lissage grossier.

Le graphique montre pour chaque intervalle le nombre de stimuli dont l'erreur se situe dans l'intervalle. La majorité des erreurs se situent en dessous de 0,2 qui correspond à un bon entraînement (en trait fin avec des marqueurs losanges) de R4. On voit que l'erreur de validation (en gras) est assez bonne pour ce cas, même si le nombre de stimuli d'entraînement est relativement faible. En effet, on n'a pas de stimulus dont l'erreur est au-dessus de 0,3 tandis que pour un réseau aléatoire non entraîné (en pointillés), et majoritairement autour de 0,85.

Comme pour les RNPA, la taille de l'ensemble d'entraînement influe sur la capacité de « généralisation » du réseau entraîné : la répartition de l'erreur de validation est plus conforme à celle de l'erreur d'entraînement pour 400 stimuli que pour 40 (cf. figure 4.8 pour le réseau R4). En quelque sorte, la « variance » entre les sorties du réseau entraîné est plus faible lorsque la taille de l'ensemble d'entraînement augmente, ce qui augure à priori une meilleure généralisation. On peut ainsi considérer l'augmentation de la taille de l'ensemble d'entraînement comme un ajout de contraintes dans l'optimisation des noeuds de sortie. Par contre, cela ne change fondamentalement ni la répartition de l'erreur d'entraînement ni celle de validation. À noter que le nombre de 400 stimuli a été

choisi car il correspond au nombre d'échantillons nécessaires pour « bien » entraîner un RNS possédant autant de paramètres à ajuster que R4 [1].



Figures 4.9 : Influence de la taille de l'ensemble d'entraînement pour R4 dans le cas du lissage fin.

On a entraîné R4 avec le lissage fin et soit 40 stimuli (A), soit 400 stimuli (B). On distingue deux groupes de stimuli dans la répartition des deux erreurs, ceux qui ont une erreur quasi nulle et ceux dont l'erreur est voisine de 0,5 (ce qui correspond à un entraînement moyen, presque réussi). On remarque que la répartition de l'erreur de validation « colle » mieux à celle de l'erreur d'entraînement quand l'ensemble d'entraînement est grand.

Dans le cas de R10, la généralisation est plutôt mauvaise avec 40 ou 400 stimuli, même si on peut noter une amélioration de la répartition pour de plus nombreux stimuli. Le nombre de stimuli nécessaire pour avoir une bonne généralisation chez un RNPA avec 72 paramètres est de 2160. Les temps de calcul étant relativement longs, on s'est contenté de 400, ce qui provient du fait que le temps de simulation des réseaux croît rapidement avec le nombre de nœuds (il est proportionnel au nombre de pulses émises). Cela dit, le fait de devoir utiliser un grand nombre d'échantillons d'entraînement pose des problèmes pour l'application de cette méthode dans le cas de grands réseaux, ce qui sera discuté dans la partie 4.5.

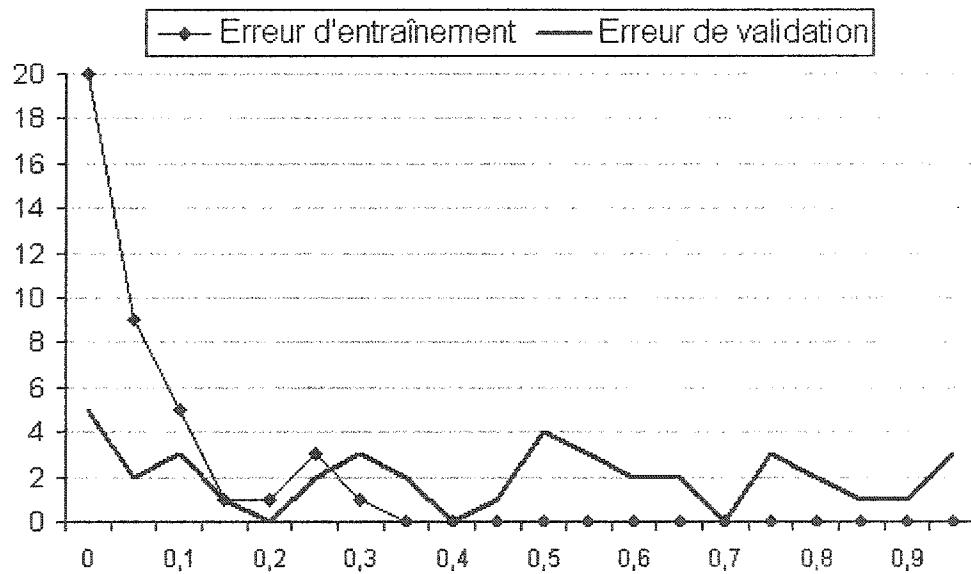


Figure 4.10 : Répartition des erreurs d'entraînement et de validation pour R10 entraîné avec 40 stimuli dans le cas du lissage grossier.

L'erreur d'entraînement est similaire à celle obtenue pour R4 dans ce cas, mais l'erreur de validation est assez mauvaise. Cela signifie que le nombre de contraintes n'est pas suffisant pour permettre une bonne généralisation. Entrainer R10 avec 400 stimuli améliore la généralisation, mais il semble falloir plus de stimuli pour obtenir une généralisation satisfaisante.

4.4.2 Comportement global du réseau

Pour analyser le comportement global du réseau, on regarde les activités des nœuds internes. Il s'avère que les 2 nœuds internes du réseau entraîné peuvent présenter des comportements très distincts de ceux du réseau-objectif en général (cf. figure 4.11), même lorsque l'entraînement de R4 réussit avec 400 stimuli et dans le cas du lissage grossier (conditions les plus favorables). Il semble donc qu'un grand nombre d'échantillons d'entraînement ne constitue pas des contraintes suffisamment fortes pour obtenir la convergence de l'activité globale.

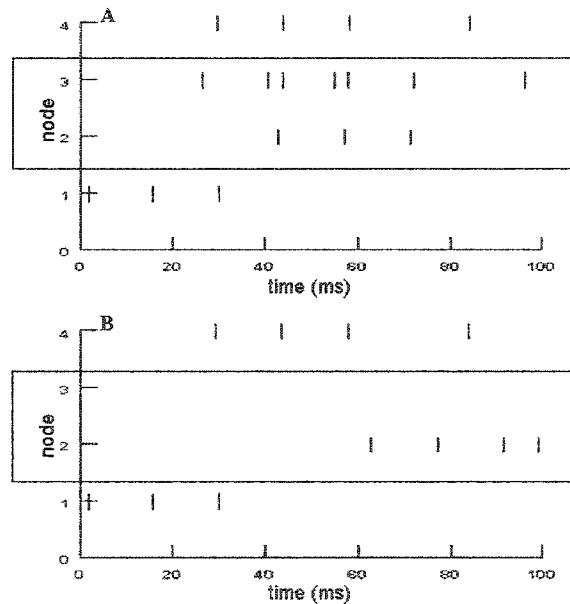


Figure 4.11 : Comparaison des activités des deux nœuds internes dans R4 entre le réseau entraîné et le réseau-objectif.

On voit que les trains de pulses des nœuds internes (2 et 3) peuvent être radicalement différents : ici, pour obtenir la même sortie, un des neurones internes est silencieux. Les seules entrées et sorties ne sont donc pas des contraintes suffisantes pour faire converger globalement le comportement du réseau, même de petite taille.

Dans le cas de la modélisation d'un tissu, on veut que l'activité globale du réseau entraîné respecte topographiquement celle du réseau-objectif. Pour tester cela, on peut par exemple découper la surface du tissu en un pavage et évaluer les activités des

neurones dans chaque élément du pavage, pour comparer les activités entre les deux réseaux entraîné et objectif. Cela pourrait permettre d'essayer de reproduire une activité globale relative à l'objectif 3 à partir de l'activité en certains points, au niveau de densités d'activité et non des trains de pulses précis (cas du lissage grossier).

On teste ainsi R10 avec un pavage particulier pour comparer les activités de manière topographique les activités des réseaux entraîné et objectif (cf. figure 4.12). On optimise le pavage (i.e. la correspondance entre les neurones et l'élément du pavage dans lequel il se trouve) puisque l'architecture utilisée est entièrement connectée et que tous les nœuds internes sont interchangeables. Une telle optimisation du pavage pose le problème dans le cas d'une modélisation de tissu biologique où l'activité globale n'est pas connue (ou alors il faut la prédire en fonction des activités connues en certains points proches). Choisir un pavage fixé sur l'architecture à priori peut ne pas donner de bons résultats, mais si l'architecture est inhomogène, chaque nœud interne est « particulier » dans le réseau et cela peut contribuer à obtenir une convergence globale de l'activité.

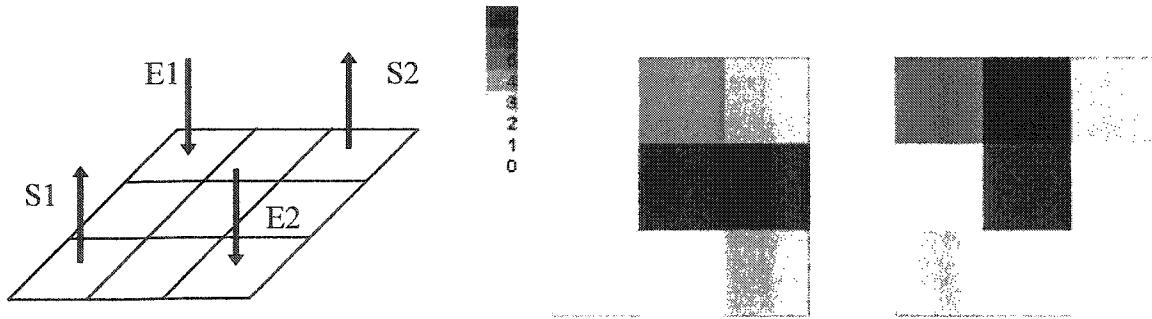
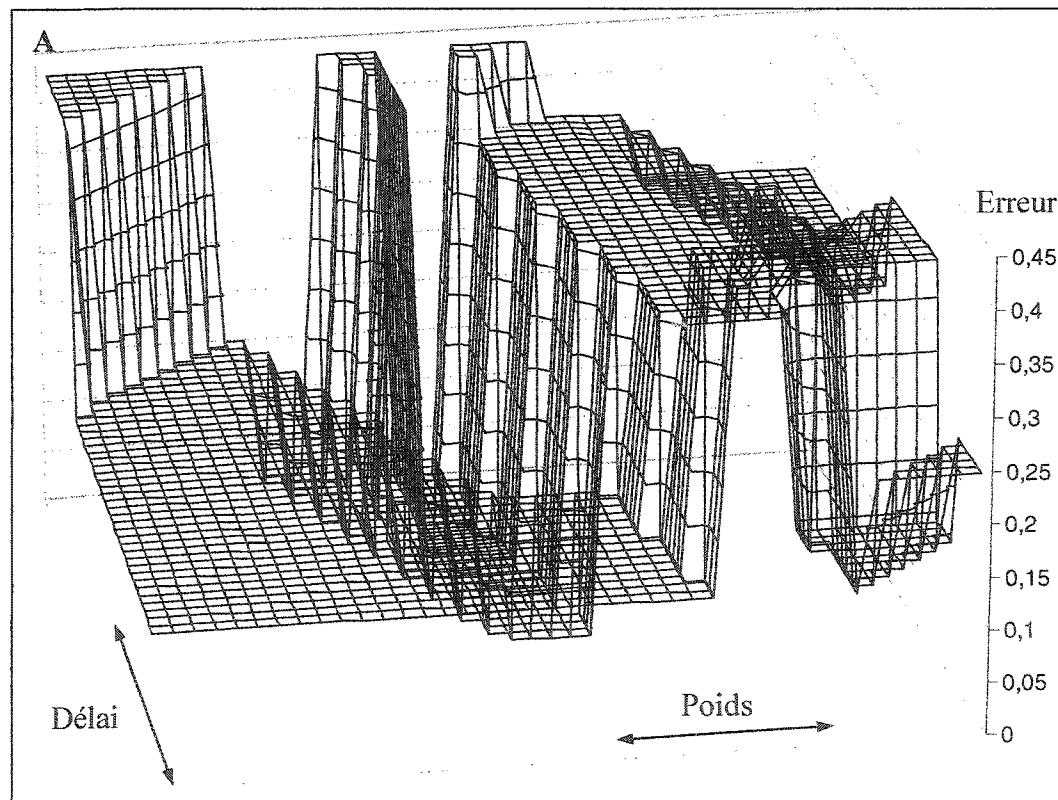


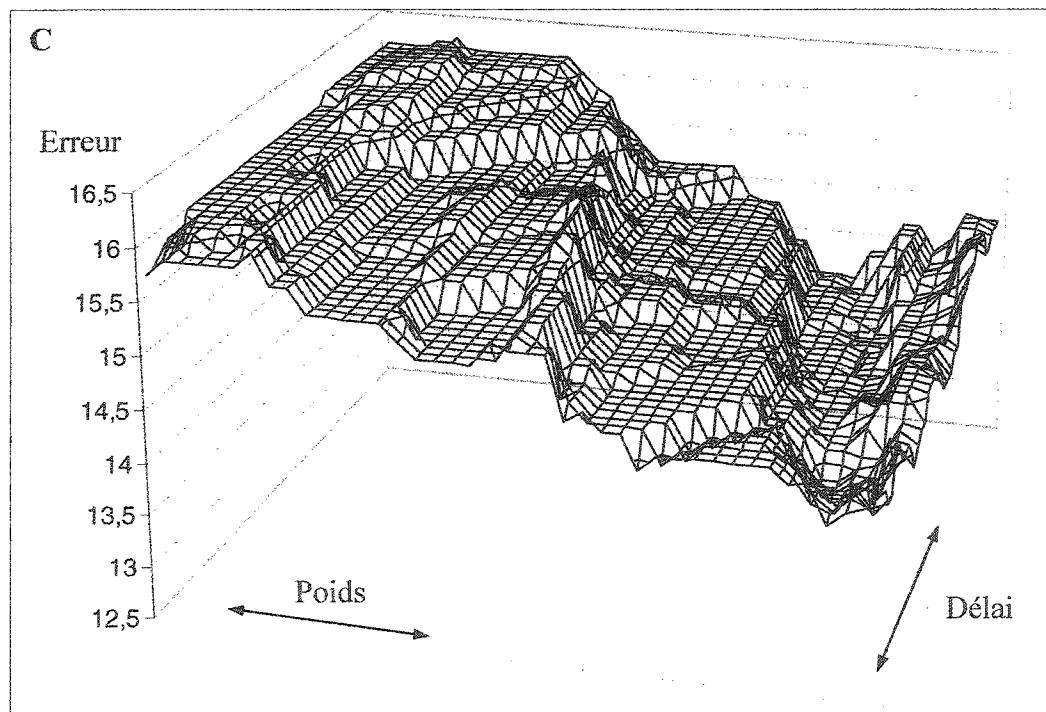
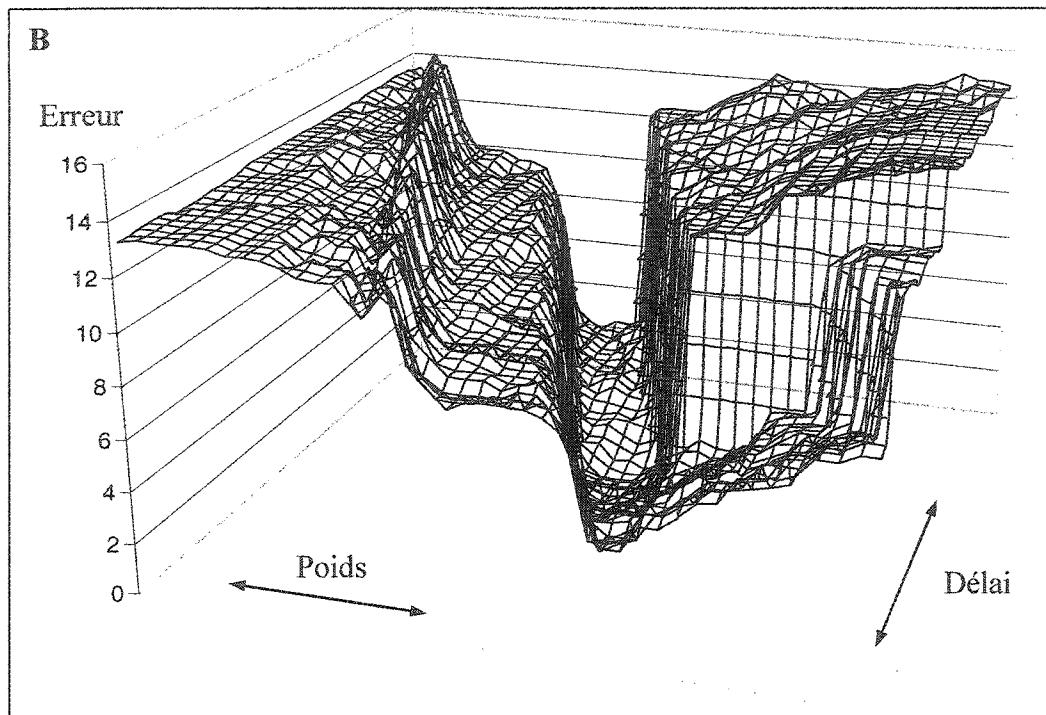
Figure 4.12 : Comparaison topographique des activités résultat et objectif.

On a déterminé les meilleures positions des neurones sur le pavage pour que les moyennes des densités d'activité du réseau entraîné soient le plus proche possible de celle du réseau-objectif (avec un pavage choisi) dans chaque élément pour l'ensemble des stimuli. On fixe dans ce pavage les entrées (E_1 et E_2) et les sorties (S_1 et S_2) et seules les positions des neurones internes sont ajustées. Plus l'élément est sombre, les neurones présents dans l'élément ont émis de pulses durant la stimulation. On constate que malgré cela, les activités globales peuvent être topographiquement distinctes pour de nombreux stimuli d'entraînement (figure de droite).

4.4.3 Analyse des fonctions d'erreur

D'autres méthodes d'exploration du domaine de variation des paramètres ont été envisagées mais elles se sont révélées inefficaces, car la nature même des RNP limite toute exploration locale : cela est illustré par les variations de la fonction d'erreur dans le cas du lissage grossier pour R4 lorsque l'on fait varier le poids synaptique d'une connexion n'impliquant pas le neurone de sortie (cf. figures 4.13). L'erreur pour un seul stimulus (A) se révèle très irrégulière, mais cela est moins net pour 40 stimuli (B), ce qui fait que la convergence est plus facile pour un nombre minimal de stimuli (quelques dizaines). Lorsque l'on s'écarte de l'objectif (C) même légèrement, l'erreur devient vite élevé et reste relativement irrégulière. La présence de nombreux minima locaux et d'irrégularités illustre la difficulté de trouver le minimum de cette fonction d'erreur.





Figures 4.13 : Variations locales de l'erreur en fonction des paramètres synaptiques d'une connexion de R4 avec le lissage grossier.

Le poids varie sur une plage de ± 0.2 et le délai sur $\pm 1\text{ms}$, pour la même connexion synaptique de R4 dans chaque graphique (cette connexion n'implique pas le neurone de sortie). Le centre du graphique correspond aux paramètres du réseau-objectif pour les figures A et B, et à une position très proche dans le domaine de variation des paramètres synaptiques ($\pm 0,05$ pour les poids et $\pm 0,5\text{ms}$ pour les délais) pour la C.

A : On voit que l'erreur est très irrégulière et que ses variations sont brutales. Cela illustre le problème d'optimiser les RNP : même une petite variation des paramètres synaptiques peut engendrer l'apparition ou la disparition d'une ou plusieurs pulses, à cause des temps de réfraction ou du seuil qui régissent le comportement des neurones. B : L'erreur cumulée pour 40 stimuli est plus lisse, même si en affinant la résolution du graphique on peut déceler des irrégularités à petite échelle. Il subsiste cependant des variations brutales, ce que l'on pensait éviter à priori en lissant beaucoup les trains de pulses avec le lissage grossier. Ce n'est malheureusement pas le cas et cela traduit la difficulté d'entraîner les RNP même au niveau des densités d'activités.

C : L'erreur cumulée pour les 40 stimuli est ici très élevée ($>0,5$ par stimulus) même si l'on se situe près des paramètres optimaux dans le domaine de variation des paramètres synaptiques. On voit que l'erreur reste irrégulière à petite échelle et qu'elle présente de nombreux minima locaux. Sur cette plage de variation des paramètres, le minimum se situe loin du centre du graphique, ce qui correspond à une position plus éloignée des paramètres optimaux que la position centrale du graphique, dans le domaine des paramètres synaptiques. Il existe également des plages de variation des paramètres où l'erreur est constante (par exemple quand le neurone de sortie est licencieux, l'erreur est égale à 1). Enfin, seuls 2 paramètres varient ici et la surface de \mathcal{R}^9 correspondant à l'erreur pour R4 avec ses 18 paramètres sur laquelle on se « déplace » pour chercher un minimum est bien plus complexe vue dans l'espace \mathcal{R}^9 .

On retrouve les mêmes phénomènes avec le lissage fin, avec plus d'irrégularités pour la même résolution des graphiques, et la fonction d'erreur lorsque l'on s'éloigne légèrement de l'optimum devient très élevée et irrégulière, avec 40 stimuli pour R4 (cf. figure 4.14). Les variations de la pente de la fonction sont aussi beaucoup plus abruptes, les bassins correspondants aux minima locaux sont bien plus localisés et les plages où la fonction est constante sont plus vastes, ce qui traduit la plus grande difficulté de trouver un minimum de cette fonction et donc de réussir l'optimisation, même avec un grand nombre de stimuli. Notamment, cela compromet plus encore l'exploration locale de cette fonction pour effectuer rechercher un chemin vers le minimum. De plus, lorsque le nombre de paramètres augmente, la surface se place dans un espace de très grande

dimension, ce qui complique encore plus, et ce qui justifie l'emploi de l'algorithme génétique.

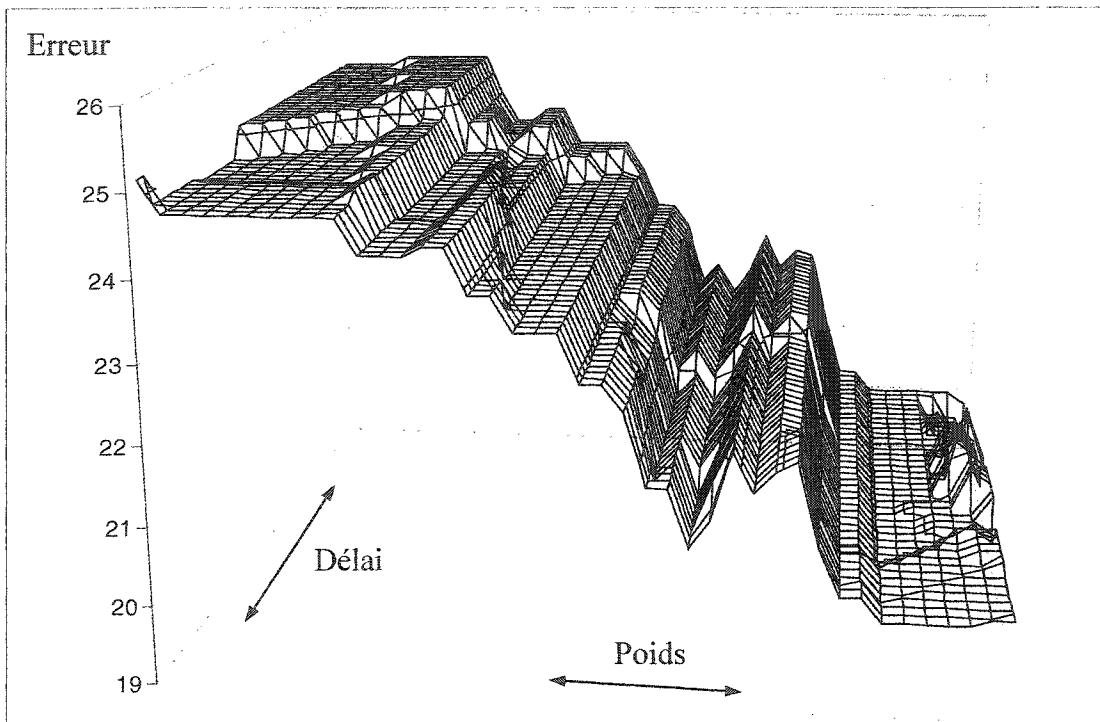


Figure 4.14 : Variations locales de l'erreur pour 40 stimuli.

Cette figure est le pendant de la figure 4.13.C pour le lissage fin. On voit que les variations de l'erreur sont plus rapides et brutales que pour le lissage grossier, et que les plages où elle est constante sont plus larges, ce qui complique l'optimisation.

4.5 Discussion

Cette méthode fonctionne donc plutôt bien avec le lissage grossier (au niveau de la rapidité et surtout de la répartition de l'erreur d'entraînement pour chaque stimulus). Dans le cas du lissage fin, il est difficile de faire diminuer le maximum de l'erreur des stimuli d'entraînement à cause de leur répartition particulière due à l'entraînement. Il faudrait améliorer l'algorithme pour déterminer s'il est possible d'obtenir un réseau

entraîné avec une meilleure généralisation dans ce cas. Cela pose le problème de savoir quelle métrique utiliser, car la convergence de l'algorithme semble être assez sensible à ce niveau. De plus, un **grand nombre de stimuli** est nécessaire pour obtenir du RNRN entraîné une généralisation satisfaisante, mais cela ne suffit pas à faire converger l'activité globale. Il paraît difficile d'effectuer le nombre d'enregistrements nécessaires à l'ajustement des paramètres synaptiques sur un même sujet vivant si ce nombre est de l'ordre de plusieurs millions. On doit donc définir d'autres contraintes sur le réseau.

Les problèmes de la **convergence du comportement global** du réseau et de la taille du réseau doivent être étudiés dans le contexte du cortex visuel : par exemple, on pourrait ajouter de nouvelles contraintes sur l'activité globale du tissu (par exemple des densités d'activité conformément à l'objectif n°3), en plus de la connaissance des enregistrements en sortie. En pratique les sites de lecture ne peuvent pas être aussi proches que l'on veut et la densité de neurones est très grande dans le cortex (plusieurs millions par mm³). On pourrait également modéliser par un neurone du modèle un grand nombre de neurones biologiques, ce dont il faut étudier la pertinence. Dans le cas où l'on voudrait singer le comportement d'un tissu biologique avec des enregistrements de certains sites, il faudrait en plus optimiser l'architecture du RNRN (taille, organisation des connexions, etc.).

Dans le cadre de la modélisation d'un tissu biologique, la **pertinence** de cette méthode dépend en grande partie de la **métrique** définie sur les trains de pulses : si un RNRN entraîné copie le réseau-objectif avec une marge d'erreur mais que cette erreur résiduelle transforme l'information véhiculée dans des proportions bien plus grandes (i.e. si la métrique est incohérente avec le codage de cette information), alors le réseau entraîné n'est d'aucune utilité. Or, il semble difficile de construire dès à présent une telle mesure étant données les connaissances actuelles sur ce codage. Se contenter des densités n'est peut-être pas suffisant mais cela peut constituer une option valable pour étudier l'induction de phosphènes. En effet, les protocoles de stimulation correspondants ne nécessitent pas à priori une fine précision sur les trains de pulses envoyées dans le tissu,

et on peut espérer obtenir des résultats en se limitant aux fréquences de trains de pulses (correspondant aux densités d'activité des neurones d'entrée dans ce modèle).

Comme dit dans la partie 3.2, on peut envisager d'entraîner un patron de colonne corticale et reconstruire le tissu grâce à ces patrons, mais cela nécessite de connaître les interactions entre les colonnes et de définir des entrées et sorties sur les patrons (considérés comme des boîtes noires) adéquates pour rendre compte de ces interactions. La fonction des entrées vers les sorties est très complexe puisque ses entrées et sorties sont des trains de pulses, qui plus est dans un ensemble de grande dimension puisque la colonne doit être connectée à un grand nombre de ses voisines. Enfin des problèmes de surentraînement peuvent apparaître dans ce type d'approximation d'une fonction. Plutôt que définir une telle fonction, on peut utiliser des règles locales simples inspirées de la loi de Hebb et de l'apprentissage défini par Kohonen pour atteindre l'objectif n°3 (cf. partie 3.3). Ainsi, la fonction paramétrant l'impact d'une colonne sur ses voisines est calculé au fur et à mesure que l'on présente des stimuli de différents types au réseau en ajustant les poids synaptiques des connexions entre neurones voisins, et ceux-ci se « regroupent » en zones synchrones symbolisant les colonnes.

5 Entraînement de cartes topographiques auto-organisatrices à états liquides

Ce chapitre présente une deuxième méthode pour modéliser le tissu biologique de la couche 4, basée sur deux concepts : les cartes auto-organisatrices et les machines à états liquides; dont on va tester la compatibilité et que l'on veut réunir dans un même réseau pour vérifier si l'auto-organisation dépendamment des stimuli peut constituer un prétraitement de l'information susceptible d'améliorer les performances du réseau en tant que machine à états liquides, en vue de construire une carte topographique auto-organisatrice à états liquides (CTAOEL).

5.1 *Les cartes topographiques auto-organisatrices relatives aux signaux visuels*

À la suite des CTAO initiées par Kohonen, des réseaux auto-organisateurs plus réalistes du point de vue biologique ont été élaborés pour modéliser le cortex visuel. Un des plus aboutis est RF-LISSOM développé par Miikkulainen et son équipe [4], il consiste en un réseau de « neurones » où chacun représente une colonne corticale (cf. figure 5.1) et modélise les connexions latérales des inter-neurones de la couche 4 et une couche de récepteurs modélisant sommairement les cônes de la rétine génère les entrées du réseau. Initialement, chaque neurone est connecté avec tous les neurones à l'intérieur d'un voisinage avec des poids aléatoires (les connexions sont inhibitrices à longue distance et excitatrices à courte distance) et on ajuste les poids à selon une méthode inspirée de l'algorithme de Kohonen (i.e. à l'intérieur d'un voisinage qui réduit au cours de l'optimisation). La répétition d'un stimulus identique favorise pour chaque neurone

certains voisins, ce qui à terme crée une zone réagissant à ce stimulus. Ce modèle rend compte de l'organisation corticale en colonnes d'orientation préférentielle en adaptant les connexions synaptiques latérales et afférentes (cf. figure 5.2).

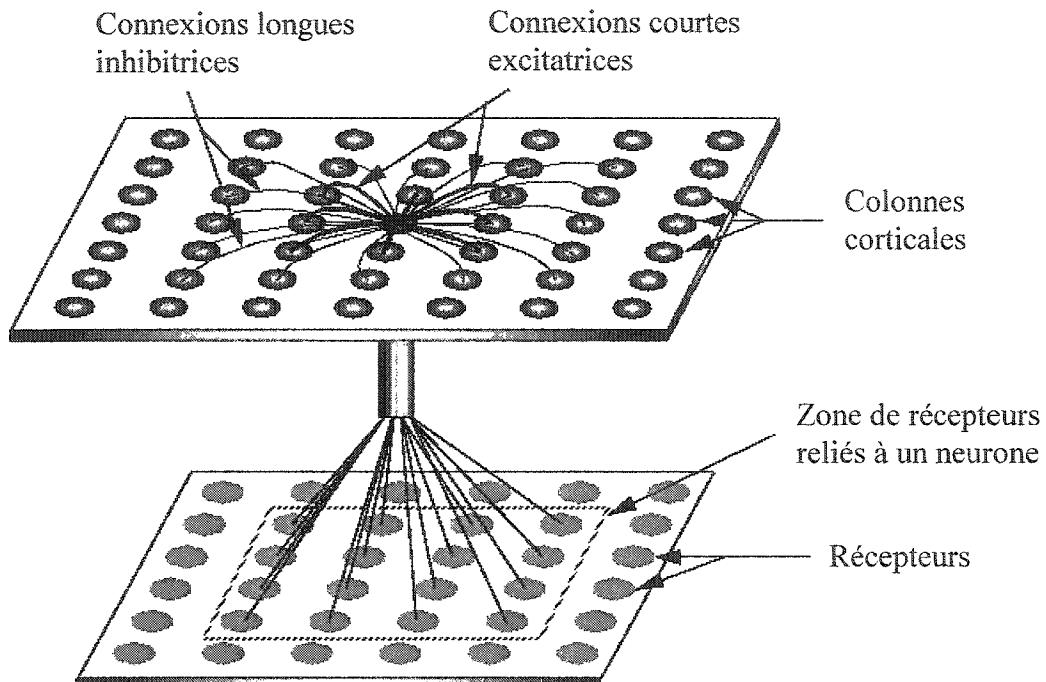


Figure 5.1 : Architecture du modèle RF-LIISOM. ([4])

Une couche de récepteurs simulant sommairement la rétine est reliée au réseau de « neurones » (colonnes corticales) par des connexions afférentes. Les « neurones » sont reliés à courte distance par des connexions excitatrices et à longue distance par des connexions inhibitrices. Chaque « neurone » est relié à toutes les cellules d'une portion du champ récepteur (la correspondance topographique étant évidente).

On se base sur la formule de Hebb pour ajuster les poids des connexions dans notre RNRP (les délais restent inchangés), à la manière du modèle RF-LIISOM : si l'activité d'un neurone dépasse un certain seuil, on va favoriser parmi ses sorties celles correspondant à des neurones d'activité synchrone (cf. équation 5.1). À noter que la formule de Kohonen dérive aussi de la formule de Hebb.

$$\forall j \in Dest_i, \Delta\omega_{ij} = \beta \times K(Act_i, Act_j)$$

$$\forall j \in Dest_i, \omega_{ij} = (\omega_{ij} + \Delta\omega_{ij}) \times \frac{C}{\sum_j (\omega_{ij} + \Delta\omega_{ij})}$$

Équation 5.1 : Formule d'ajustement des poids synaptiques des connexions latérales.
Dest_i correspond aux neurones cibles reliés au neurone i, Act correspond à l'activité du neurone et K est une fonction-noyau de corrélation temporelle entre les activités des neurones : elle est positive et augmente lorsqu'une pulse de neurone j coïncide dans le temps juste après une de i (de l'ordre de 2ms). β est un coefficient de normalisation et il est du signe de ω_{ij} , de sorte que la valeur absolue du poids augmente (hors renormalisation). On renormalise ensuite les poids de sorte que le module du vecteur des poids soit constant (égal à C).

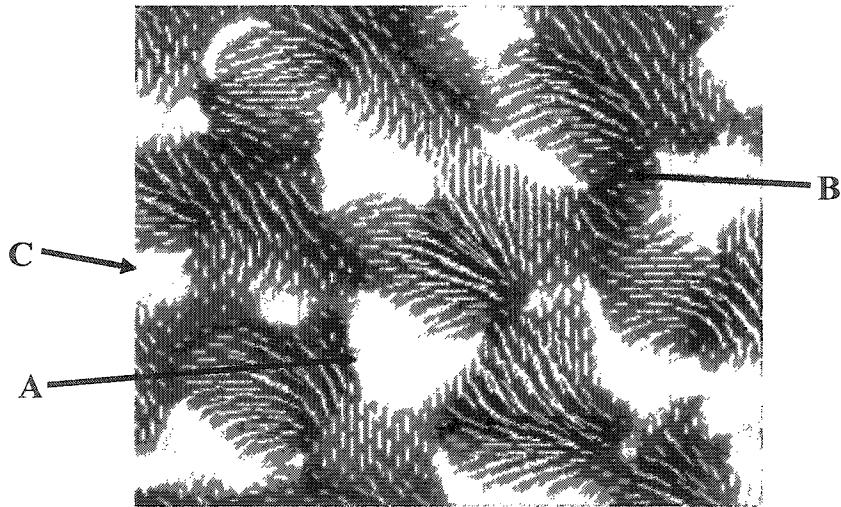


Figure 5.2 : Carte topographique corticale obtenue par RF-LISSOM. ([4])
Cette carte topographique obtenue par le modèle RF-LISSOM montre l'organisation du cortex en colonnes d'orientation préférentielle (correspondant aux traits clairs). On retrouve des variations des orientations observées expérimentalement : par rupture brutale (A), présenter des spirales (B) ou encore varier de manière quasi-linéaire (C, de haut en bas) [27].

Il s'agit donc de transposer ces résultats dans un réseau d'architecture moins « cristalline » et utilisant des neurones à pulses à la place des neurones sigmoïdaux. Le point important est de garder des connexions inhibitrices à longue distance et excitatrices à courte distance, conformément aux tailles respectives des arborescences des cellules

stellaires de chaque type (cf. figure 2.8). On va mettre à jour régulièrement les poids des connexions synaptiques latérales et afférentes durant des intervalles de comptage de durée fixe (double des durées entre deux mises à jour successives). La fonction K de l'équation 5.3 est définie pour évaluer la synchronisation entre les activités des deux neurones de chaque connexion (cf. figure 5.3).

Ainsi, le RNRN devrait s'auto-organiser de sorte que des stimuli semblables excitent des aires voisines, à la manière de l'organisation en colonnes, sans toutefois arriver au même niveau de détail (cela nécessite à priori un grand nombre de neurones pris en compte). Une différence majeure d'avec le modèle RF-LISSOM réside dans l'architecture du réseau (on utilise des neurones de style biologique avec des synapses dynamiques avec délai).

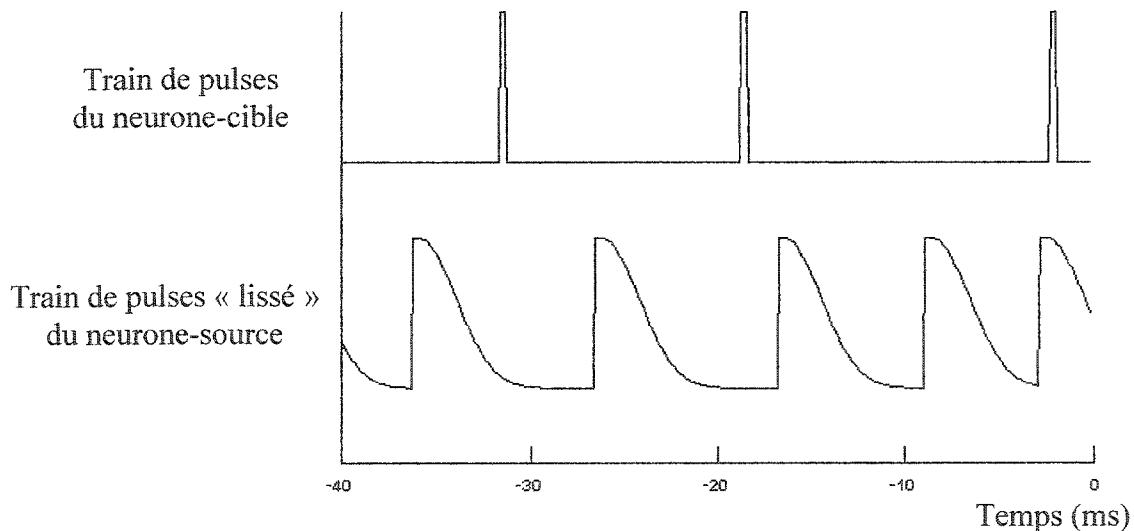


Figure 5.3 : Fonction-noyau intervenant dans la formule d'auto-organisation.
On « lisse » le train de pulses du neurone-source en utilisant un noyau qui décroît dans le temps (en bas, on remonte le temps en allant à gauche), puis on multiplie terme à terme le vecteur obtenu avec le vecteur-activité (le train de pulses en haut) du neurone cible, décalé en tenant compte du délai synaptique. Il résulte que les contributions non nulles correspondent à des pulses du neurone-cible qui sont émises peu après que l'effet de la pulse du neurone-source ait lieu, c'est-à-dire qu'il y ait synchronisation entre les deux neurones. Dans l'exemple ici, la 1^{ère} pulse de la cellule-cible est faiblement corrélée avec le neurone-source, la 2^{nde} ne l'est pas et la 3^{ème} l'est fortement.

On effectue en plus une pondération qui est fonction de la distance entre les neurones source et cible, pour permettre la stabilisation sans devoir ajuster le voisinage de mise à jour, à l'instar de la fonction-chapeau de la figure 3.6. Ici, la renormalisation abaisse les poids non modifiés, on s'affranchit des poids négatifs et on se contente d'une gaussienne que l'on explicitera dans la partie 5.3. Sur un neurone, on voit que certaines liaisons (actives de manière synchrone) sont renforcées au détriment des autres (cf. figure 5.15 dans la partie 5.4 avec les résultats).

5.2 *Les machines à états liquides*

5.2.1 Définition du concept d'états liquides

Ce nouveau concept de calcul appelé en anglais « Liquid-State Machine » (que l'on traduit ici par machine à états liquides - MEL) a été récemment développé Maass et son équipe [18]. Pour calculer n'importe quelle fonction-objectif sur des entrées, une MEL tire parti d'un ensemble de nombreuses fonctions sur ces entrées générées par un filtre adéquat, et une fonction de sortie combine ensuite « simplement » ces fonctions, sans devoir connaître le codage de l'information au niveau du filtre (cf. figure 5.4).

Ainsi une MEL peut être vue comme un « calculateur universel » en temps réel, de même qu'un RNS est un « approximateur universel » de fonctions mathématiques grâce à la rétro-propagation. Le terme « états liquides » réfère au fait que le système ne fonctionne pas avec des états stables mais de manière dynamique : $\tilde{x}(t)$ passe par un grand nombre d'états (cf. figure 5.5) sous l'influence des entrées. Pour exemple de F, Maass propose un perceptron à une couche entraîné par une règle d'apprentissage (cf. partie 5.1.2).

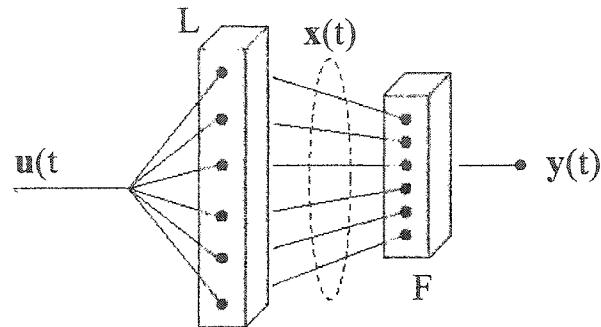


Figure 5.4 : Schéma général d'une MEL à une entrée et une sortie. ([18])

Le filtre L transforme le vecteur-entrées $\bar{u}(t)$, variable dans le temps t , en un vecteur $\bar{x}(t)$ qui dépend non seulement de $\bar{u}(t)$, mais aussi de façon non linéaire des valeurs $\bar{u}(s)$ antérieures à t : $\bar{x}(t) = (L \bar{u})(t)$ et non $\bar{x}(t) = L(\bar{u}(t))$. En pratique, le vecteur $\bar{x}(t)$ appelé état liquide du système est de très grande dimension par rapport au vecteur $\bar{u}(t)$ et il dépend seulement du passé récent de $\bar{u}(t)$ (L est un filtre à mémoire temporelle). La fonction F (qui peut être sans mémoire) calcule en temps réel la sortie $\bar{y}(t)$ et est entraînée en pratique pour calculer la sortie désirée.

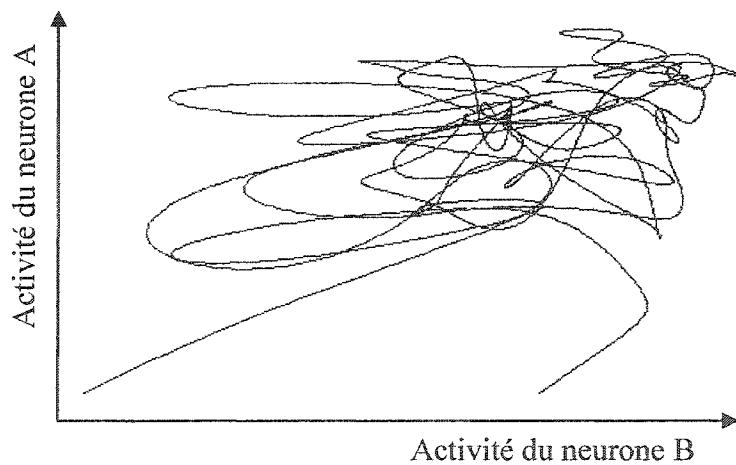


Figure 5.5 : Illustration des états liquides d'un RNRP.

La courbe représente les variations conjointes des activités de 2 neurones A et B (reliés entre eux) : il ne semble pas exister de trajectoire stable ou asymptotique, à la différence de machines à états plus classiques.

Maass a montré qu'un RNP pouvait être en pratique une bonne approximation du filtre L : il calcule au niveau de chaque neurone en temps réel une fonction sur les entrées avec une mémoire à court terme, si bien qu'il effectue en quelque sorte une projection des entrées dans un espace de très grande dimension pour une taille suffisante. Pour garantir sa puissance computationnelle en tant que MEL, un RNP doit respecter certaines conditions sur son architecture et ses connexions synaptiques inspirées du cortex somatosensoriel [20] :

- par référence à l'organisation biologique du cortex [3], on définit un système **excitateur (SE)** et un **inhibiteur (SI)**, chacun ayant des caractéristiques propres (les connexions partant de SI sont inhibitrices et celles de SE sont excitatrices),
- les caractéristiques internes des neurones sont biologiquement réalistes (période réfractaire de 2 ou 3ms suivant que le neurone est dans SE ou SI par exemple); Maass place les neurones sur une grille 3D, ce qui entraîne que leur répartition est homogène,
- la **connectivité** des neurones doit être éparse, en particulier pour les connexions avec les plus proches voisins, et les probabilités de connexions dépendent à quel système chaque neurone appartient (tableau 5.1.A),
- les **synapses** sont **dynamiques**, c'est à dire que leur sensibilité varie en fonction de leur activité, avec des constantes de dépression et de facilitation respectant des distributions précises (cf. tableau 5.1.B), tandis que les délais synaptiques sont choisis uniformément entre 0 et une valeur maximale.

La puissance computationnelle de la MEL dépend en grande partie de l'organisation des connexions synaptiques, selon Maass. On aurait donc besoin de moyens d'évaluation de cette puissance, car la CTAOEL doit conserver son caractère de MEL même si les poids des synapses sont modifiés par l'auto-adaptation et malgré son architecture « réaliste biologiquement ». On se contentera pour le moment de respecter des tendances définies par Maass (densité de connexions, importances relatives des poids, etc.) pour construire la CTAOEL (cf. partie 5.3) et on analysera dans la partie 5.4 le comportement du réseau pour évaluer ses performances comme MEL.

A \ B	SE	SI
SE	0,3	0,2
SI	0,4	0,1

Tableau 5.1.A : Coefficients relatifs aux probabilités de connexion de 2 neurones.

Dans la formule de calcul de la probabilité apparaît le coefficient C qui varie suivant l'appartenance de A et B à SE ou SI, et qui est explicité dans le tableau 5.1.A. A et B sont respectivement le neurone-source et le neurone-cible de la connexion. La fonction "d" de A et B correspond à la distance entre les neurones et λ est du même ordre que la distance inter-neurones et intervient sur la densité de connexions et sur la distance moyenne des « liaisons » synaptiques.

$$P(\text{connexion}(A, B)) = C \times e^{-d(A, B)^2 / \lambda^2}$$

Équation 5.2 : Formule de la probabilité de connexion de 2 neurones.

A \ B	SE	SI
SE	0,5 1,1 0,05	0,05 0,125 1,2
SI	0,25 0,7 0,02	0,32 0,144 0,06

Tableau 5.1.B : Distributions des paramètres synaptiques.

Les données dans chaque case correspondent dans l'ordre aux moyennes des poids, des temps caractéristiques (en s) de dépression et de facilitation relatifs aux connexions synaptiques. Les écarts-type sont pris égaux à la moitié de chaque valeur (en excluant les valeurs négatives). A et B sont respectivement les sources et cibles des connexions. Le modèle de synapse est décrit dans [19].

5.2.2 La règle d'apprentissage p-delta

La règle p-delta [2] a été développée pour entraîner un perceptron à approximer n'importe quelle fonction mathématique, comme la rétro-propagation pour les RNPA. Cette règle reste valide pour entraîner un neurone à pulses : on définit ainsi pour chaque échantillon en entrée un niveau d'activité en sortie et on met à jour les poids synaptiques en entrée du neurone en comparant l'objectif avec la densité de pulses obtenue en sortie, en tenant compte d'une marge définie (cf. équation 5.2). On fixe les délais synaptiques à 0ms. L'introduction d'une marge assurant la convergence de l'algorithme a été inspirée par les machines à vecteur de support. Enfin, les objectifs et les résultats utilisés dans l'équation 5.2 correspondent dans le cas des neurones à pulses aux nombres de pulses émis pendant des intervalles de temps de même durée.

$$\vec{\omega} \leftarrow \vec{\omega} - \eta \left(\|\vec{\omega}\|^2 - 1 \right) \vec{\omega} + \eta \begin{cases} -\vec{x} & \text{si } \hat{o} > o + \varepsilon \text{ et } 0 \leq \vec{\omega} \cdot \vec{x} \\ \vec{x} & \text{si } \hat{o} < o - \varepsilon \text{ et } \vec{\omega} \cdot \vec{x} \leq 0 \\ \mu \vec{x} & \text{si } \hat{o} \leq o + \varepsilon \text{ et } 0 \leq \vec{\omega} \cdot \vec{x} < \gamma \\ -\mu \vec{x} & \text{si } \hat{o} \geq o - \varepsilon \text{ et } -\gamma < \vec{\omega} \cdot \vec{x} \leq 0 \\ \vec{0} & \text{sinon} \end{cases}$$

Équation 5.3 : Formules de mise à jour des poids synaptiques de la règle p-delta.
Le vecteur ω correspond aux poids synaptiques en entrée et le vecteur x désigne les entrées correspondantes. \hat{o} est le résultat (densité de pulses obtenues en sortie pour les entrées x) et o l'objectif à atteindre, ε est l'erreur tolérée, γ correspond à la marge devant être respectée pour assurer la convergence de l'entraînement et μ est un facteur entre 0 et 1 relié à l'importance de respecter la marge (généralement fixé à 0,5). η est le facteur d'apprentissage. La multiplication des vecteurs est un produit scalaire. Durant la mise à jour, on ajoute un premier terme qui fait tendre asymptotiquement la norme du vecteur ω vers 1 et un deuxième terme d'apprentissage dépendant de la comparaison du résultat \hat{o} avec l'objectif o .

D'une part, il faut adéquatement choisir les méta-paramètres d'entraînement des neurones d'extraction (erreur tolérée en sortie, marge et son importance, intervalle de comptage des pulses en sortie), car ils influent sur la classe des densités de pulses qu'il peut approximer en sortie (cf. figure 5.6). Leur temps de réfraction sera du même ordre que celui des autres neurones du réseau (2-3ms) pour être réaliste, mais il pourrait être également ajusté. Comme on appliquera cet entraînement à des objectifs de type tout-ourien et on n'aura pas besoin de trop affiner les choix de ces paramètres.

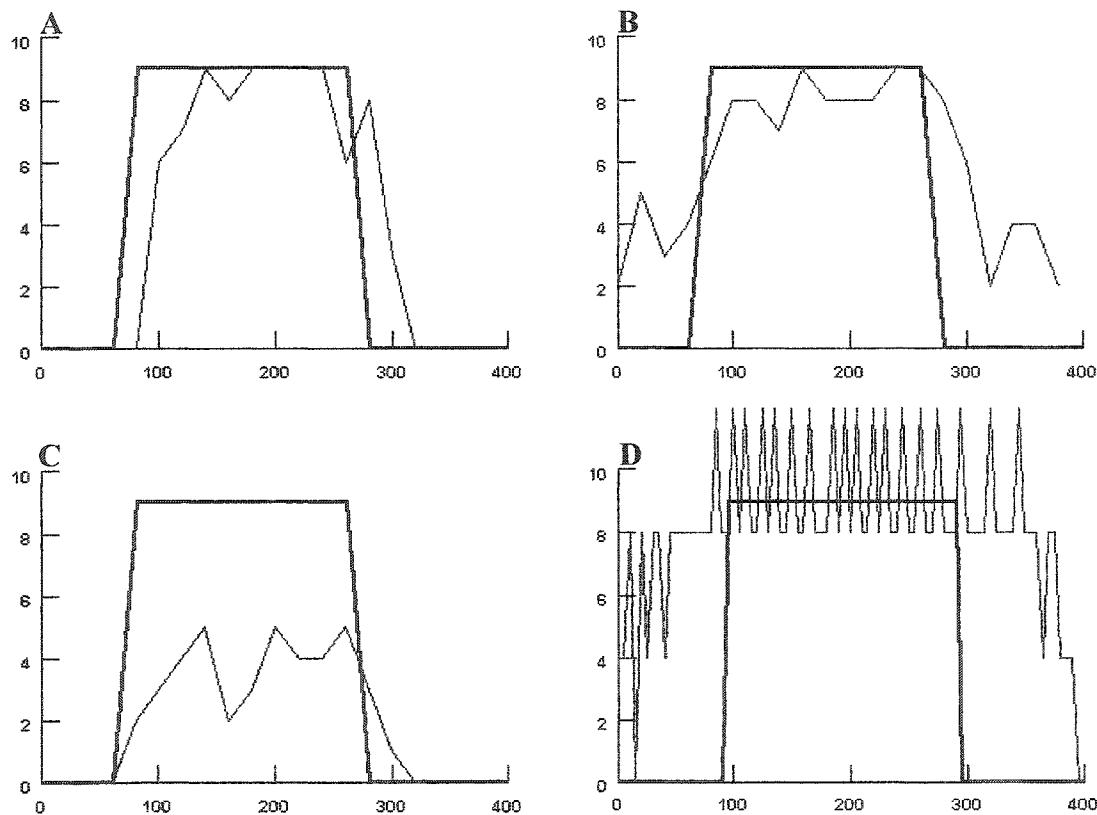


Figure 5.6 : Influence des méta-paramètres d'entraînement de la règle p-delta.
Le graphe A correspond à un entraînement de 20 cycles avec une erreur tolérée de 1, une marge de 50 et des intervalles de comptage de 20ms (avec en abscisses le temps et en ordonnées les densités de pulses). L'objectif défini ici (en trait gras, le résultat en trait fin) correspond soit au silence soit à la saturation du neurone I&E en sortie. Si l'erreur tolérée est trop élevée (B avec 2) ou si la marge est trop faible (C avec 0), la fonction du neurone entraîné ne respecte pas bien l'objectif. Si l'intervalle de comptage des pulses est trop court (D avec 5ms), la sortie est plus sensible à optimiser et peut ne pas bien stabiliser.

Le succès de l'entraînement dépend aussi de la « diversité » des entrées, i.e. de l'existence d'un jeu de valeurs pour les poids combinant les entrées pour obtenir en sortie l'objectif désiré (cf. figure 5.7 et 5.8). La MEL doit donc générer à partir des entrées des « fonctions » aussi « variées » que possible, qui sont envoyées ensuite dans les neurones d'extraction. L'entraînement est donc d'autant plus facile que la MEL est puissante, ce dont on peut avoir une idée en analysant les activités des neurones du réseau.

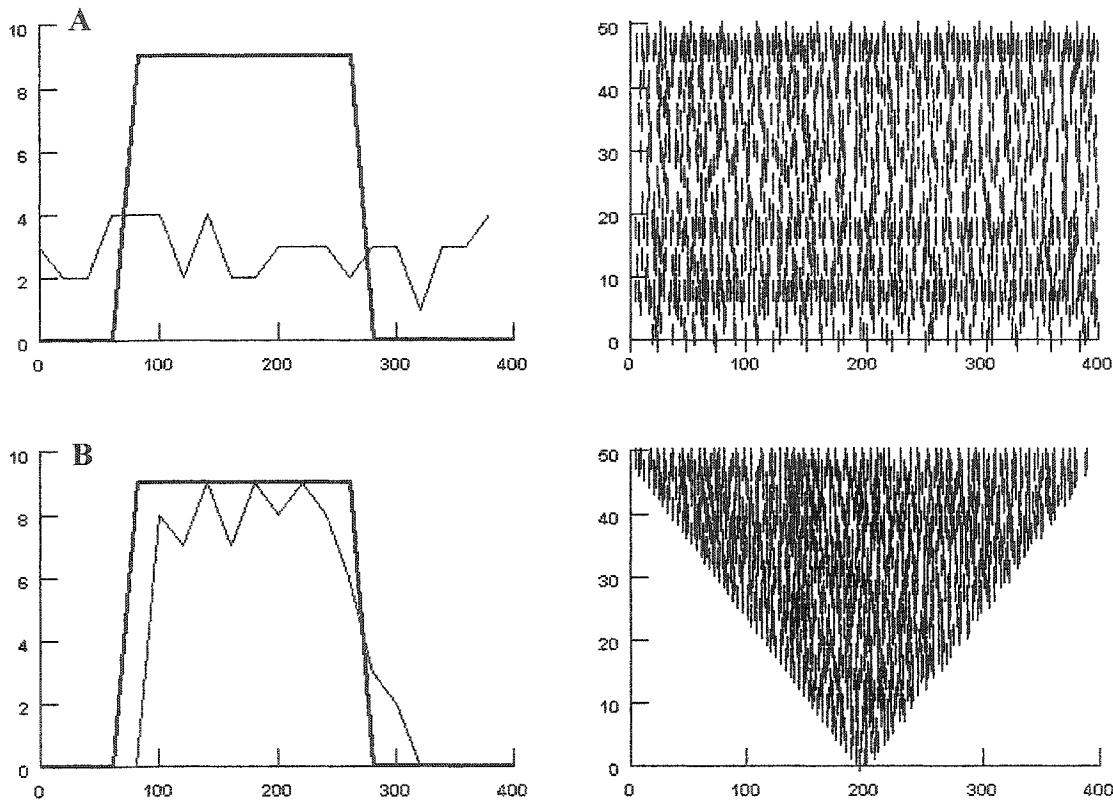


Figure 5.7 : Influence des entrées sur l'entraînement d'un neurone à pulse.
On a entraîné le neurone en présentant 50 fois des trains de pulses aléatoirement espacés pour A (à droite, on voit les 50 trains de pulses qui sont les entrées) et des trains d'entrées respectant des distributions centrées à $t=200\text{ms}$ et d'étalement variable pour B (idem). La fonction-objectif en gras sur les graphes de gauche (avec le temps en abscisses) et la sortie du neurone d'extraction est en trait fin. On voit que dans le cas A l'algorithme parvient mieux à trouver des poids synaptiques pour approcher l'objectif que dans le cas B : les pulses étant situées aléatoirement dans les trains en entrée, une combinaison généralisable des poids synaptiques est quasi impossible à trouver. À noter que la densité-objectif n'est pas facile à obtenir pour un neurone à pulses.

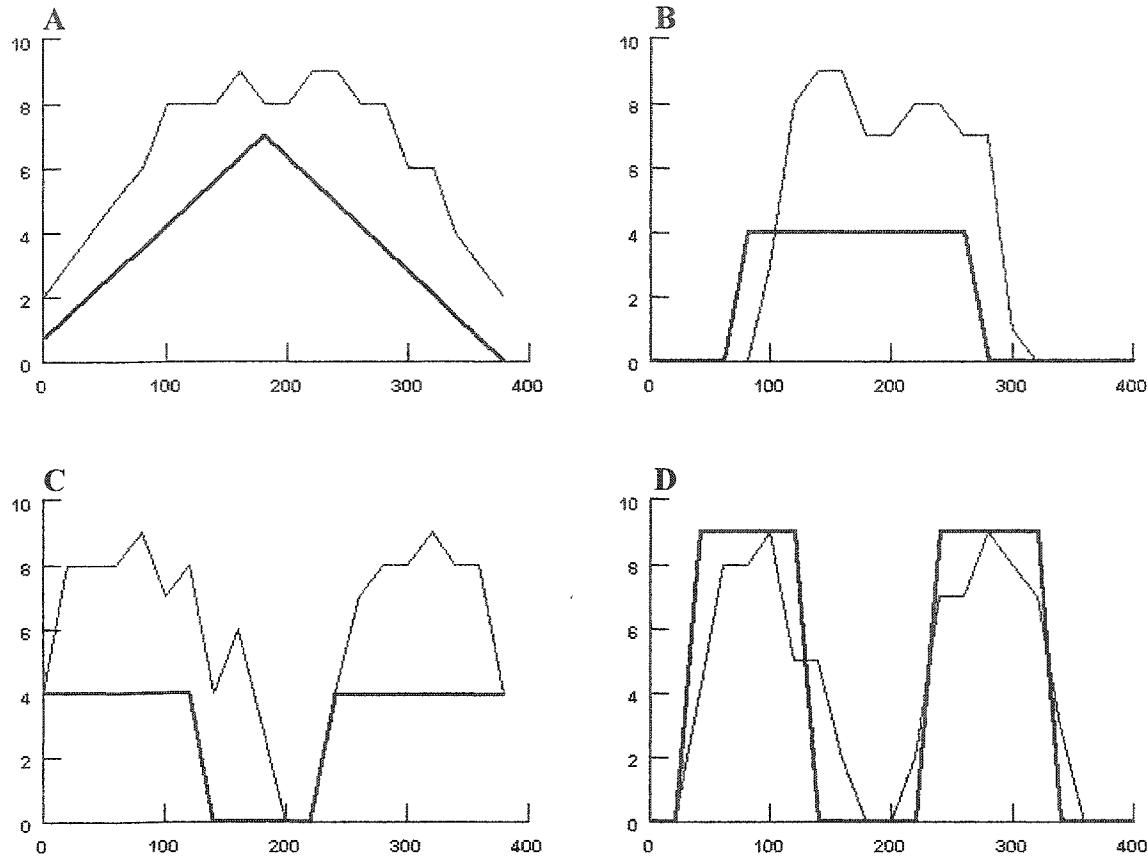


Figure 5.8 : Classes des fonctions-objectifs que l'on peut approcher en sortie d'un neurone à pulses.

Le neurone à pulses est plutôt bien adapté pour approcher en sortie des fonction tout-ou-rien : l'ordre de pulses en entrée est important et lorsque les trains présentent un caractère aléatoire, il n'est pas possible de contrôler la sortie à la pulse près (ce qui est demandé ici). Ainsi, la forme triangulaire est respectée dans le cas A, mais l'erreur à chaque cycle excède la limite tolérée. Les cas B et C illustrent le fait qu'il est plus facile d'obtenir du tout-ou-rien en sortie. La figure D montre par contre que la sortie peut être relativement précise dans le temps.

5.3 Protocole de stimulation

On va construire un RNP de 1000 neurones I&E répartis aléatoirement dans un espace 3D de $1000 \times 1000 \times 150 \mu\text{m}^3$ (pour une densité de 6000 neurones par mm^3 environ, soit 100 fois moins qu'en réalité) en veillant à avoir une certaine homogénéité. On sépare les neurones en deux groupes : un système excitateur (SE) et un inhibiteur (SI), SI représentant 20% environ du total. La période réfractaire des neurones de SE (resp. de SI) suit une distribution normale de moyenne 3ms (resp. 2ms) et d'écart-type 0,1ms (resp. 0,2ms); la durée caractéristique de fuite avec en moyenne 30ms et un écart-type de 2ms pour tous. On crée des liaisons courtes excitatrices et des liaisons longues inhibitrices comme dans le modèle RF-LISSOM, avec les formules de probabilité des équations 5.4 (modification pour SI \rightarrow SE, cf. tableau 5.2 et figure 5.9 et 5.10).

A \ B	SE	SI
A		
SE	C=0,3 $\lambda=200$	0,2 200
SI	0,3 300	0,25 300

$$P(\text{connexion}(A, B)) = C \times e^{-d(A, B)^2 / \lambda^2}$$

$$P(\text{connexion}(A, B)) = C \times e^{-d(A, B)^2 / \lambda^2} \times \frac{d(A, B)^2}{\lambda^2}$$

Tableau 5.2 : Coefficients des probabilités des connexions latérales.

Équations 5.4 : Formules des probabilités des connexions latérales.

La 1^{ère} formule concerne toutes les connexions partant d'un neurone de SE et la 2^{ème} formule pour celles partant d'un de SI. Les paramètres C et λ pour chaque type de connexion sont explicités dans le tableau (A est la source et B la cible), et ils ont été choisis pour avoir des connexions inhibitrices plus longues que les connexions excitatrices en gardant le nombre de connexions du même ordre de grandeur qu'avec Maass (un neurone de SE a approximativement 35 connexions, et 45 pour un de SI).

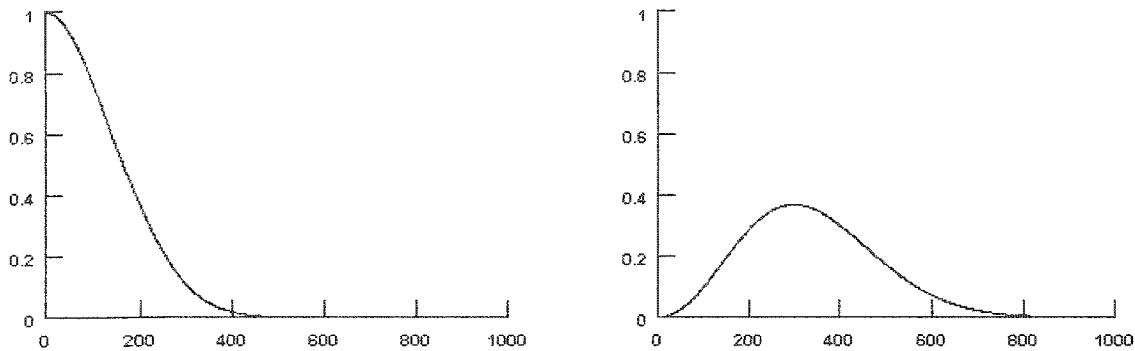


Figure 5.9 : Graphes des distributions des probabilités de connexion.

Ces graphes correspondent aux formules des équations 5.4 (respectivement la 1^{ère} à gauche avec $C=1$ et $\lambda=200 \mu\text{m}$ et la 2^{nde} à droite avec $C=1$ et $\lambda=300 \mu\text{m}$; en abscisses la distance entre A et B en μm). Ainsi, la longueur moyenne des connexions d'une cellule de SE est de l'ordre de $150 \mu\text{m}$ et celle d'une cellule de SI environ $350 \mu\text{m}$.

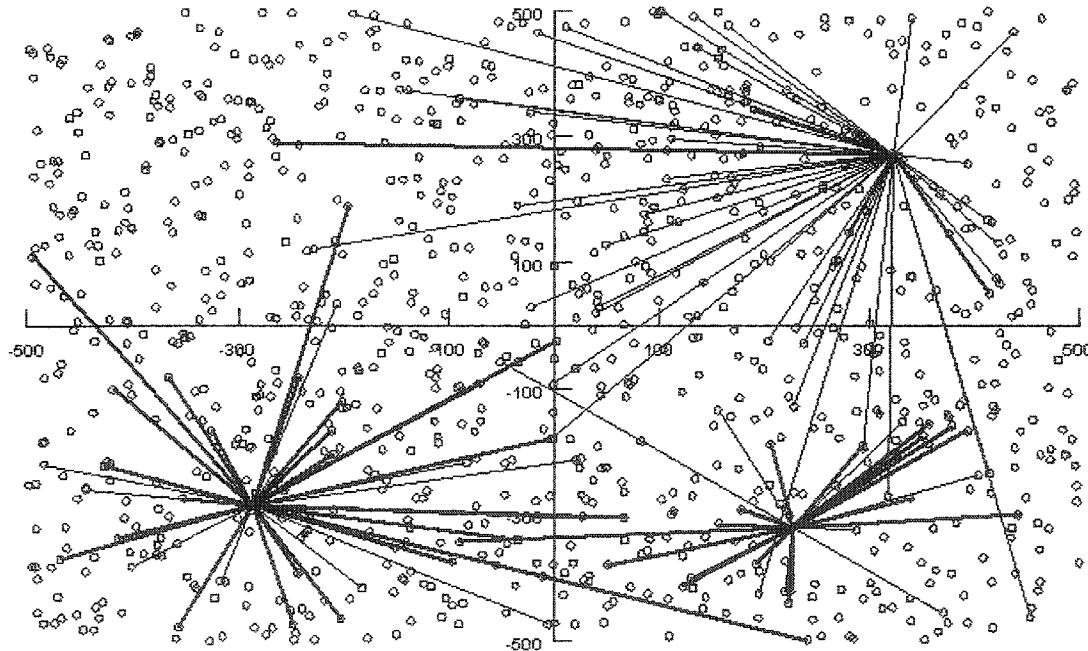


Figure 5.10 : Exemple d'architecture du RNP.

Chaque rond représente un neurone. Un tel réseau de 1000 neurones comporte environ 31000 connexions. On voit sur ce réseau les connexions d'un neurone exciteur (en bas à droite) avec en gras les connexions $SE \rightarrow SE$ (20 connexions) et en trait fin $SE \rightarrow SI$ (5); celles d'un neurone inhibiteur (en haut à droite) avec en gras $SI \rightarrow SI$ (4) et en trait fin $SI \rightarrow SE$ (38); et celles d'un récepteur (en bas à gauche, c'est une cellule qui génère les entrées du RNP, voir plus loin) avec en gras $R \rightarrow SE$ (30) et en trait fin $R \rightarrow SI$ (7).

On modélise les phénomènes de facilitation et de dépression grâce aux formules des équations 2.1. On s'est inspiré des grandeurs relatives des poids et des constantes de dépression et facilitation synaptiques du tableau 5.1.B pour définir des valeurs adéquats pour le modèle de synapse que l'on utilise (cf. tableau 5.3), et les délais sont tirés uniformément dans l'intervalle [0,20]ms. Ces choix seront commentés au début de la partie suivante.

A \ B	SE	SI
SE	0,25 30 10	0,3 10 20
SI	-0,3 20 2	-0,35 10 10

Tableau 5.3 : Valeurs des moyennes des distributions normales régissant les poids et les temps caractéristiques de dépression synaptiques.

A et B sont les sources et cibles des connexions, on a dans l'ordre les temps caractéristiques de dépression sont en ms. Chaque paramètre suit une distribution normale de moyenne donnée dans le tableau et d'écart-type égal au tiers de la valeur absolue de la moyenne. Ces valeurs s'inspirent de celles décrites par Maass qui utilise le modèle de synapse de Tsodyks [19] et on été adapté pour le modèle décrit par les équations 2.1.

Durant les simulations, on adapte les poids synaptiques connexions latérales et afférentes selon la méthode décrite dans la partie 5.2, la fenêtre temporelle dure 200ms et on effectue les mises à jour tous les 100ms. Le paramètre β de l'équation 5.3 est fixé à 1 (trouvé empiriquement de sorte que les modifications ne soient pas trop brutales). Seuls les neurones dont l'activité excède 60% du niveau de saturation sont mis à jour, pour accélérer les calculs et compte tenu que les modifications sont bien moins significatives dans ce cas. La gaussienne de pondération des mises à jour correspond à la formule de l'équation 5.2 avec une distance caractéristique de 70 μ m pour un neurone de SE et de 150 pour un de SI.

Chaque neurone d'extraction possède une période de réfraction de 2ms et une durée caractéristique de fuite de 30ms, il est connecté à tous les neurones du réseau avec des poids aléatoirement tirés dans [-0,5;1] et il est entraîné par la règle p-delta avec les méta-paramètres suivants : l'erreur tolérée vaut 10% de l'objectif maximal (saturation du neurone en émission), la marge d'optimisation est de 30 et le coefficient traduisant son importance est 0,5 (selon Maass). On fixe la durée de comptage des pulses à 200ms et les mises à jour ont lieu tous les 100ms. La décroissance du facteur d'apprentissage est inversement proportionnelle au nombre d'époques d'entraînement.

Les entrées de la CTAOEL sont générées par des récepteurs, chacun couvrant un secteur d'une fenêtre de perception (cf. figure 5.11). Un récepteur est activé si une forme « lumineuse » se trouve dans sa zone de détection et il émet alors des pulses qui sont envoyées vers le RNP. Chaque récepteur est connecté à des neurones du RNP situés dans un voisinage lui correspondant en superposant le RNP et la couche réceptrice, avec une distribution de probabilité gaussienne définie la 1^{ère} formule des équations 5.4. On obtient ainsi un influx monoculaire provenant de ce modèle simplifié de la rétine.

À la différence du modèle RF-LISSOM, la réponse des récepteurs ne dépend pas de l'orientation des formes géométriques (points ovoïdes pour le modèle RF-LISSOM), mais uniquement avec une sensibilité gaussienne fonction de la distance entre la forme et le centre du secteur de détection, et compte tenu d'un temps de relaxation de 2ms. Les stimuli consistent en des barres mobiles « lumineuses » que l'on affiche dans la fenêtre de perception, avec des vitesses et des orientations diverses.

On va par exemple entraîner un neurone d'extraction à reconnaître une certaine orientation moyenne ou encore lui apprendre à différencier les barres verticales des barres horizontales. D'autre part, il s'agit d'observer comment la nature des stimuli présentés affecte l'auto-organisation du réseau : si l'on présente uniquement des barres de même orientation, va-t-il se spécialiser pour les vitesses de déplacement? C'est-à-dire permettra-t-il de mieux discerner les différentes vitesses si seule cette composante varie d'un stimulus à l'autre. Enfin, on veut étudier les relations entre l'auto-organisation et l'entraînement des neurones d'extraction, notamment la compatibilité entre eux.

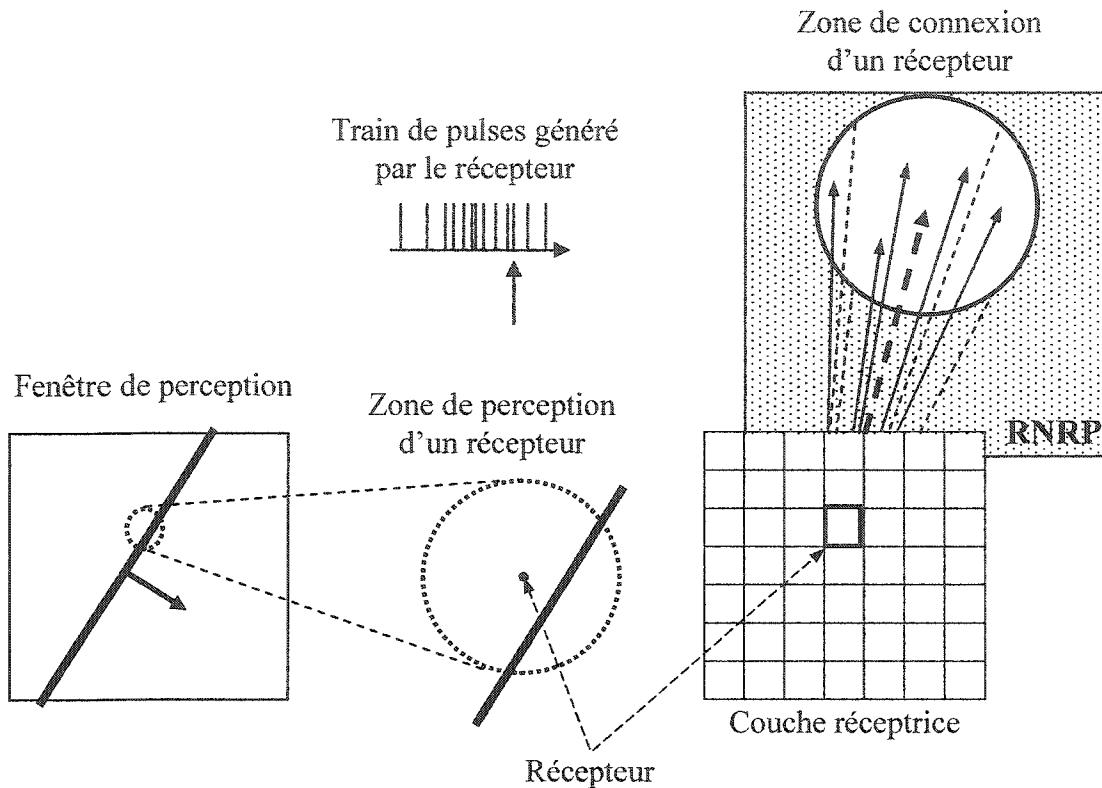


Figure 5.11 : Génération des entrées du RNP par des cellules réceptrices.

On affiche des barres « lumineuses » mobiles (la flèche en gras donne la direction du mouvement) dans une fenêtre de perception (à gauche). Sur la couche réceptrice (7×7), chaque cellule génère des pulses lorsque la barre se trouve dans son secteur, avec une fréquence plus importante lorsque la barre est proche du centre (avec une distribution gaussienne de la probabilité d'émettre une pulse, cf. agrandissement au milieu, et la flèche verticale indique le moment correspondant au dessin agrandi du secteur dans le train de pulses généré). Chaque récepteur est connecté (flèches fines en trait plein dans le dessin à droite) avec des neurones situés autour de la position lui correspondant dans le RNP (flèche en gras et pointillé). La probabilité de connexion correspond à la première formule des équations 5.4 avec $C=0,25$ et $\lambda=300\mu\text{m}$ (correspondant à une moyenne de 50 connexions environ par récepteur). Ces connexions sont excitatrices et sont régies par des synapses statiques avec un poids suivant une distribution gaussienne de moyenne 0,4 et d'écart-type le tiers de cette moyenne, avec un délai tiré aléatoirement dans $[0,10]\text{ms}$.

Les simulations sont effectuées grâce au logiciel NEURON, et durent 200ms avec un pas temporel de 0,1ms.

5.4 Résultats

5.4.1 Comportement général du réseau

On a dit que les paramètres synaptiques ont été choisis de manière que l'activité du réseau soit « biologiquement » réaliste de manière globale. Elle doit augmenter dans des proportions significatives lors de la présentation d'un stimulus, sans être saturée (cf. figure 5.12).

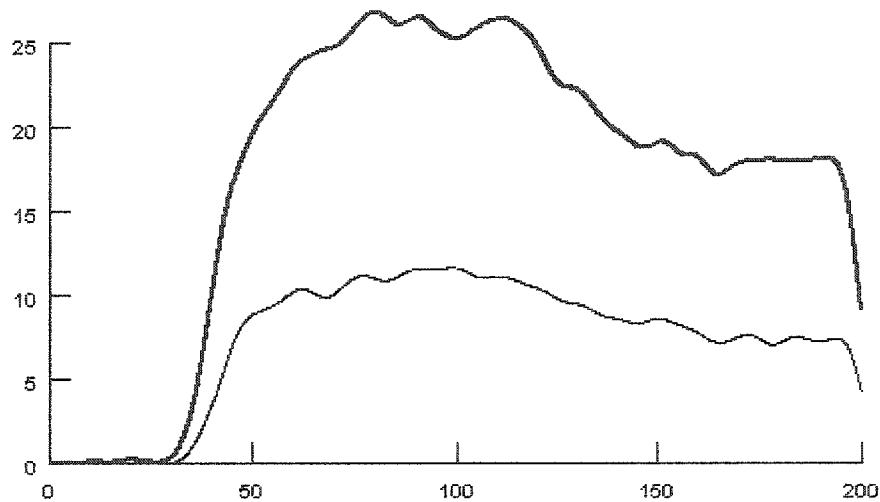


Figure 5.12 : Variation de l'activité globale du RNRN lors d'une stimulation.
L'activité de SE est en gras et celle de SI en trait fin. La stimulation débute à $t=32\text{ms}$ et finit à $t=118\text{ms}$. Pour SE et SI, l'activité est nulle avant, connaît un pic pendant et est entretenue après la stimulation (la chute est due au lissage à $t=200\text{ms}$). Les systèmes ne saturent pas durant la stimulation.

5.4.2 Puissance du RNRN en tant que MEL

Nos choix des paramètres synaptiques définissent un bon caractère de MEL pour le RNRN. Conformément à Maass, on vérifie que des trains de pulses relativement semblables en entrée doivent générer des activités semblables, selon une métrique sur les trains en entrée et une autre sur l'activité (cf. figures 5.13).

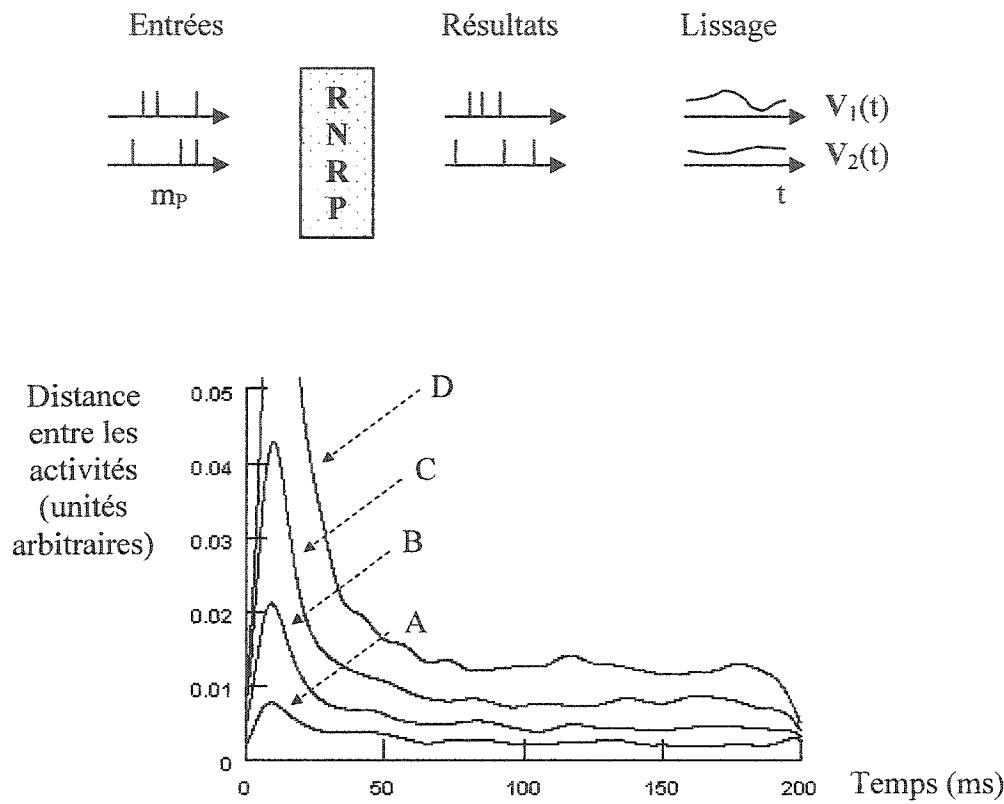
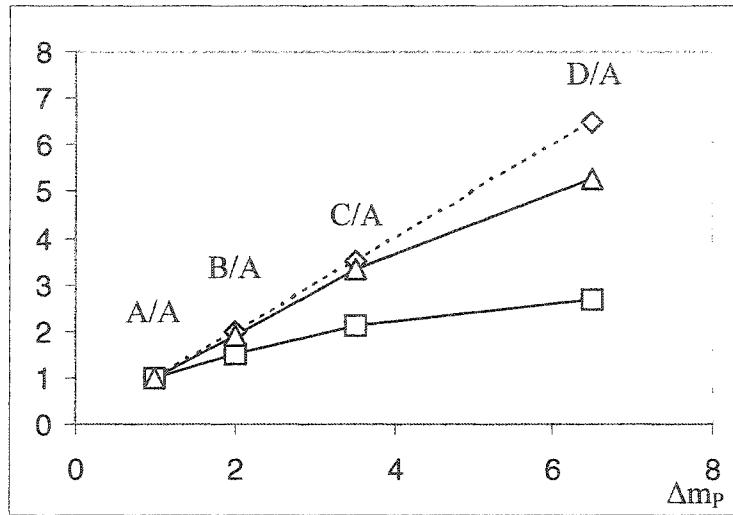


Figure 5.13.A : Comparaison des activités générées par des stimuli distincts.

On présente en entrée du RNP des trains de pulses de 200ms dont les durées inter-pulses suivent des lois de Poisson avec différentes moyennes m_p . Comme montré dans le schéma en haut, on construit le vecteur V (variable dans le temps) des activités lissées des neurones (avec une demi-largeur de fenêtre de 20ms, cf. chapitre 4 pour le lissage).

Les courbes montrent la variation dans le temps de la distance euclidienne entre les vecteurs V pour les moyennes suivantes : 3ms et 5ms (A), 3ms et 7ms (B), 3ms et 10ms (C) 3ms et 15ms (D). Le début de la stimulation provoque de grandes disparités dans l'activité neuronale (la courbe D dépasse du graphe). La distance se stabilise après une période transitoire de 50ms et on peut remarquer que plus les paramètres des lois de Poisson sont distincts, plus l'erreur est grande.



Figures 5.13.B : Évolution de la distance calculée sur le RNRP en fonction de la distance entre les stimuli.

On calcule la moyenne d^* de la distance entre les vecteurs V pendant la phase stable (de $t=75\text{ms}$ à $t=175\text{ms}$). On compare l'évolution de cette moyenne d^* (triangles) avec l'évolution du rapport entre les nombres totaux de pulses émis (carrés) par les RNRP pour les deux entrées dans chaque cas, lorsque l'écart entre les moyennes m_p des entrées (losanges, en pointillés) augmente (on ramène toutes les valeurs à celles du cas A, symbolisé par « /A », l'écart entre les moyennes m_p est en abscisses). On voit que la courbe des différences des totaux de pulses tend à s'infléchir, tandis que celle de d^* varie presque linéairement (comme la courbe relative aux écarts des moyennes m_p). Ce qui signifie que même si le nombre total de pulses dans le réseau ne varie plus beaucoup (entre les cas C et D), la structure temporelle de l'activité dans le RNRP rend compte des différences de nature des trains de pulses en entrées. Le lissage temporel est assez important, ce qui revient à comparer les densités d'activité au niveau de chaque neurone. C'est le grand nombre de fonctions générées dans le RNRP qui permet de discriminer les entrées; et même beaucoup lissées, les activités au niveau de chaque neurone constituent une information riche dans ce cas (à total de pulses émises constant). Cela illustre la puissance du RNRP en tant que MEL (conformément à [18]).

Dans le même ordre d'idée, l'auto-organisation en fonction des stimuli devrait augmenter topographiquement la différenciation des activités générées dans le RNRP pour les différents types de stimuli que l'on présente en entrée. Si les deux algorithmes sont complémentaires, on suppose alors que la distance entre des stimuli distincts augmente.

En ce qui concerne le neurone d'extraction en l'absence d'auto-organisation, son entraînement réussit correctement pour les tâches simples énumérées dans la partie 5.3, par exemple reconnaître une orientation par rapport à son opposé ou faire la distinction entre des barres mobiles verticales et horizontales (figure 5.14). Cependant, quelques difficultés subsistent pour obtenir d'un neurone d'extraction un profil d'activité réaliste avec une orientation préférentielle (courbe en cloche, cf. figure 2.10). Il suffit d'une centaine de stimuli pour l'adaptation du neurone d'extraction de manière générale, ce qui est assez rapide.

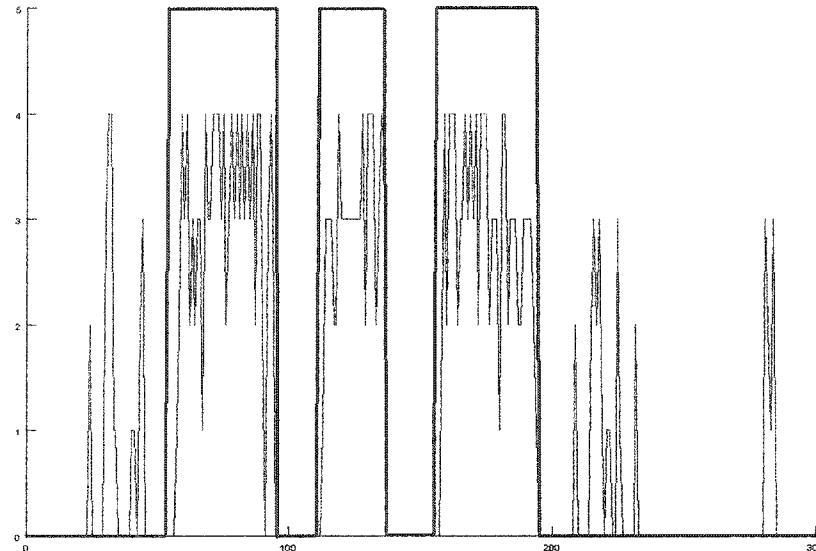


Figure 5.14 : Entraînement d'un neurone d'extraction pour discriminer des barres horizontales et verticales.

La courbe en gras correspond à la fonction-objectif et la courbe en trait fin à l'activité du neurone d'extraction. Ici, on a présenté deux types de stimuli : des barres mobiles de verticales (de haut en bas et de bas en haut) et d'autres horizontales (de gauche à droite et de droite à gauche); et le neurone d'extraction a appris à reconnaître le premier type. On voit que l'activité du neurone lors de la présentation d'un stimulus varie d'un facteur 10 environ lorsqu'il appartient au type appris.

5.4.3 Apport de l'auto-organisation

La convergence des poids d'un neurone lors de l'auto-organisation se fait sans devoir recourir à un rétrécissement du voisinage à l'intérieur duquel les mises à jour sont effectuées (cf. figure 5.15) : comme prévu, la pondération associée à la renormalisation entraîne une convergence normale du système lorsque l'on ne présente qu'un type de stimulus. La stabilisation s'opère également dans le cas dans le cas de quelques types de stimuli.

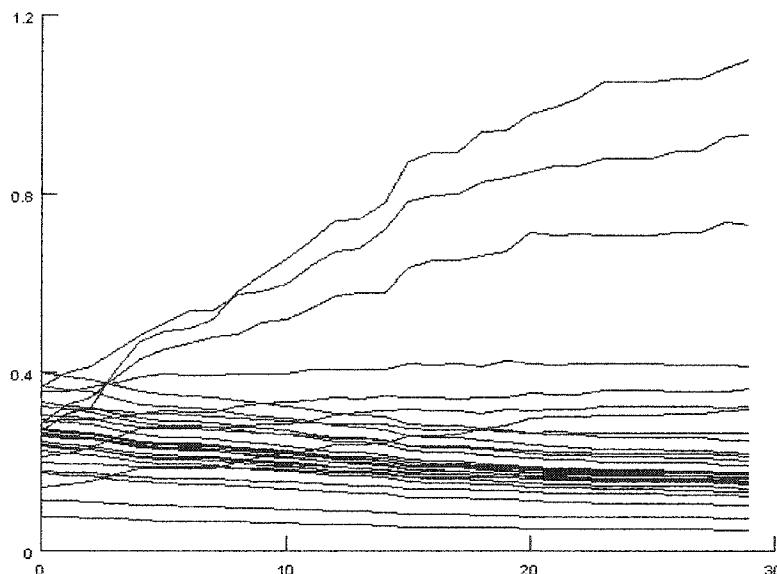


Figure 5.15 : Effet de la répétition d'un stimulus sur les connexions efférentes d'un neurone.

Les courbes correspondent aux variations des poids des connexions synaptiques pour un neurone excitateur, durant 30 présentations d'un même stimulus. On voit que certaines connexions deviennent privilégiées tandis que d'autres deviennent moins fortes, mais en restant significatives. Cela signifie que l'activité d'un neurone dépend de nombreux neurones encore après cette auto-adaptation, ce qui est nécessaire pour avoir une MEL.

L'auto-organisation du RNP engendre l'émergence de zones topographiques qui réagissent préférentiellement pour certains types de stimuli. Dans notre cas, on a entraîné un RNP avec des barres verticales mobiles (de gauche à droite ou de droite à gauche), et des barres horizontales (de haut en bas et de bas en haut), avec une erreur de $\pm 10^\circ$. Les zones présentent une préférence pour l'un des quatre types (cf. figure 5.16).

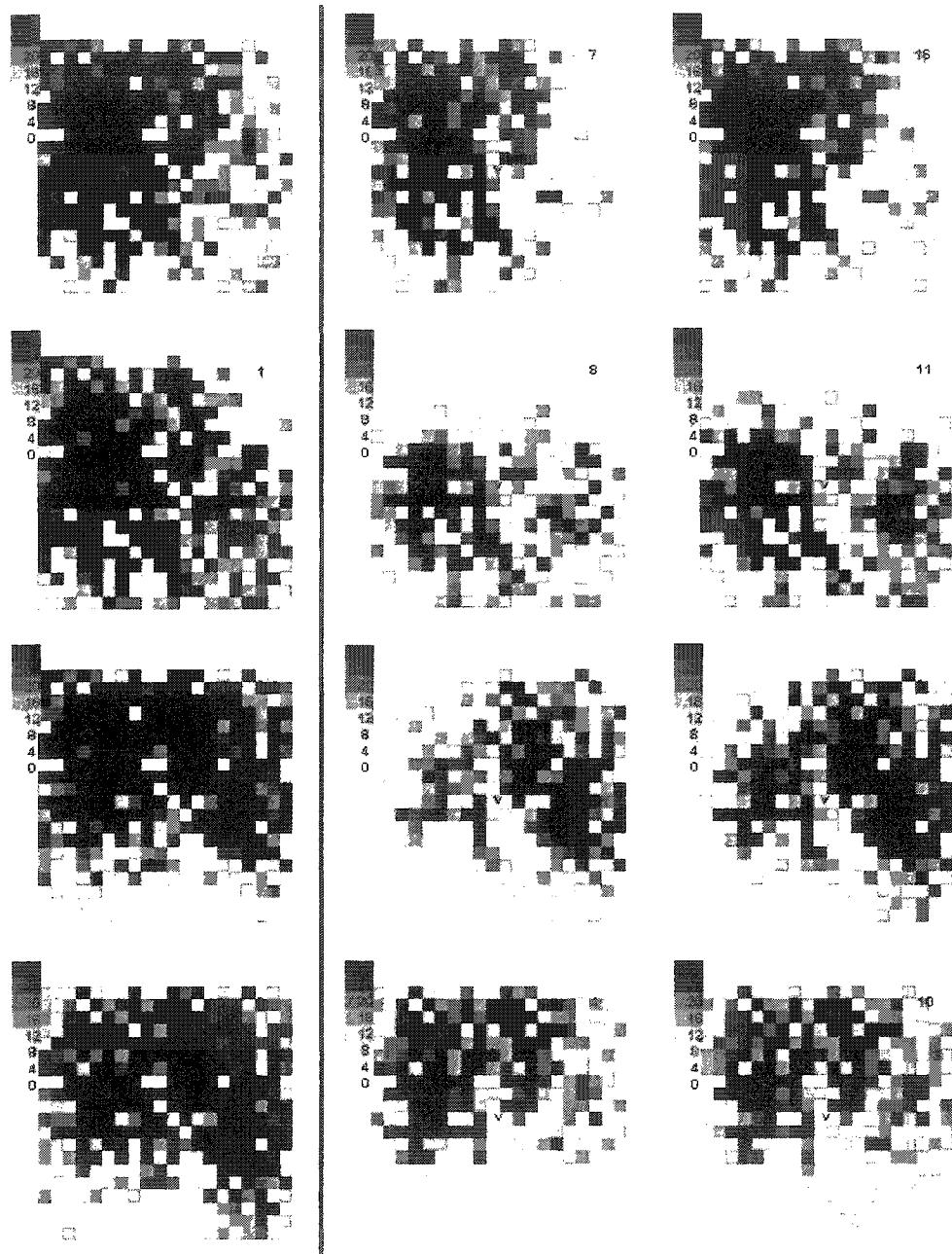
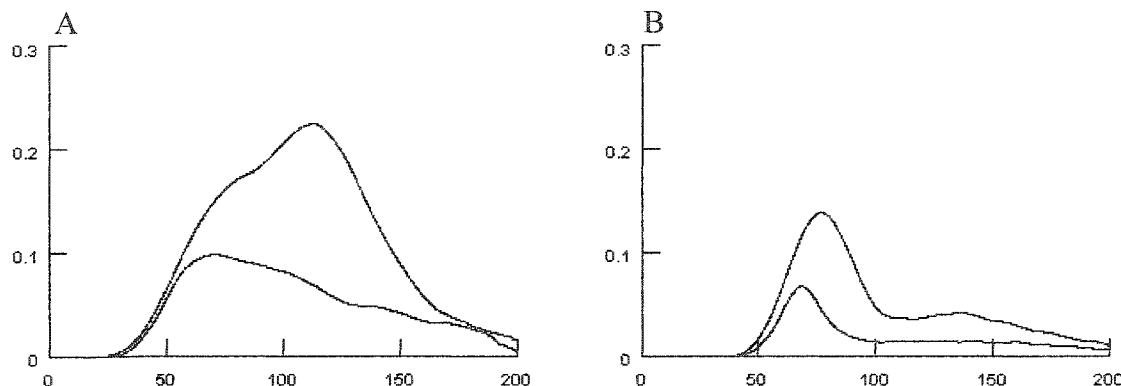


Figure 5.16 : Topographie de l'activité d'un RNP suite à son auto-organisation.
 On laisse le RNP s'auto-adapter en présentant 30 stimuli de 4 types distincts. Chaque carte représente l'activité cumulée durant une stimulation (chaque ligne correspond à un même type de stimuli) : plus la zone est sombre, plus les neurones situés dedans ont eu une activité élevée durant la stimulation. La colonne de gauche montre les activités avant l'auto-adaptation, celles de droite après 30 présentations. On voit que l'activité se concentre dans des zones topographiques distinctes pour chaque type, en particulier pour les 2 lignes du bas où les activités avant auto-adaptation sont semblables.

Cette auto-organisation topographique augmente la « distance » (telle que définie pour les figures 5.13) entre les activités générées par des stimuli de types différents (cf. figure 5.17). Il reste à montrer qu'une telle auto-organisation peut conduire à une organisation en colonnes similaire à celle obtenue par le modèle RF-LISSOM, mais il faut alors utiliser un nombre de neurones bien supérieur, afin que « chaque » orientation puisse être prise en compte par suffisamment de neurones pour former un région stable lors de l'auto-adaptation.



Figures 5.17 : Évolution de la distance entre les activités générées par des stimuli après auto-organisation du RNRP.

On présente des stimuli des 4 classes définies pour la figure 5.16. Dans la figure A, les 2 stimuli appartiennent à des classes distinctes, et on voit que la distance a augmenté suite à l'auto-organisation, ce qui est évident compte tenu que les zones topographiques de forte activité sont distinctes. Dans la figure B, il s'agit de stimuli appartenant à la même classe mais dont les vitesses de déplacement sont différentes. On voit que la distance a augmenté également, même si on n'observe pas de différence topographique majeure entre les activités comme dans le cas des orientations (cf. figure 5.16).

L'auto-organisation du RNRP influe de manière positive sur l'entraînement du neurone d'extraction : il est possible de l'entraîner plus finement dans un réseau auto-organisé (cf. figure 5.18). Ainsi, on peut l'entraîner à distinguer les déplacements lents de ceux rapides, ce qui signifie que malgré l'auto-adaptation, la structure fine des trains de pulses conserve les caractéristiques propres à une MEL. Un tel entraînement est difficile

à obtenir sans auto-organisation préalable, c'est-à-dire que l'information semble alors moins facile à extraire pour le neurone entraîné avec la règle p-delta.

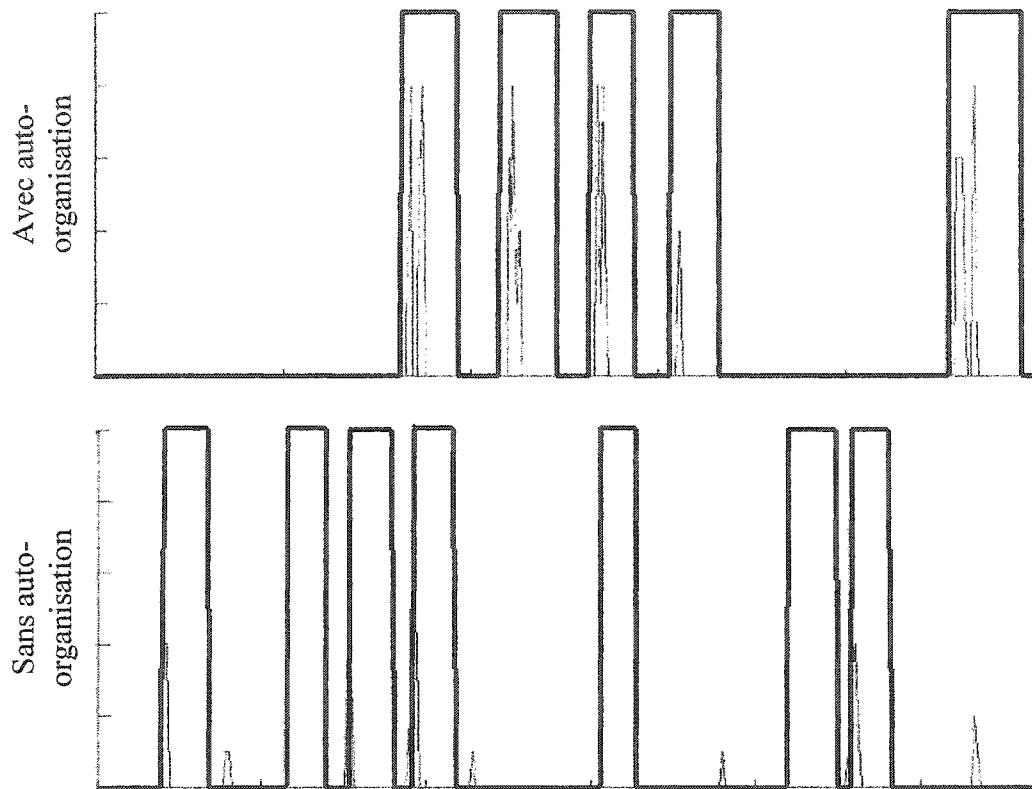


Figure 5.18 : Comparaison des performances d'un neurone d'extraction avant et après l'auto-organisation du RNP.

On a entraîné le neurone d'extraction du RNP à reconnaître les déplacements à vitesse élevée uniquement pour un des quatre types de stimuli (la fonction objectif est en trait gras). Le graphique montre les résultats pour 10 déplacements : 5 rapides (fonction-objectif au maximum) et 5 lents (fonction-objectif nulle); après auto-organisation (en haut) et avant auto-organisation (en bas). Sans auto-organisation, l'entraînement est beaucoup plus difficile et l'on arrive beaucoup moins bien à séparer les cas de vitesses rapide et lente. Avec auto-organisation, même si le neurone n'émet pas beaucoup de pulses, on voit qu'il fait bien la distinction entre les deux types de déplacements que l'on lui a demandé de reconnaître, puisqu'il est silencieux lorsque la fonction-objectif est nulle. De façon générale, il n'est pas facile de contrôler le nombre de pulses émises avec cet entraînement : des pulses sont émises en début de stimulation uniquement lorsque le motif est reconnu pour cette figure, tandis que pour la figure 5.14 c'est la densité de pulses qui évolue conformément à l'objectif fixé, mais le neurone n'est pas silencieux lorsqu'un motif non appris est présenté.

5.5 *Discussion*

On a donc construit un modèle monoculaire qui rend compte de phénomènes physiologiques du tissu cortical que l'on a énumérés dans la partie 2.4, à savoir une auto-organisation qui évoque les colonnes d'orientation préférentielle, la présence de neurones sensibles à certains stimuli (barres en mouvement de translation). Les deux concepts sont complémentaires et la CTAOEL s'auto-organise topographiquement au niveau des densités d'activité de ses neurones suivant les stimuli en entrée, tout en conservant une structure temporelle fine au niveau des trains de pulses émis par ces mêmes neurones, et conserve ainsi un caractère de MEL. On est amené dans un premier temps à raffiner le modèle en connectant chaque neurone d'extraction à une région précise du tissu, c'est-à-dire en vis-à-vis d'un nombre limité de récepteurs, en changeant d'échelle (au minimum plusieurs centaines de milliers de neurones) pour observer une véritable organisation en colonnes d'orientation, et en incluant des influx binoculaires. On peut également tester d'autres motifs de stimulation plus fins que la vitesse et l'orientation, comme des clignotements à diverses fréquences des barres lumineuses, afin de mieux tirer parti de

Plus formellement, cette étude illustre la puissance computationnelle du concept de CTAOEL : contrairement au modèle RF-LISSOM où les récepteurs transmettent au réseau des informations qui dépendent de l'orientation des stimuli, le RNP reçoit des trains de pulses fonction uniquement de la quantité de « lumière » reçue par un récepteur et c'est le neurone d'extraction qui effectue la corrélation entre les trains de pulses générés à partir de ces entrées pour extraire une information significative sur la nature des stimuli. Il n'y a donc pas de codage préalable des informations des stimuli, la CTAOEL peut s'organiser en fonction de n'importe quel type de signaux reçus. On a donc besoin de développer de meilleurs instruments d'analyse des performances du RNP en tant que MEL pour s'assurer d'une puissance minimale, et pour évaluer plus finement l'effet de l'auto-organisation sur l'entraînement du neurone d'extraction.

Une autre différence d'avec le modèle RF-LIISOM est que l'auto-organisation peut être permanente, puisque l'on n'a pas besoin de limiter progressivement le nombre des mises à jour pour parvenir à une stabilisation des poids. Mieux, cette stabilisation intervient lorsque l'on ne présente plus de nouveaux types de stimuli, mais redevient active lorsque de nouveaux apparaissent. Enfin, l'adaptation est rapide : seulement quelques dizaines de stimuli sont nécessaires à modifier significativement le comportement du réseau, on s'approche ainsi d'un calcul permanent (mieux qu'en temps réel tel que Maass le définit, c'est-à-dire que l'information reçue affecte le réseau qui la conserve pendant un certain temps intrinsèquement sans que l'on sache précisément le codage mis en oeuvre). Même si dans la réalité biologique, cette auto-organisation a lieu majoritairement durant les semaines suivant la naissance, cela peut se révéler utile dans une application en traitement du signal.

Enfin, on peut envisager de combiner en même temps l'auto-organisation permanente avec un entraînement supervisé des neurones d'extraction (en particulier la stabilisation des poids en entrée du neurone d'extraction). On pourrait aussi appliquer de manière permanente la règle p-delta sur les neurones d'extraction, mais en utilisant les dendrites apicales pour superviser cet apprentissage et en définissant les objectifs de manière « automatique » [28], de manière à se rapprocher encore de la réalité biologique où le cortex agit comme un filtre auto-adaptatif.

6 Conclusions et travaux futurs

On a montré dans le chapitre 4 la difficulté d'entraîner des sorties dans un RNRN même de petite taille en ajustant les paramètres synaptiques individuellement. Le chapitre 5 prend donc le contre-pied et vise à tirer parti de la multitude de signaux générés dans un RNRN de grande taille pour entraîner rapidement des sorties. Ainsi, on arrive à discriminer différents motifs de stimulation, avec un entraînement très rapide (quelques dizaines de stimuli, tous types confondus, suffisent). Le modèle construit rend compte de phénomènes physiologiques tels que la reconnaissance de motifs par des neurones spécialisés (correspondant à des cellules pyramidales). On utilise en sus des règles locales simples inspirées de la loi de Hebb pour auto-organiser le comportement du RNRN en fonction des stimuli reçus. A noter que cette auto-organisation peut être faite avec des signaux très généraux, sans codage particulier. Il s'ensuit une répartition topographique de l'activité (au niveau des densités d'activité des neurones) avec des zones spécialisées pour chaque type de stimuli présentés. Sous les conditions définies dans ce mémoire, la disparité entre les activités des neurones générées par chaque type de stimuli selon une mesure définie par Maass augmente, mais la structure temporelle fine des trains de pulses est respectée et le réseau conserve un caractère de MEL. Ces deux concepts complémentaires fonctionnent très rapidement, ce qui est réaliste du point de vue biologique.

L'auto-organisation du tissu en colonnes corticales nécessite de changer d'échelle pour simuler un réseau contenant plusieurs centaines de milliers de neurones au lieu du millier utilisé dans le chapitre 5 (la densité de cellules sera alors réaliste), ainsi qu'un modèle de rétines binoculaires couvrant une fenêtre de perception plus grande. De plus il est suffisamment souple pour intégrer d'autres aspects du comportement du cortex visuel : reconnaissance de contours, de couleurs, etc.; en plus du mouvement et des orientations. Le codage de l'information visuelle au niveau des CGL peut également être

intégrée (à la manière du modèle RF-LIISOM qui a servi de base à un modèle du système visuel plus global G-LIISOM avec les CGL et d'autres aires visuelles). Une autre amélioration concerne le type de neurones utilisés dans le réseau : des neurones HH peuvent permettre de reconstruire un vrai réseau 3D pour simuler le tissu biologique de manière plus précise, avec des connexions entre des neurites proches dans l'espace reconstruit et des délais de transmission des PA plus réalistes. Enfin, lorsque l'on voudra appliquer ce modèle au cas de l'implant visuel développé par l'équipe PolySTIM, il faudra un modèle d'électrode adéquat qui va permettre de modéliser la matrice au sein du tissu et de tester les protocoles de stimulation. Il pourra alors entraîner des neurones en sortie du tissu à reconnaître certains types de composantes (déplacements, segments orientés, etc.) et tester des protocoles de stimulation pour en mesurer l'effet sur ces neurones d'extraction.

D'un point de vue plus général, une CTAOEL peut être considérée comme un filtre auto-adaptatif dont on peut créer des sorties (avec des neurones I&E comme dans ce mémoire ou avec des perceptrons comme Maass). L'apport de l'auto-structuration au concept de MEL développé par Maass consiste en un prétraitement des signaux reçus, ce qui rend l'ajustement des poids du neurone d'extraction plus facile pour une tâche fixée. L'ensemble respecte en sus la notion de calcul en temps réel (« real-time computation » [18]) puisque l'auto-organisation permanent se stabilise pour un réseau suffisamment grand lorsque les types de stimuli présentés ont été déjà présentés, et qu'alors un neurone d'extraction entraîné calcule une fonction de sortie qui dépend des signaux présentés antérieurement, avec une mémoire temporaire.

BIBLIOGRAPHIE

- [1] AMARI S., MURATA N., MULLER K., FINKE M., YANG H. H., "Statistical Theory of Overtraining - Is Cross-Validation Asymptotically Effective?", In: *Advances in Neural Information Processing Systems 8*, Editors D. Touretzky, M. Mozer, and M. Hasselmo, pp.176-182, MIT Press, Cambridge MA. (1996)
- [2] AUER P., BURGSTEINER H., MAASS W., "The p-delta learning rule for parallel perceptrons.", *submitted for publication.* (2001)
- [3] BUDD J. M. L., KISVÁRDAY Z. F., "Local lateral connectivity of inhibitory clutch cells in layer 4 of cat visual cortex (area 17)", *Springer-Verlag.* (2001)
- [4] CHOE Y., MIIKKULAINEN R., "Self-organization and segmentation with laterally connected spiking neurons.", *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, pages 1120–1125, San Mateo, CA. (1997)
- [5] DOBELLE W.H., MLADEJOVSKY W.G., "Phosphenes produced by electrical stimulation of human occipital cortex, and their application to the development of a prothesis for the blind." *J. Physiol.*, 243, 553-576. (1974)
- [6] FOERSTER O., "Beitrage zur Pathophysiologie der Sehbahn und der Sehsphare." *J. Psychol. Neurol., Lpz*, 39, 463-485. (1929)
- [7] GOLDBERG D.-E., "Genetic Algorithms in Search Optimization and Machine Learning", Addison-Wesley. (1989)
- [8] HAMBRECHT F. T., "Visual prostheses based on direct interfaces with the visual system." *Bailliere's Clinical Neurology Vol.4 #1*, Tindall, London. (1995)
- [9] HARVEY J.-F., ROY M., SAWAN M., "Visual Cortex Stimulation Prototype based on Mixed-Signal Technology Devices.", *Int. Functional Electrical Stimulation Society Conference (IFESS'99)*, Sandai (Japan). (1999)
- [10] HODGKIN A. L., HUXLEY A. F., "A Quantitative Description of Membrane Current and its Application to Conduction and Excitation in Nerve" *Journal of Physiology* 117: 500-544. (1952)

- [11] HOPFIELD J. J., "Neural networks and physical systems with emergent collective computational abilities.", *Proceedings of the National Academy of Sciences*, 79:2554-2558 (1982)
- [12] <http://webvision.med.utah.edu/VisualCortex.html>
- [13] <http://www.dobelle.com>
- [14] <http://www.neuron.yale.edu/neuron>
- [15] HUBEL D. H., WIESEL T. N., "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex." *J. Physiol.*, 160:-154. (1962)
- [16] KOHONEN T., "Self-organised formation of topologically correct feature map", *Biological Cybernetics*, 49:53-69. (1982)
- [17] KOHONEN T., "Self-organization and Associative Memory", 3rd ed. New York, Springer-Verlag. (1988)
- [18] MAASS W., NATSCHLÄGER T., MARKRAM H., "Real-time computing without stable states: A new framework for neural computation based on perturbations.", *Neural Computation*, 14(11):2531-2560. (2002)
- [19] MARKRAM H., WANG Y., TSODYKS M. "Differential signaling via the same axon of neocortical pyramidal neurons." *Proc. Natl. Acad. Sci.*, 95:5323–5328. (1998)
- [20] NATSCHLÄGER T., MAASS W., ZADOR A., "Efficient temporal processing with biologically realistic dynamic synapses." *Network: Computation in Neural Systems*, 12:75-87. (2001)
- [21] PRINCIPE J., GILES L., MORGAN N., WILSON E. (Editors), "Neural Networks for Signal Processing", *IEEE Press*. (1997)
- [22] REICH D.S., MECHLER F., VICTOR J.D., "Temporal coding of contrast in primary visual cortex: when, what, and why?" *J. Neurophysiology*, 85, 1039-1050. (2001)
- [23] ROUSCHE P.J., NORMANN R.A., "Chronic intracortical microstimulation (ICMS) of cat sensory cortex using the Utah Intracortical Electrode Array." *IEEE Trans. Rehab. Eng.*, 7, 56-68. (1999)

- [24] ROY M., "Conception et réalisation d'un prototype de la partie implantable d'un stimulateur visual cortical." *mémoire de maîtrise*, École Polytechnique de Montréal. (1999)
- [25] RUMELHART D. E., HINTON, G. E., WILLIAMS, R. J., "Learning representations by error propagation.", in *Parallel Distributed Processing*, pp. 318-362. (1986)
- [26] SCHMIDT E. M., BAK M. J., HAMBRECHT F. T., KUFTA C. V., O'ROURKE D. K., VALLABHANATH P., "Feasibility of a visual Prosthesis for the blind based on intracortical microstimulation of the visual cortex." *Brain* 119: 507-522. (1996)
- [27] SILLITO A. M., KEMP J. A., MILSON J. A., BERARDI N., "A re-evaluation of the mechanisms underlying simple cell orientation selectivity." *Brain Res.*, 194: 517-520. (1980)
- [28] SPRATLING M. W. "Cortical region interactions and the functional role of apical dendrites.", *Behavioral and Cognitive Neuroscience Reviews*, 1(3):219-28. (2002)
- [29] VOGELSTEIN J.V., DAGNELIE G., "Phosphene Mapping Strategies for a Cortical Visual Prosthesis." Poster presentation at: *29th Neural Prostheses Workshop*, NINDS at NIH, Bethesda, Maryland. (1998)